

École Docotorale ED488 Sciences, Ingénierie, Santé

Theory and Algorithms for Learning Metrics with Controlled Behaviour

---

Théorie et Algorithmes pour l'Apprentissage de Métriques à Comportement Contrôlé

Thèse préparée par **Michaël Perrot**
au sein de l'**Université Jean Monnet de Saint-Étienne**
pour obtenir le grade de :

Docteur de l'Université de Lyon
Spécialité : **Informatique**

Univ Lyon, UJM-Saint-Etienne, CNRS, Institut d Optique Graduate School,
Laboratoire Hubert Curien UMR 5516, F-42023, SAINT-ETIENNE, France.

Thse soutenue publiquement le **13/12/2016** devant le jury composé de :

Stéphane CANU	Professeur, INSA de Rouen	Rapporteur
Élisa FROMONT	Maître de Conférences HDR, Université de Saint-Étienne	Examinatrice
Amaury HABRARD	Professeur, Université de Saint-Étienne	Directeur
Liva RALAIVOLA	Professeur, Aix-Marseille Université	Rapporteur
Marc TOMMASI	Professeur, Université de Lille	Examinateur
Kilian Q. WEINBERGER	Associate Professor, Cornell University	Examinateur

Table of Contents

List of Figures

List of Tables

Introduction

Machine Learning is a field of *Artificial Intelligence* aiming at acquiring new knowledge from data. This new knowledge generally takes the form of a model, learned from a limited number of observed examples, and able to generalize well to future queries. In other words the goal is to learn how to automatically solve a problem from a finite set of observations. For example, the objective in spam detection is to use the annotated mail box of a user to learn how to separate solicited emails from unsolicited ones; in tracking the underlying problem is to follow an object in a video; in face recognition the goal is to identify a person in a set of images.... The large diversity of problems raised in machine learning has attracted a lot of attention in the past and still deserves a lot of active research.

In this thesis we are mainly interested in *Supervised Learning* problems. The idea behind this paradigm is that the examples are accompanied by a label. This label can be either a value or a class and it corresponds to the solution of the problem for the given example. As illustrative examples, let us consider the problems of house pricing and poisonous mushrooms recognition. For the former the goal is to predict the price of a house, each example then corresponds to a set of characteristics of a particular building while the label is its price. For the latter we want to recognize from images poisonous mushrooms from edible ones, each example is the picture of a mushroom while the label is its class, i.e. poisonous or not. From these examples one can see the importance, in supervised learning, of generalization to new data. Indeed, the labels for the training examples are already known, if the model cannot find the correct labels for new examples then its interest becomes limited. Note that the previous examples correspond to two widely studied problems in supervised learning, namely regression and classification. The difference between the two is that the goal of the former is to predict a continuous value while the objective of the latter is to guess the correct class.

Supervised learning is not the sole paradigm existing in machine learning. In fact it can be opposed to *Unsupervised Learning* where examples are unlabelled. For example a widely studied problem in this paradigm is called clustering. The underlying goal is to obtain a meaningful partition of the space where the examples share common properties. The performance of unsupervised learning algorithms is often difficult to assess as, contrary to the supervised learning case, there is no labels to provide an obvious feedback on the model.

Drawing from these two paradigms, the idea behind *Semi-Supervised Learning* is to consider two sets of examples where the first one is labelled while the second one is not. In this

case the goal is often to use the labelled examples to help solve an unsupervised learning task or to consider the unlabelled examples to aid in a supervised learning problem.

So far we have considered that the goal of a machine learning approach is to solve a single task. Taking a different point of view the idea behind *Transfer Learning* is to transfer some knowledge learned on a so-called source to a so-called target. Following this idea, in *Domain Adaptation* the goal is to transfer the model learned on a source task to solve a different but related target problem. For example, in the spam detection problem, the two tasks could be to detect unsolicited emails from the mailboxes of two different users. The two users share the same problem but their email distributions might differ, e.g. because they did not subscribe to the same mailing lists. In this case the goal is to adapt the model learned from one of the users to the other.

In this manuscript we will see that despite being mainly interested in supervised learning problems, several of our contributions also share some ties with the other paradigms presented here.

When presenting the supervised learning paradigm we stressed the fact that a model, learned with a limited number of training examples, should generalize well to new examples. One way to verify this property is to evaluate the learned model on a set of new test examples independent from training examples and for which the solution to the problem is known. However the number of test examples that can be obtained is often limited. It might make this approach insufficient to ensure that the model generalizes well. Other approaches are then necessary. To this extent note that a common assumption in machine learning is that the task that we want to solve is completely defined by an unknown distribution from which the training examples are drawn. Then, one possible solution is to use a cross-validation procedure where the idea is to partition the learning sample into k parts. The model is learned on $k-1$ parts and tested on the last one. This procedure is repeated k times, i.e. until each part was used as a test set, and the accuracy is averaged over the different test samples. Anyway, this procedure still requires a significant amount of examples to be valid. Another possibility following the assumption evoked previously consists in theoretically studying the learning algorithm in order to derive so-called generalization bounds. The idea behind these bounds is to show that the true error of the model, i.e. its error on the unknown distribution, is bounded by its empirical error, i.e. its error on the training sample, plus a term which decreases when the size of the training set increases. Obtaining such bounds guarantees that models learned by the concerned algorithm generalize reasonably well.

Many different approaches have been proposed to solve supervised learning problems. Among these several rely on a notion of distance or similarity between the examples to learn a model. A very representative example is the nearest neighbour classifier which is based on the idea that two similar examples should share the same label. Another example is the support vector machine algorithm. It proposes to classify examples depending on their similarity to landmarks points called the support vectors. In these two examples the notion of

similarity is of critical importance. However different tasks often call for different measures of similarity. As an example recall the two examples used previously in this introduction, it does not make sense to compare houses and mushrooms in the same way. Manually choosing an appropriate measure of similarity can be tedious and difficult. However it might be possible to automatically infer it from the data. This is the underlying idea behind *Metric Learning* which is the field of interest of this thesis.

We identify several limits of the current approaches in metric learning. First some methods propose to make use of side informations to help during the learning process. However there is no theoretical understanding of the impact of such information on the learned metric. Second the intrinsic properties of the learned metrics are often the same. Indeed metrics are usually learned with the objective of bringing closer similar examples while pushing far away dissimilar ones. In some cases it might be interesting to consider different kinds of constraints. One example is to obtain a metric whose behaviour is not limited to the examples but is more global in the sense that it is, for example, able to move masses of examples together. A third limit to current approaches is that there is often no theoretical justifications on the proposed approaches, i.e. there is no guarantees on the generalization ability of the learned metrics.

Contributions: Learning Metrics with Controlled Behaviour

In this thesis we propose several approaches to learn metrics whose behaviour is controlled. First we propose to use side informations in the form of a reference metric to either strictly or loosely guide the learned metric. Hence in our first contribution we propose to address the problem of regressing the values of a reference metric only accessible through a limited training set. In our second contribution we theoretically study how using a reference metric coming from a related but different problem can help during the learning process. In particular we derive several measures of goodness of the reference metric for the problem at hand. Second we propose two approaches able to consider new kinds of constraints for metric learning. Hence in our third contribution we consider that the training examples should not be moved with respect to each other but rather with respect to some virtual points which lay in the output space of the learned metric. By this way it is possible to carefully control the movement of each example. In our fourth contribution we build upon our third contribution and on recent advances in Optimal Transport to propose a new approach to learn a metric able to move masses of examples across the space. As a last remark, note that throughout this thesis we put particular emphasis on providing theoretically sound approaches.

Outline

In the first part of this thesis we propose some preliminary informations. In the first chapter we introduce some concepts that are used throughout the manuscript while in the second chapter we propose a review of the state of the art in metric learning.

Chapter 1 The first chapter of this thesis is dedicated to the presentation of several notions and tools used throughout it. The first part of the chapter presents the risk minimization framework which is the basis of all our algorithmic contributions. The second part is dedicated to the theoretical analysis of algorithms. More precisely we present two frameworks used to derive generalization bounds. They correspond to the uniform stability and the Rademacher complexity frameworks. The third part is interested in the notion of losses and regularization terms. These elements are core in the formulation of a regularized risk minimization problem. Through several examples we show that there exist a wide range of possibilities with different properties. This second part is also interested in the formal definition of the notion of metric as a general term to design a similarity, a dissimilarity or a distance. As for the losses and regularization terms, several examples are presented. The last part of this first chapter introduces other useful notions such as the nearest neighbour classifier, which we often use with our leaned metrics, and the domain adaptation setting in which two of our contributions are evaluated.

Chapter 2 The second chapter of this thesis corresponds to a review of metric learning. Here we present the main approaches which have made the success of the field. We propose to divide this review in four parts answering four basic questions about metric leaning problems. In the first part we consider the problem of the kind of metrics which can be learned. Then in the second part we answer the question of how, technically, these metrics can be learned. In the third part of this chapter we review some approaches deriving theoretical guarantees for metric learning. In the last part we present several works interested in making use of metric learning for different kind of applications ranging from classification to clustering or domain adaptation.

In the second part of this thesis we present our first two contributions which are interested in using reference metrics to help during the metric learning process.

Chapter 3 In the third chapter of this thesis we present our first contribution. It corresponds to a metric learning method able to approximate an existing metric. The first part of this chapter is dedicated to the presentation of the main optimization problem considered. It corresponds to a regression of the values of a metric. Furthermore we show that when the reference metric is too complex it is possible to use local metric learning to obtain a better approximation. In the second part we present a theoretical analysis of the approach both in the global and the local settings. It shows that the metrics learned by our algorithm generalize well. In the third and fourth parts we consider the problem of learning perceptual color differences to show the interest of our approach in a real life application.

Chapter 4 The fourth chapter of this thesis is dedicated to our second contribution. As in the third chapter it corresponds to a metric learning approach able to use some knowledge given by an existing metric. The difference is that, this time, we do not want to approximate

this metric but we want to use some information that it carries to help during the learning process. This contribution is thus strongly related to the field of transfer learning/domain adaptation. The chapter is divided in seven parts. In the first part we present the framework of Metric Hypothesis Transfer Learning which corresponds to a minimization problem equipped with a biased regularization term. In the second, third and fourth parts we propose a theoretical analysis of metric hypothesis transfer learning using three different theoretical frameworks. It allows us to derive different notions of goodness of the reference metric. In the fifth part of the chapter we summarize the different bounds and in the sixth part we present several loss functions and regularization terms which fall into our framework. In the last part we show that this framework can be used in practice to obtain competitive results on several widely used transfer learning problems.

In the last part of this thesis we introduce our last two contributions where we propose new ways to control the behaviour of the learned metric.

Chapter 5 In the fifth chapter we present our third contribution. Here instead of using standard similarity and dissimilarity constraints we propose to consider that the metric should bring the examples closer to virtual points defined a priori. It allows us to learn a metric with a regression and to reduce the number of constraints considered. In the first part of the chapter we present our algorithm. In the second part we address the problem of selecting the virtual points and defining the constraints. In the third part we propose a theoretical analysis of the algorithm showing that learning a metric with our approach is founded but also that it is possible to obtain some links with a standard metric learning method. In the last part we validate our approach with several experiments.

Chapter 6 The sixth chapter introduces the last contribution of this thesis. It corresponds to a new method able to learn a metric that moves masses of examples by approximating the transformation corresponding to the solution of an Optimal Transport problem. In the first part of this chapter we formally introduce the problem of optimal transport. In the second part we present our formulation while in the third we propose an efficient way to optimize it. In the fourth part we discuss some theoretical aspects showing that if standard assumptions made in the optimal transport community are correct, then our approach is founded. In the last part we empirically validate our approach on a domain adaptation and an image editing problem.

Notations

In this thesis $\mathbb{R}$ and $\mathbb{R}_+$ respectively represent the sets of real and non negative real numbers. A vector is denoted by a bold lower case letter. For example $\mathbf{x} \in \mathbb{R}^d$ is a d-dimensional real valued column vector. For $i \in \{1, \dots, d\}$, $\mathbf{x}(i)$ is its i-th feature. In a similar fashion, matrices are denoted by a bold upper case letter. For examples $\mathbf{M} \in \mathbb{R}^{d \times d'}$ is a real valued matrix with d rows and d' columns. We also denote by $\mathbb{S}$ the set of symmetric real valued

matrices and by $\mathbb{S}_+$ the set of symmetric positive semi-definite matrices. Let $i \in \{1, \ldots, d\}$ and $j \in \{1, \ldots, d'\}$, $\mathbf{M}(i, j)$ corresponds to the value at row i and column j while $\mathbf{M}(i,)$ and $\mathbf{M}(, j)$ are respectively the row and column vectors of indices i and j. $\mathbf{x}^T$ and $\mathbf{M}^T$ stand for the transpose of vector $\mathbf{x}$ and matrix $\mathbf{M}$. $\langle \cdot, \cdot \rangle$ represents the dot product between two vectors while $\langle \cdot, \cdot \rangle_{\mathcal{F}}$ corresponds to the Frobenius product[1] between two matrices.

We are interested in supervised learning. Hence throughout this thesis we consider that we are working in a domain $\mathcal{T}$ corresponding to the space $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ equipped with a probability distribution $\mathcal{D}_{\mathcal{T}}$. In this case $\mathcal{X} \subseteq \mathbb{R}^d$ is the example space while $\mathcal{Y}$ is the label space, e.g. $\mathcal{Y} = \{-1, 1\}$ in a binary classification problem. We consider that we have access to a set of n examples $T = \{\mathbf{z}_i = (\mathbf{x}_i, y_i)\}_{i=1}^n$ with $\mathbf{x}_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$. The fact that the examples of the set T are drawn i.i.d. from the distribution $\mathcal{D}_{\mathcal{T}}$ is denoted as $T \sim \mathcal{D}_{\mathcal{T}}$. In matrix form we write $T = (\mathbf{X}, \mathbf{y})$ where $\mathbf{X}$ contains one example per row and $\mathbf{y}$ is a column vector of the labels.

We denote by $[\cdot]_+$ the hinge loss function, i.e. $[x]_+ = \max(0, x)$, $\mathbb{E}_{X \sim \mathcal{D}_{\mathcal{T}}}[X]$ corresponds to the expectation of the random variable X drawn from the distribution $\mathcal{D}_{\mathcal{T}}$, $\Pr(E)$ denotes the probability of an event E and $\mathcal{D}_{\mathcal{T}}(x)$ corresponds to the probability of drawing x from distribution $\mathcal{D}_{\mathcal{T}}$. We denote the composition of two functions as $g \circ f$, i.e. $(g \circ f)(x) = g(f(x))$.

All the notations are summarized in Table 1.

In the various mathematical proofs of this thesis we propose to explain the derivations step by step by adding a justification surrounded by brackets and flushed on the right between the concerned lines.

[1] $\langle \mathbf{A}, \mathbf{B} \rangle_{\mathcal{F}} = \mathrm{Tr}\left(\mathbf{A}^T \mathbf{B}\right)$

Table 1: Notations.

Notation	Description
$\mathbb{R}, \mathbb{R}_+$	Sets of real and non negative real numbers
B, C	Constants
x	Scalar
$\mathbf{x}$	Vector
$\mathbf{X}$	Matrix
$\mathbf{X}(i,),\mathbf{X}(,j)$	Row i and column j of matrix $\mathbf{X}$
$\mathbf{X}(i,j)$	Entry in row i and column j of matrix $\mathbf{X}$
$\mathbf{x}(i)$	Entry i in vector $\mathbf{x}$
y	Label
$\mathcal{X}, \mathcal{Y}, \mathcal{H}, \mathcal{M}$	Input Space, Output Space, Hypothesis Space, Space of Metrics
$\mathbb{S}, \mathbb{S}_+$	Sets of symmetric and symmetric positive semi-definite matrices
$\mathcal{S}, \mathcal{T}$	Domains
S, T	Sets
$\mathcal{D}_\mathcal{S}, \mathcal{D}_\mathcal{T}$	Distributions over the domains $\mathcal{S}$ and $\mathcal{T}$ respectively
$f(\cdot)$	Function
$\|\cdot\|$	Norm
$\lceil\cdot\rceil$ $\lfloor\cdot\rfloor$	Ceil and floor functions
$\|\cdot\|$	Absolute value
$[\cdot]_+$	Hinge loss function
$\langle\cdot,\cdot\rangle$	Dot product between vectors
$\langle\cdot,\cdot\rangle_\mathcal{F}$	Frobenius product between matrices
$\mathbb{E}[\cdot]$	Expectation
$\Pr[\cdot]$	Probability
l	Loss function
$L_\mathcal{T}$	True risk over the domain $\mathcal{T}$
$\hat{L}_T$	Empirical risk over the set T
$\mathcal{A}$	Algorithm
$f \circ g$	Composition of functions

Part I

Background

Chapter 1

Preliminaries

Abstract

In this chapter we present several notions used throughout this thesis. In particular we formalize the risk minimization framework in which fall our algorithmic contributions. From a theoretical standpoint we present two frameworks interested in deriving generalization guarantees for risk minimization. As we will see in the next chapter these two frameworks have been successfully extended to the metric learning problem. We use them to theoretically analyse our contributions demonstrating the ability of our algorithms to learn metrics able to generalize well. From a more practical point of view we present several loss functions and regularization terms which can be used in the risk minimization framework and we introduce a formal definition of the notion of metric considered in this thesis. Finally we present the nearest neighbour classifier and the domain adaptation setting which will be used to empirically demonstrate the interest of most of our contributions.

1.1 Introduction

In this chapter we are interested in supervised learning problems. We consider that we have access to a domain $\mathcal{T}$ which corresponds to the space $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ equipped with the probability distribution $\mathcal{D}_{\mathcal{T}}$. In this case $\mathcal{X} \subseteq \mathbb{R}^d$ is the example space and $\mathcal{Y}$ is the label space. The goal is to find the correct relation between the examples in $\mathcal{X}$ and the labels in $\mathcal{Y}$. In other words we are looking for an hypothesis $h : \mathcal{X} \to \mathcal{Y}$ coming from an hypothesis space $\mathcal{H}$ and able to solve the problem of associating each example in $\mathcal{X}$ with the correct value in $\mathcal{Y}$. A key assumption in machine learning is that the distribution $\mathcal{D}_{\mathcal{T}}$ is unknown and that we only have access to it through a finite size sample $T = \{\mathbf{z}_i = (\mathbf{x}_i, y_i)\}_{i=1}^n$. This sample of size n is assumed to be representative of the true distribution and is called the training set. One of the key objective is then to use T to learn an hypothesis h which generalizes well to new examples drawn from the distribution $\mathcal{D}_{\mathcal{T}}$. To present this notion of generalization we start by introducing two notions of risk of an hypothesis. Before that we address the problem of assessing the performance of an hypothesis with respect to an example.

One of the most intuitive way to assess the performance of a model for a given problem is to measure its error. However the notion of error can vary from one problem to another. To illustrate this let us consider an example $\mathbf{z} = (\mathbf{x}, y)$ and the two problems of classification and regression. In classification the label space $\mathcal{Y}$ is discrete and of limited size, e.g. in binary classification $\mathcal{Y} = \{-1, 1\}$. The goal is to correctly choose the class of an object. Hence an error is the prediction of the wrong class, i.e. $h(\mathbf{x}) \neq y$. In regression the label space $\mathcal{Y}$ is continuous, i.e. $\mathcal{Y} = \mathbb{R}$. The goal is to return the correct value given an example. Hence an error is the prediction of a value far from the ground truth, i.e. $h(\mathbf{x}) \ll y$ or $h(\mathbf{x}) \gg y$. In this thesis we consider that the error, or the risk, is defined with respect to a loss function able to quantify it. More formally we consider that each problem is associated with a loss function $l : \mathcal{H} \times \mathcal{Z} \to \mathbb{R}_+$ which, given an hypothesis $h \in \mathcal{H}$ and an example $\mathbf{z} = (\mathbf{x}, y) \sim \mathcal{D}_\mathcal{T}$ is able to return a positive real value in $\mathbb{R}_+$. This value is a numerical representation of the error committed by the hypothesis on the example. It should be large if the error is significant and small otherwise. Considering the same examples as before we can define intuitive loss functions. On the one hand in classification we can consider the following loss, called the 0/1 loss and presented in Figure 1.2:

$$l\left(h, \mathbf{z}\right) = \begin{cases} 0 & \text{if } h(\mathbf{x}) = y, \\ 1 & \text{otherwise.} \end{cases} \tag{1.1}$$

On the other hand in regression we can consider the following loss function, called the absolute loss and presented in Figure 1.3:

$$l\left(h, \mathbf{z}\right) = \left|h(\mathbf{x}) - y\right|. \tag{1.2}$$

Note that we come back to this notion of loss function in Section 1.4 where we give a formal definition and several examples. We can now define the notions of empirical and true risk in the two following definitions.

Definition 1.1 (Empirical risk)**.** *Given a loss function $l : \mathcal{H} \times \mathcal{Z} \to \mathbb{R}_+$ and a set of examples T, the empirical risk of an hypothesis h is defined as:*

$$\hat{L}_T(h) = \frac{1}{n} \sum_{\mathbf{z} \in T} l\left(h, \mathbf{z}\right).$$

It corresponds to the average error of the hypothesis on the training set.

Definition 1.2 (True risk)**.** *Given a loss function $l : \mathcal{H} \times \mathcal{Z} \to \mathbb{R}_+$ and a distribution $\mathcal{D}_\mathcal{T}$, the true risk of an hypothesis h is defined as:*

$$L_\mathcal{T}(h) = \mathop{\mathbb{E}}_{\mathbf{z} \sim \mathcal{D}_\mathcal{T}} l\left(h, \mathbf{z}\right).$$

It corresponds to the expected error of the hypothesis on the whole distribution. The goal of a supervised learning algorithm is to learn an hypothesis with the smallest possible true risk. However this quantity is only theoretical and cannot be computed. Indeed in practice

we do not have access to the distribution $\mathcal{D}_{\mathcal{T}}$ but only to the so called training set T which is assumed to be a good empirical approximation of $\mathcal{D}_{\mathcal{T}}$. Hence we say that an algorithm generalizes well when the difference between the true risk and the empirical risk is small. In Section 1.3 we will see that even if this difference cannot be computed it can sometimes be upper bounded by a small value in a generalization bound.

We have seen that we have access to the empirical risk but not to the true risk. However we argued that the ideal hypothesis should have the smallest possible true risk. Hence in Section 1.2 we discuss the problem of empirically learning an hypothesis and we see that directly minimizing the empirical risk might not always be a good idea. In Section 1.3 we consider a theoretical standpoint and we address the problem of linking the empirical risk of an hypothesis to its true risk. In Sections 1.4 and 1.5 we give some formal definitions and examples of several notions that will be used throughout this thesis.

1.2 Learning by Risk Minimization

In this first section we address the problem of learning an hypothesis h when we only have access to a training set T and not to the whole distribution $\mathcal{D}_{\mathcal{T}}$. We first propose to tackle the problem by simply minimizing the empirical risk. We will see that this Empirical Risk Minimization approach presents some drawbacks. We then turn our attention to two other frameworks, namely Structural Risk Minimization and Regularized Risk Minimization, which have been specifically designed to alleviate these drawbacks.

1.2.1 Empirical Risk Minimization (ERM)

The idea of ERM is to select the best hypothesis $h \in \mathcal{H}$ minimizing the empirical risk over the training set T. The objective is to solve the following optimization problem:

$$\arg\min_{h \in \mathcal{H}} \hat{L}_T(h). \tag{1.3}$$

This approach allows us to learn an hypothesis with a small empirical error. However we have no information about its true risk. What can happen is that the hypothesis is very good on the training set but do not generalize well to unseen examples, i.e. it has a big true risk despite its small empirical risk. This is not a desirable property as we recall that our goal is to learn an hypothesis with a small true risk.

The problem described above is called over-fitting. It often arises when the considered hypotheses are too complex for the problem at hand. Indeed they are more prone to noise fitting than simpler ones. To overcome this we can follow Occam's razor which says that *among a set of hypotheses able to explain a phenomenon, choosing the simplest one is better.* Hence the idea is to limit the complexity of the hypothesis class $\mathcal{H}$. However we also have to be careful to not limit the hypothesis class too much as it may lead to a situation where the empirical risk blows up. This second issue is called under-fitting. To sum up, the best case

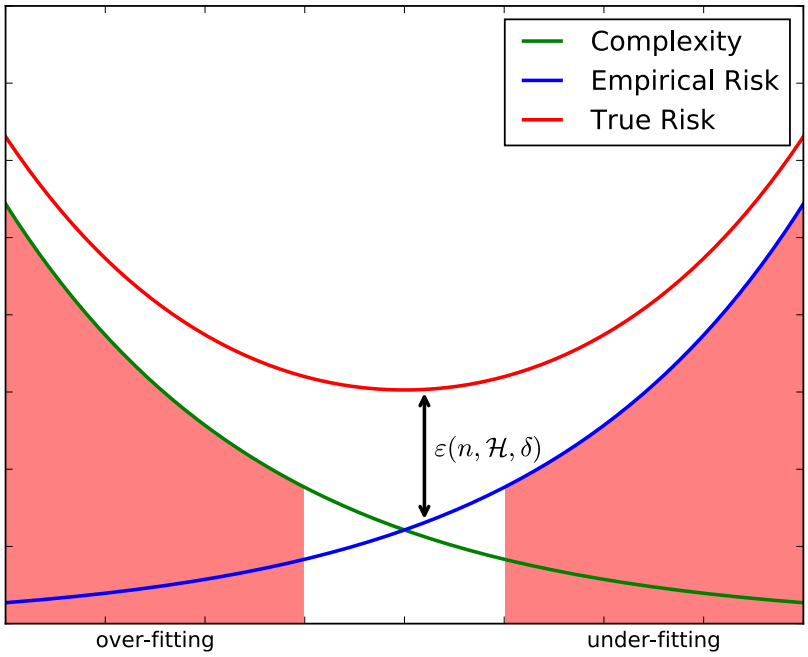


Figure 1.1: Illustration of the bias-variance trade-off problem. On the left of the plot there is a high risk of over-fitting while one the right there is a high risk of under-fitting.

scenario is to find the correct trade-off between a small empirical risk and a simple hypothesis. This trade-off is called the bias-variance trade-off and is illustrated in Figure 1.1.

In ERM the candidates are chosen uniformly from the space $\mathcal{H}$. It implies that all the hypotheses are considered to have the same complexity. Hence, to avoid over-fitting, $\mathcal{H}$ must be chosen very carefully. However this space is often defined beforehand and does not depend on the data. It implies that one has to resort to a costly trial and error procedure to select $\mathcal{H}$ in a correct way. It makes ERM hard to use in practice and opened the door for new frameworks that we present below.

1.2.2 Structural Risk Minimization (SRM)

In ERM the bias-variance trade-off is hard to satisfy since hypotheses are chosen uniformly from $\mathcal{H}$. In SRM instead of considering a single hypothesis space $\mathcal{H}$, we consider an infinite number of hypothesis spaces of increasing complexity such that $\mathcal{H}_1 \subseteq \mathcal{H}_2, \cdots$. The idea is then to solve the following optimization problem:

$$\underset{h \in \mathcal{H}_i, i \in \mathbb{N}}{\arg\min}\ \hat{L}_T(h) + \text{pen}(\mathcal{H}_i) \tag{1.4}$$

where $\text{pen}(\mathcal{H}_i)$ is a term penalizing the complexity of the hypothesis space $\mathcal{H}_i$. In this case we consider hypotheses by increasing complexity. We then select the one with the best trade-off

between empirical risk and complexity. In other words, the goal is to minimize the empirical risk while avoiding over-fitting by selecting the simplest hypothesis.

1.2.3 Regularized Risk Minimization (RRM)

RRM takes the idea of ordering the hypothesis in terms of their complexity one step further and considers the complexity of each hypothesis individually. The idea is to consider a single, large hypothesis space $\mathcal{H}$ and to solve the following optimization problem:

$$\arg\min_{h \in \mathcal{H}} \hat{L}_T(h) + \lambda \|h\| \tag{1.5}$$

where $\|h\|$ is a measure of the complexity of the hypothesis h. Furthermore λ is an hyper-parameter which controls the trade-off between low error and low complexity. Choosing the value of λ can be difficult in practice. However several heuristics have been proposed to cope with this problem. We can for example cite the leave-one-out or the cross-validation approaches.

Most of the algorithms presented in this thesis build upon the last framework presented in this section. As such in Section 1.4 we give a formal definition of a loss function and a regularization term. Furthermore we provide several examples which have been successfully used in the literature. Note that using a slight abuse of language we will often say that an optimization problem is an algorithm in itself since, in this thesis, we do not put much emphasis on the problem of developing efficient solvers. Implicitly we refer to an algorithm able to solve it. In the next section we propose to take a more theoretical point of view on the problem of learning an hypothesis with small true risk.

1.3 Deriving Generalization Guarantees

In the previous section we have presented empirical solutions to learn an hypothesis with small empirical risk and small true risk. In this section we consider a more theoretical point of view. We will show that, under some conditions on the algorithm used to learn the hypothesis, it is possible to derive what is called a generalization bound. The idea of these bounds is to show that the true risk of an hypothesis is upper bounded by its empirical risk plus a small quantity which usually depends on the complexity of the hypothesis and the number of examples in the training set. Furthermore these bounds build upon the PAC-Learning framework (Valiant, 1984) and as such are probabilistic bounds which hold true almost everywhere. For $\delta > 0$ they have the following form:

$$\Pr\Big(L_{\mathcal{T}}(h) \leq \hat{L}_T(h) + \varepsilon(n, \mathcal{H}, \delta)\Big) \geq 1 - \delta. \tag{1.6}$$

It means that deriving a probabilistic generalization bound boils down to showing that the probability that the true risk is upper bounded by the empirical risk plus a small quantity $\varepsilon(n, \mathcal{H}, \delta)$ is greater than $1 - \delta$. This is illustrated in Figure 1.1. The key point is then to

obtain the value of $\varepsilon(\mathcal{H}, n, \delta)$. It can be seen as a measure of the generalization ability of the learned hypothesis. As stated before this quantity should be small and depends on three elements:

- The number of examples n: as the number of examples increases the value of $\varepsilon(\mathcal{H}, n, \delta)$ should decrease. Furthermore a desirable property is to have $\lim_{n \to +\infty} \varepsilon(\mathcal{H}, n, \delta) = 0$. Indeed when we have access to all the examples $L_{\mathcal{T}}(h) = \hat{L}_T(h)$.

- The hypothesis class $\mathcal{H}$: to be more precise $\varepsilon(\mathcal{H}, n, \delta)$ depends on the complexity of the hypothesis. As hinted by Occam's razor, the value of $\varepsilon(\mathcal{H}, n, \delta)$ should increase if the hypothesis is more complex. Note that depending on the framework we either consider the complexity of the learned hypothesis h or the overall complexity of the hypothesis class $\mathcal{H}$.

- The probability δ: these bounds are generally based on concentration inequalities such as McDiarmid's inequality (McDiarmid, 1989) or Bennett's inequality (Bennett, 1962). They are probabilistic bounds which hold true with probability $1 - \delta$. The value of $\varepsilon(\mathcal{H}, n, \delta)$ increases when δ decreases. Indeed, if we want the bound to hold everywhere we have to take more particular cases into account which loosen the result.

Several frameworks have been proposed to derive generalization bounds. The main differences between these different frameworks is the concentration inequality considered and how they handle the complexity of the hypothesis class. In this thesis we consider two frameworks to derive generalization bounds, namely the uniform stability framework (Bousquet and Elisseeff, 2002b) and the Rademacher complexity framework (Bartlett and Mendelson, 2002). These two frameworks are presented below. Note that there is several other possible approaches that we will not detail here as they are less relevant to this thesis. We can for example cite the uniform convergence framework, the VC-dimension framework Vapnik and Chervonenkis (1971); Vapnik (1982) or the algorithmic robustness framework (Xu and Mannor, 2010, 2012). Also see Boucheron et al. (2004) for a survey on concentration inequalities and Langford (2005) for a general tutorial on prediction theory.

1.3.1 Uniform Stability

We first present the *Uniform Stability* framework introduced by Bousquet and Elisseeff (2002b). This framework is applicable to any algorithm which is uniformly stable, i.e. which respects the following definition:

Definition 1.3 (Uniform Stability (Bousquet and Elisseeff, 2002b, Definition 6))**.** *Let $T \sim \mathcal{D}_{\mathcal{T}}$ be a training set of size n and $\mathbf{z} \sim \mathcal{D}_{\mathcal{T}}$ be any example. Let T^i be the training set obtained by replacing example i in T by $\mathbf{z}$. Let $\mathcal{A}$ be an algorithm which returns hypothesis h_T when learning with the training set T and h_{T^i} when learning with the training set T^i. An algorithm*

$\mathcal{A}$ has uniform stability β with respect to its loss function l if the following holds:

$$\forall i \in \{1, \ldots, n\}, \sup_{\mathbf{z}' \sim \mathcal{D}_\mathcal{T}} \left| l\left(h_T, \mathbf{z}'\right) - l\left(h_{T^i}, \mathbf{z}'\right) \right| \leq \frac{\beta}{n}. \tag{1.7}$$

The idea is to say that an algorithm is uniformly stable if under small changes in the training set, the difference in the errors of the learned hypotheses is bounded. Furthermore the term on the r.h.s. should decrease as the number of examples increases. Note that this property should hold for all size n training sets and only the hypotheses effectively learned by the algorithm are considered, i.e. if some hypotheses in $\mathcal{H}$ are never learned by the algorithm under any training set then these hypotheses do not impact the result. As a consequence this framework has the nice property to focus on hypotheses that will really be learned by the considered algorithm. The value of β usually depends on the loss function and the regularization term. In Chapters 3, 4 and 5 we use this framework and show that the proposed algorithms are uniformly stable. Note that in their paper Bousquet and Elisseeff (2002b) consider several definitions for the notion of stability. However they show that the notion of uniform stability is the strongest one and that it implies all the others.

Using the McDiarmid's inequality (Theorem A.1) it is possible to show that a β-uniformly stable algorithm generalizes well:

Theorem 1.1 (Generalization bound (Bousquet and Elisseeff, 2002b, Theorem 12))**.** *Let $\mathcal{A}$ be an algorithm with uniform stability β with respect to a bounded loss function $0 \leq l \leq B$, for all $\mathbf{z} \in \mathcal{Z}$ and all sets T. Then given a randomly drawn sample T and given that h_T is the solution given by $\mathcal{A}$, for any $n \geq 1$, and any $\delta \in (0, 1)$, the following bound holds with probability at least $1 - \delta$:*

$$L_\mathcal{T}(h_T) \leq \hat{L}_T(h_T) + \frac{\beta}{n} + (2\beta + B)\sqrt{\frac{\ln(\frac{1}{\delta})}{2n}}. \tag{1.8}$$

The bounds derived using this framework converge in $\mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$. Note that this rate is standard for generalization bounds.

This framework has been shown to be applicable to a wide range of algorithms such as Support Vector Machines or Regularized Least Square Regression (Bousquet and Elisseeff, 2002b). Due to the proof techniques used to derive the bound, the main limitation of this framework lays in the kind of regularization terms that it can handle. For example Xu et al. (2012) have shown that algorithms based on sparsity inducing regularization terms are not stable.

1.3.2 Rademacher Complexity

We now switch our attention to the *Rademacher Complexity* framework introduced by Bartlett and Mendelson (2002). This framework is based on the notion of Rademacher complexity defined as follows:

Definition 1.4 (Rademacher Complexity (Shalev-Shwartz and Ben-David, 2014b, Equation (26.4)))**.** *Let $T \sim \mathcal{D}_{\mathcal{T}}$ and let $\mathcal{F}$ be a function space such that $f : \mathcal{X} \to \mathbb{R}$. Let $\boldsymbol{\sigma}$ be a vector of n Rademacher Variables, i.e. variables which can take a value of either 1 or -1 with probability $\frac{1}{2}$. The empirical Rademacher complexity is defined as follows:*

$$\hat{R}_n(\mathcal{F}) = \mathop{\mathbb{E}}_{\boldsymbol{\sigma}} \left[\sup_{f \in \mathcal{F}} \left| \frac{1}{n} \sum_{i=1}^{n} \sigma_i f(\mathbf{x}_i) \right| \middle| \mathbf{x}_1, \cdots, \mathbf{x}_n \right]. \tag{1.9}$$

The Rademacher complexity is then defined as:

$$R(\mathcal{F}) = \mathop{\mathbb{E}}_{T \sim \mathcal{D}_{\mathcal{T}}} \hat{R}_n(\mathcal{F}) \tag{1.10}$$

where the expectation is taken over size n sets.

The idea of empirical Rademacher complexity is to measure the capacity of the functions in $\mathcal{F}$ at fitting random noise. This noise is generated by the Rademacher variables and all the possibilities are considered through the expectation. Note that the supremum is considered over all the possible functions in $\mathcal{F}$. It implies that instead of considering only the hypotheses learned by the algorithm as for the uniform stability framework, the Rademacher complexity framework considers the complexity of the whole hypothesis class, i.e. some hypotheses which are never learned by the considered algorithm might impact the bound. Furthermore Rademacher complexity is defined in expectation with respect to all the training sets of size n and is not specific to the training set T considered during the learning process.

Using the McDiarmid's inequality (Theorem A.1) it is possible to derive a generalization bound based on the Rademacher complexity.

Theorem 1.2 (Generalization bound (Shalev-Shwartz and Ben-David, 2014b, Theorem 26.5))**.** *Let $\mathcal{A}$ be an algorithm with Rademacher Complexity $R(\mathcal{F})$ with respect to a bounded loss function $0 \leq l \leq B$. Note that $\mathcal{F} = \{f = l \circ h\}$ with $h \in \mathcal{H}$. Then, for any $n \geq 1$, any $\delta \in (0, 1)$ and any $h \in \mathcal{H}$, the following bound holds with probability at least $1 - \delta$ over the random draw of the sample T:*

$$L_{\mathcal{T}}(h) \leq \hat{L}_T(h) + 2R(\mathcal{F}) + B\sqrt{\frac{2\ln(\frac{2}{\delta})}{n}}. \tag{1.11}$$

On the one hand this bound holds for any hypothesis $h \in \mathcal{H}$ and for a wide range of regularization terms including several sparsity inducing ones. On the other hand the uniform stability based bound only holds for the hypothesis learned by the algorithm and for a limited number of regularization terms. The price paid by the Rademacher complexity framework to obtain such a behaviour is the convergence of the complexity related term. Indeed in the uniform stability framework this term was decreasing in $\mathcal{O}\left(\frac{1}{n}\right)$ while in the Rademacher complexity framework it can often be shown that $R(\mathcal{F}) \leq \mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$. Note that overall both bounds converge in $\mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$ due to the probabilistic term. We make use of the Rademacher complexity framework in Chapter 4.

In this section we presented two frameworks that can be used to derive generalization bounds. In the next section we formally define several notions used throughout this thesis. These include loss functions, regularization terms and metrics. These notions are accompanied by several illustrating examples.

1.4 Loss Functions, Regularization Terms and Metrics

The performance of regularized risk minimization primarily depends on the chosen loss function and regularization term. In this section we present a formal definition of these notions along with some examples. We also introduce the notion of metric as a way to compare learning examples. This is often a key component of machine learning methods.

1.4.1 Loss Functions and Regularization Terms

Loss functions and regularization terms are fundamental building blocks of regularized risk minimization algorithms. We start by presenting the formal definitions of what we consider as a loss function and a regularization term. Note that both of these definitions exhibit the property of *Hypothesis Ordering*. This is key to select the best hypothesis, i.e. an hypothesis with low error (with respect to the loss function) and as simple as possible (with respect to the regularization term). Hence given a training set, changing the loss or the regularization can lead one to learn different hypotheses. After the formal definitions we present several examples used by state of the art approaches.

Loss Function: Definition

Definition 1.5 (Loss function)**.** *Let $\mathcal{T}$ be a domain corresponding to the space $\mathcal{Z}$ equipped with the probability distribution $\mathcal{D}_{\mathcal{T}}$. Let $\mathcal{H}$ be an hypothesis space of candidates able to give a solution to the problem associated with the domain $\mathcal{T}$. A loss function is any function $l : \mathcal{H} \times \mathcal{Z} \to \mathbb{R}_+$ such that:*

1. $\forall h \in \mathcal{H}, \forall \mathbf{z} \in \mathcal{Z}, l(h, \mathbf{z}) \geq 0$ *(Non-negativity),*
2. $\forall h_1, h_2 \in \mathcal{H}, \forall \mathbf{z} \in \mathcal{Z}, l(h_1, \mathbf{z}) \leq l(h_2, \mathbf{z})$ *implies that h_1 gives a better prediction than h_2 on example $\mathbf{z}$ (Hypothesis ordering).*

A loss function can take its values in $[0, u]$ rather than $\mathbb{R}_+$. It is then said to be upper bounded or bounded.

Loss Function: Examples

Depending on the problem at hand different loss functions should be used. Here we propose to consider two different problems which have already been introduced before, namely classification and regression.

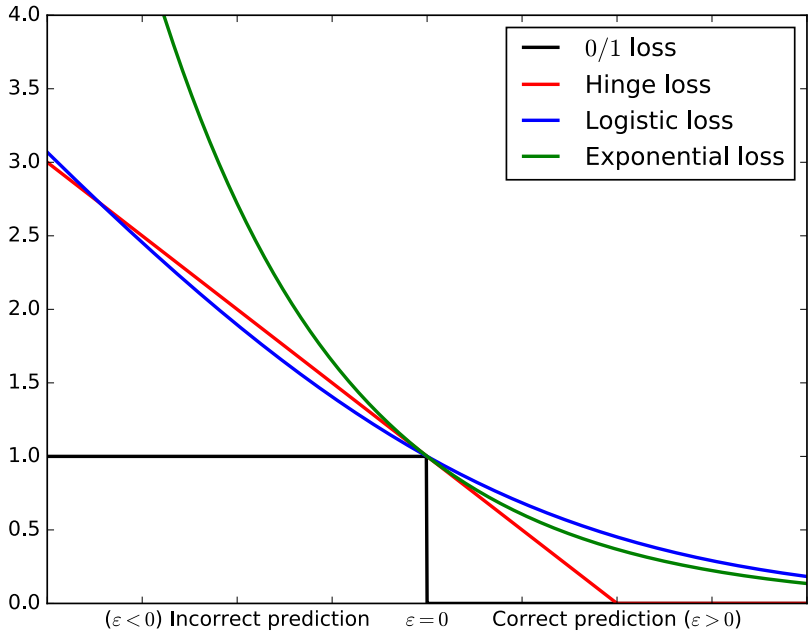


Figure 1.2: Graphical representation of commonly used loss functions for the classification problem.

We first propose several examples of loss functions mainly used in classification. These are depicted in Figure 1.2.

0/1 loss This loss is probably the most intuitive one as the idea is simply to count the number of errors of the hypothesis. This loss returns 1 if the hypothesis makes an incorrect prediction and 0 otherwise:

$$\forall h \in \mathcal{H}, \forall \mathbf{z} \in \mathcal{Z}, l\left(h, \mathbf{z}\right) = \begin{cases} 0 & \text{if } h(\mathbf{x}) = y, \\ 1 & \text{otherwise.} \end{cases} \tag{1.12}$$

The main drawback of this loss is that it is not convex and not differentiable everywhere. As such an optimization problem based on it is hard to solve and thus this loss is not used in practice. One solution is to use a surrogate loss. The idea is to upper-bound the 0/1 loss with a convex function which is easier to include in an optimization problem. To present several examples of surrogate loss functions we start by defining $\varepsilon \in \mathbb{R}$ as the degree of agreement between the prediction $h(\mathbf{x})$ and the ground truth y. The value of ε mainly depends on the confidence of the prediction, see Figure 1.2.

Hinge loss It is defined as follows:

$$\forall h \in \mathcal{H}, \forall \mathbf{z} \in \mathcal{Z}, l\left(h, \mathbf{z}\right) = \left[1 - \varepsilon\right]_+ = \max\left\{0, 1 - \varepsilon\right\} \tag{1.13}$$

This loss has, for example, been successfully used in Cortes and Vapnik (1995).

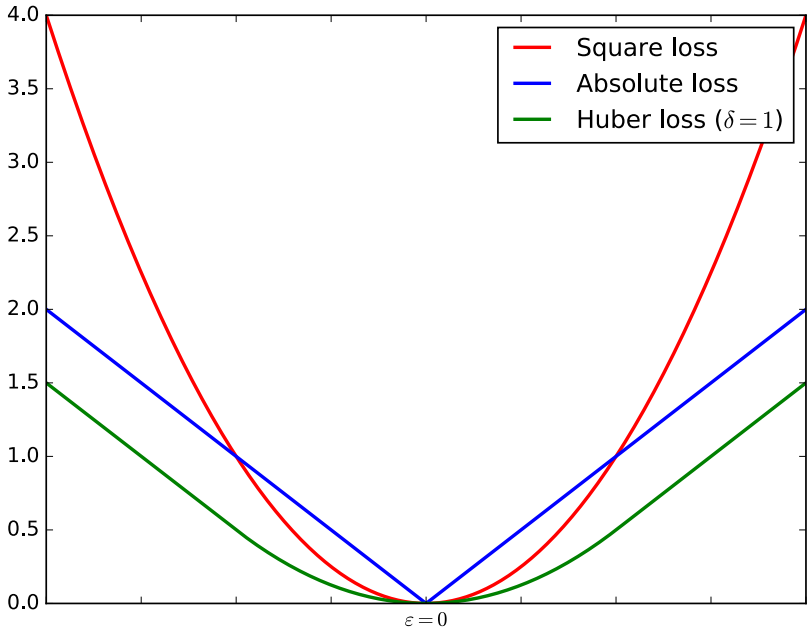


Figure 1.3: Graphical representation of commonly used loss functions for the regression problem.

Logistic loss It is defined as follows:

$$\forall h \in \mathcal{H}, \forall \mathbf{z} \in \mathcal{Z}, l\left(h, \mathbf{z}\right) = \frac{\log(1 + \exp(-\varepsilon))}{\log(2)} \tag{1.14}$$

This loss has, for example, been successfully used in Friedman et al. (2000).

Exponential loss It is defined as follows:

$$\forall h \in \mathcal{H}, \forall \mathbf{z} \in \mathcal{Z}, l\left(h, \mathbf{z}\right) = \exp(-\varepsilon) \tag{1.15}$$

This loss has, for example, been successfully used in Freund and Schapire (1997).

We also propose some loss functions which can be used in regression. These are depicted in Figure 1.3. Note that in this case the degree of agreement ε between the prediction and the ground truth is defined as the residual, i.e. $\varepsilon = h(\mathbf{x}) - y$.

Square loss It is defined as follows:

$$\forall h \in \mathcal{H}, \forall \mathbf{z} \in \mathcal{Z}, l\left(h, \mathbf{z}\right) = \varepsilon^2 \tag{1.16}$$

This loss has, for example, been successfully used in Tibshirani (1996).

Absolute loss It is defined as follows:

$$\forall h \in \mathcal{H}, \forall \mathbf{z} \in \mathcal{Z}, l\left(h, \mathbf{z}\right) = |\varepsilon| \tag{1.17}$$

A survey on the use of this loss can be found in Dielman (2005). Note that this loss is not differentiable everywhere and as such can be harder to use in an optimization problem.

Huber loss It is parametrized by δ and defined as follows:

$$\forall h \in \mathcal{H}, \forall \mathbf{z} \in \mathcal{Z}, l\left(h, \mathbf{z}\right) = \begin{cases} \frac{1}{2}\varepsilon^2 & \text{if } |\varepsilon| \leq \delta, \\ \delta\left(|\varepsilon| - \frac{1}{2}\delta\right) & \text{otherwise.} \end{cases} \tag{1.18}$$

This loss has been proposed by Huber (1964) and has been designed to be more robust to outliers than the square loss while still being differentiable everywhere.

We now turn our attention to regularization terms.

Regularization Term: Definition

Definition 1.6 (Regularization term)**.** *Let $\mathcal{H}$ be an hypothesis space. A regularization term is any function $\|\cdot\| : \mathcal{H} \to \mathbb{R}_+$ such that:*

1. $\forall h \in \mathcal{H}, \|h\| \geq 0$ *(Non-negativity),*
2. $\forall h_1, h_2 \in \mathcal{H}, \|h_1\| \leq \|h_2\|$ *implies that h_1 is less complex than h_2 (Hypothesis ordering).*

A regularization term can take its values in $[0, u]$ rather than $\mathbb{R}_+$. It is then said to be upper bounded or bounded. As our notation suggests, most of the time we choose the regularization term as a norm over the hypothesis space.

Definition 1.7 (Norm)**.** *Let $\mathcal{X} \subseteq \mathbb{R}^d$ be a d-dimensional vector space. A norm is any function $\|\cdot\| : \mathcal{X} \to \mathbb{R}_+$ such that:*

1. $\forall \mathbf{x} \in \mathcal{X}, \|\mathbf{x}\| \geq 0$ *(Non-negativity),*
2. $\forall \mathbf{x} \in \mathcal{X}, \|\mathbf{x}\| = 0 \Leftrightarrow \mathbf{x} = \mathbf{0}$ *where $\mathbf{0}$ is the zero vector (Separate points),*
3. $\forall \mathbf{x} \in \mathcal{X}, \forall a \in \mathbb{R}, \|a\mathbf{x}\| \leq |a| \, \|\mathbf{x}\|$ *(Absolute homogeneity),*
4. $\forall \mathbf{x}, \mathbf{x}' \in \mathcal{X}, \|\mathbf{x} + \mathbf{x}'\| \leq \|\mathbf{x}\| + \|\mathbf{x}'\|$ *(Triangle inequality).*

For the sake of simplicity we presented the definition of a norm with respect to a vector space. However it can be easily extended to the notion of metric or hypothesis as long as the different properties are respected. Furthermore a norm can take its values in $[0, u]$ rather than $\mathbb{R}_+$. It is then said to be upper bounded or bounded.

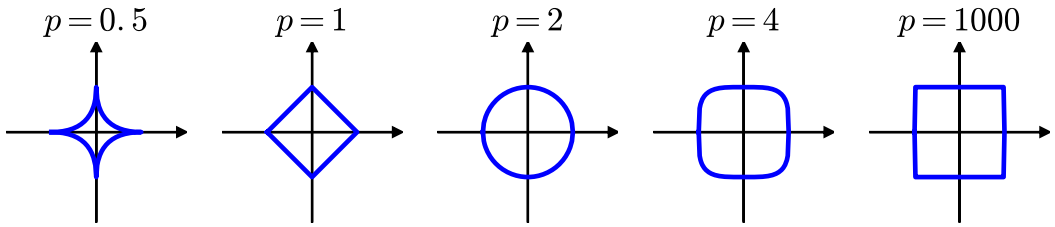


Figure 1.4: Two dimensional representation of the ℓ_p norm for different values of p. Each ball represents all the points with a norm of 1.

Regularization Term: Examples

As stated before regularization terms are often defined as norms. This is for example true when we consider that the hypothesis space is a vector space (as in linear classification or regression). Hence we now propose several examples of norms.

ℓ_p norms The ℓ_p norms are parametrised by a value p:

$$\forall \mathbf{x} \in \mathbb{R}^d, \|\mathbf{x}\|_p = \left(\sum_{i=1}^{d} |x_i|^p \right)^{\frac{1}{p}}. \tag{1.19}$$

For particular values of p we retrieve some well known norms depicted in Figure 1.4.

- **ℓ_1 norm**: If $p = 1$ it corresponds to the ℓ_1 norm. The ℓ_1 norm has been widely used for its sparsity inducing properties (Tibshirani, 1996). However, it is not differentiable everywhere and thus is harder to use in practice.
- **ℓ_2 norm**: If $p = 2$ it corresponds to the ℓ_2 norm. The ℓ_2 norm is strongly convex and differentiable everywhere. As such it has been used in many practical applications (Cortes and Vapnik, 1995). It tends to penalize large values.
- **Max norm or ℓ_∞ norm**: If $p = \infty$ it corresponds to max norm or ℓ_∞ norm.

Note that for $p < 1$ this norm is not convex and thus is hard to use in an optimization problem.

$\ell_{p,q}$ norm The $\ell_{p,q}$ norm is a generalization to matrices of the vectors ℓ_p norm. The idea is to apply a ℓ_p norm on each row of the matrix and then to apply the ℓ_q norm on the vector composed of the values obtained for each row:

$$\|\mathbf{M}\|_{p,q} = \left\| \left(\|\mathbf{M}(1,)\|_p, \cdots, \|\mathbf{M}(d,)\|_p \right) \right\|_q \tag{1.20}$$

When $p = q = 2$ we retrieve the Frobenius norm which is a natural extension to the matrix case of the ℓ_2 norm. Note that it is also possible to use this norm for vectors by separating the different features into several groups. For example, the $\ell_{2,1}$ norm has been used to induce sparsity constraints on groups of features (Yuan and Lin, 2006).

Schatten p norms The Schatten p norms are norms obtained by applying the ℓ_p norms to $\boldsymbol{\sigma}$ the vector of singular values of the matrix:

$$\|\mathbf{M}\|_p = \|\boldsymbol{\sigma}\|_p . \tag{1.21}$$

If $p = 2$ it corresponds to the Frobenius norm. If $p = \infty$ its the spectral norm and if $p = 1$ it corresponds to the nuclear or trace norm. The latter norm has been used for its capacity in producing low rank matrices (Wang et al., 2016).

We have presented a formal definition of loss functions and regularization terms along with several examples. We now turn our interest to the notion of metric.

1.4.2 Metrics

As mentioned before metrics are often a key component of machine learning algorithms as a way to compare examples. Before switching to the formal definition of what we consider as a metric we cite several well known algorithms which heavily rely on this notion:

- k-Nearest Neighbours (Cover and Hart, 1967): The idea behind this classification algorithm is to consider that examples which are close to each other share the same label. Hence to predict the label of a new example the algorithm considers its k nearest examples in the training set and chooses the majority label. Here the notion of metric is critical as one has to compare any new examples to the training examples.

- Support Vector Machines (Cortes and Vapnik, 1995): The idea behind this classification algorithm is to assume that there exist an high dimensional space in which the problem is linearly separable. This space is induced by a kernel which is a kind of metric.

- k-Means (Lloyd, 1982): The goal of this clustering algorithm is to partition the space into k regions whose members share a similar meaning. To achieve this, the idea is to randomly select k centres and to associate each example to its closest centre. The centres are then updated and the algorithm proceeds iteratively until convergence. The notion of closeness is controlled by a metric.

In this thesis we consider as a metric any similarity or dissimilarity which respect Definition 1.8. It includes but is not limited to the notion of Distance, Definition 1.9, and the notion of Kernel, Definition 1.10.

Metrics: Definitions

We start by presenting the general notion of similarity and dissimilarity that we consider in this thesis[1].

Definition 1.8 ((Dis)Similarity)**.** *Let $\mathcal{X} \subseteq \mathbb{R}^d$ be a d-dimensional vector space. A (dis)similarity is any pairwise function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. We say that a (dis)similarity is symmetric if $\forall \mathbf{x}, \mathbf{x}' \in \mathcal{X}, k(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}', \mathbf{x})$.*

A similarity should return a large positive value when two examples are similar and a large negative value otherwise. Conversely a dissimilarity should return a large negative value when two examples are similar and a large negative value otherwise. A (dis)similarity is said to be lower bounded, respectively upper bounded, if instead of taking its values in $\mathbb{R}$ it takes its values in an interval $[l, +\infty[$, respectively $]-\infty, u]$, such that $-\infty < l, u < +\infty$. When a (dis)similarity is lower and upper bounded, with $l \leq u$, we simply say that it is bounded.

A particular kind of lower bounded dissimilarity is a distance.

Definition 1.9 (Distance)**.** *Let $\mathcal{X} \subseteq \mathbb{R}^d$ be a d-dimensional vector space. A distance is a lower bounded dissimilarity function $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$ such that:*

1. $\forall \mathbf{x}, \mathbf{x}' \in \mathcal{X}, d(\mathbf{x}, \mathbf{x}') \geq 0$ *(Non-negativity),*
2. $\forall \mathbf{x}, \mathbf{x}' \in \mathcal{X}, d(\mathbf{x}, \mathbf{x}') = 0 \Leftrightarrow \mathbf{x} = \mathbf{x}'$ *(Identity of indiscernible),*
3. $\forall \mathbf{x}, \mathbf{x}' \in \mathcal{X}, d(\mathbf{x}, \mathbf{x}') = d(\mathbf{x}', \mathbf{x})$ *(Symmetry),*
4. $\forall \mathbf{x}, \mathbf{x}', \mathbf{x}'' \in \mathcal{X}, d(\mathbf{x}, \mathbf{x}') \leq d(\mathbf{x}, \mathbf{x}'') + d(\mathbf{x}'', \mathbf{x}')$ *(Triangle inequality).*

As in the case of (dis)similarities, a distance can take its values in $[0, u]$ and is then said to be upper bounded or bounded. Note the similarities between the definition of a distance and the definition of a norm. These two notions are closely related:

- Given a norm, the function $\mathbf{x}, \mathbf{x}' \mapsto \|\mathbf{x} - \mathbf{x}'\|$ is a distance.
- Given a distance, if it further respects the two following properties:
  1. $\forall \mathbf{x}, \mathbf{x}', \mathbf{x}'' \in \mathcal{X}, d(\mathbf{x} + \mathbf{x}'', \mathbf{x}' + \mathbf{x}'') = d(\mathbf{x}, \mathbf{x}')$,
  2. $\forall \mathbf{x}, \mathbf{x}' \in \mathcal{X}, \forall a \in \mathbb{R}, d(t\mathbf{x}, t\mathbf{x}') = |t|\, d(\mathbf{x}, \mathbf{x}')$

  then the function $\mathbf{x} \mapsto d(\mathbf{x}, \mathbf{0})$ is a norm.

The notion of kernel is a particular kind of similarity.

[1] Here we only consider metrics for feature vectors. However there also exist some metrics for structured data but this is beyond the scope of this thesis. We refer the interested reader to Bellet et al. (2015).

Definition 1.10 (Kernel)**.** *Let $\mathcal{X} \subseteq \mathbb{R}^d$ be a d-dimensional vector space and $\mathcal{K}$ be an Hilbert space. A symmetric similarity function $k(\cdot)$ is a kernel if there exists a function $\phi : \mathcal{X} \to \mathcal{K}$ such that:*

$$\forall \mathbf{x}, \mathbf{x}' \in \mathcal{X}, k\big(\mathbf{x}, \mathbf{x}'\big) = \big\langle \phi(\mathbf{x}) \,, \phi\big(\mathbf{x}'\big) \big\rangle \,. \tag{1.22}$$

Equivalently, $k(\cdot)$ is a kernel if it is positive semi-definite:

$$\forall \mathbf{x}_1, \cdots, \mathbf{x}_n \in \mathcal{X}, \forall c_1, \cdots, c_n \in \mathbb{R}, \sum_{i=1}^{n} \sum_{j=1}^{n} c_i c_j k(\mathbf{x}_i, \mathbf{x}_j) \geq 0. \tag{1.23}$$

Note that $\mathcal{K}$ can be very high dimensional or even infinite. In this case $\phi(\cdot)$ is intractable. However it is still possible to compute the value of the kernel through the expression of $k(\cdot)$. This is called the kernel trick[2]. This trick has for example been successfully used in Cortes and Vapnik (1995); Schölkopf et al. (1997).

Metrics: Examples

We give several examples of well-known metrics.

Minkowski distances The Minkowski distances is a family of distances induced by the ℓ_p norms and as such parametrised by a value p:

$$d\big(\mathbf{x}, \mathbf{x}'\big) = \left(\sum_{i=1}^{d} \big| x_i - x'_i \big|^p \right)^{\frac{1}{p}} = \big\| \mathbf{x} - \mathbf{x}' \big\|_p \,. \tag{1.24}$$

For particular values of p we retrieve some well known distances.

- **Manhattan distance**: If $p = 1$ it corresponds to the Manhattan distance induced by the ℓ_1 norm.
- **Euclidean distance**: If $p = 2$ it corresponds to the Euclidean distance induced by the ℓ_2 norm.
- **Chebyshev distance**: If $p = \infty$ it corresponds to the Chebyshev distance induced by the ℓ_∞ norm or max norm.

Mahalanobis distances The Mahalanobis distances is a family of distances parametrised by a matrix $\mathbf{M}$ such that:

$$d\big(\mathbf{x}, \mathbf{x}'\big) = \sqrt{(\mathbf{x} - \mathbf{x}')^T \mathbf{M} (\mathbf{x} - \mathbf{x}')}. \tag{1.25}$$

[2]Note that even if the feature map is tractable it is most of the times more interesting to compute the value of the kernel through the expression of $k(\cdot)$.

To obtain a proper distance, the matrix $\mathbf{M}$ has to be positive definite. If the matrix is positive semi-definite (PSD)[3] then it is a pseudo distance, i.e. the constraint on the identity of indiscernibles is relaxed and it is only required that $\forall \mathbf{x} \in \mathcal{X}, d(\mathbf{x}, \mathbf{x}) = 0$. Note that if $\mathbf{M} = \mathbf{I}$ the identity matrix, it corresponds to the Euclidean distance. In its original definition (Mahalanobis, 1936) the Mahalanobis distance was using the inverse variance-covariance matrix of the examples, i.e. $\mathbf{M} = \mathbf{\Sigma}^{-1}$. The intuition behind the Mahalanobis distance is to reweight the features of the examples. As such using a Cholesky decomposition such that $\mathbf{M} = \mathbf{L}^T\mathbf{L}$, one can see that the Mahalanobis distance corresponds to the Euclidean distance in a space linearly dependent on $\mathcal{X}$.

Bilinear similarities The bilinear similarities is a family of similarities parametrised by a matrix $\mathbf{M}$ and which is strongly related to the dot product:

$$k\big(\mathbf{x}, \mathbf{x}'\big) = \big\langle \mathbf{x}, \mathbf{M}\mathbf{x}' \big\rangle = \big\langle \mathbf{M}^T\mathbf{x}, \mathbf{x}' \big\rangle \tag{1.26}$$

If $\mathbf{M} = \mathbf{I}$ it corresponds to the dot product in the original space. Similarly if $\mathbf{M} = \frac{1}{\|\mathbf{x}\|_2\|\mathbf{x}'\|_2}\mathbf{I}$ it corresponds to the cosine similarity. While in general there is no constraints on $\mathbf{M}$, we can choose it to be positive semi-definite and, using a Cholesky decomposition, one can see that it corresponds to the dot product in a new space linearly dependent on $\mathcal{X}$.

In Chapter 2 we propose a review of several metric learning methods whose goal is to learn the parameters of $\mathbf{M}$, either for the Mahalanobis distance or the Bilinear similarity.

Kernels Kernels are defined with respect to a function $k(\cdot)$ and sometimes it is possible to explicitly compute the feature map $\phi(\cdot)$:

- **Linear kernel**: It corresponds to the dot product in the original space

$$\forall \mathbf{x}, \mathbf{x}' \in \mathcal{X}, k\big(\mathbf{x}, \mathbf{x}'\big) = \mathbf{x}^T\mathbf{x}' = \big\langle \mathbf{x}, \mathbf{x}' \big\rangle . \tag{1.27}$$

- **Polynomial kernel**: It is parametrized by its order p and a bias c

$$\forall \mathbf{x}, \mathbf{x}' \in \mathcal{X}, k\big(\mathbf{x}, \mathbf{x}'\big) = \big(\mathbf{x}^T\mathbf{x}' + c\big)^p . \tag{1.28}$$

  It is possible to compute the feature map explicitly. For example, for two dimensional vectors and $p = 2$ each example is implicitly mapped to a 6 dimensional vector:

$$\phi(\mathbf{x}) = \begin{pmatrix} x_1^2 & x_2^2 & \sqrt{2}x_1x_2 & \sqrt{2c}x_1 & \sqrt{2c}x_2 & c \end{pmatrix}^T . \tag{1.29}$$

  Note that small values of p are often preferred for this kernel as it becomes numerically unstable when p tends to infinity.

[3] To denote the fact that a matrix is positive semi-definite we interchangeably use the notation $\mathbf{M} \succeq 0$ or $\mathbf{M} \in \mathbb{S}_+^{d\times d}$ where d is the dimension of the matrix.

- **Gaussian kernel**: It is parametrized by its width σ and the feature map is infinite dimensional and thus intractable.

$$\forall \mathbf{x}, \mathbf{x}' \in \mathcal{X}, k\big(\mathbf{x}, \mathbf{x}'\big) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|_2^2}{2\sigma^2}\right). \tag{1.30}$$

  When using the Gaussian kernel if one has access to a training set T of n examples a standard heuristic is to set σ to the mean of all the euclidean distances between the examples (Kar and Jain, 2011).

1.5 Other Notions

In this section we propose to consider other notions that will be used throughout the thesis but do not fall in any of the previous sections. Hence we present the nearest neighbour classifier and the domain adaptation setting. We use the former in our experiments as the classifier making use of our metrics while we evaluate two of our contributions in the latter.

1.5.1 Nearest Neighbours Classifier

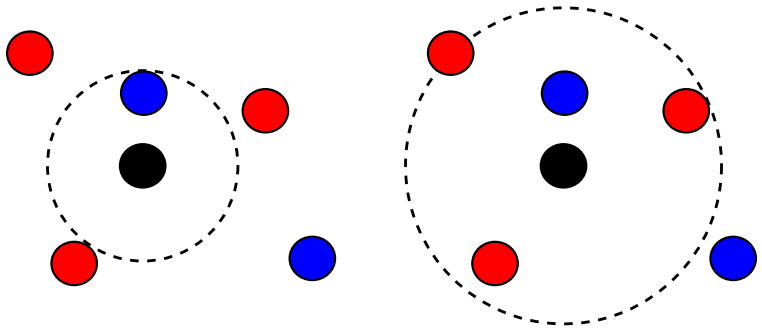
As mentioned in the introduction metric learning algorithms are often used as a preprocessing step to improve the performance of another algorithm. In this thesis we propose to consider the nearest neighbour classifier (Cover and Hart, 1967) as this subsequent approach. It is probably one of the most intuitive method in classification. The idea stems from the saying *birds of a feather flock together*, i.e. if two examples are close to each other they probably share the same label. To formally present this approach we consider that we want to classify $\mathbf{z} \sim \mathcal{D}_{\mathcal{T}}$ using the training set T and a measure of closeness between the examples under the form of a distance d.

1 nearest neighbour (1-NN) The idea behind the 1-NN classifier is to predict the class $\hat{y}$ for $\mathbf{z}$ as $\hat{y} = y_i$ where $\mathbf{z}_i$ is the closest example of $\mathbf{z}$ in T, i.e. the example $\mathbf{z}_i$ satisfying:

$$\mathbf{z}_i = \underset{\mathbf{z}' \in T}{\arg\min}\, d\big(\mathbf{z}, \mathbf{z}'\big). \tag{1.31}$$

k nearest neighbours (k-NN) The idea behind the k-NN classifier is that instead of only considering the closest example of $\mathbf{z}$ as for the 1-NN, one can select the k closest examples and set $\hat{y}$ as the majority class among these k nearest neighbours.

The nearest neighbours algorithm is illustrated in Figure 1.5. Note that here we considered that the closeness between the examples is determined by a distance. However this algorithm can be used with any metric and, in particular, learned ones.



(a) 1-NN. (b) 3-NN.

Figure 1.5: The goal is to classify the black point. The 1-NN algorithm,Figure 1.5(a), selects the closest example and classify the point as blue. The 3-NN classifier,Figure 1.5(b), chooses the majority class among the 3 closest examples and classifies the point as red.

1.5.2 Domain Adaptation Setting

Throughout this chapter we considered a supervised learning setting. If most of our contributions in this thesis fall into this first setting, we will also show that two of them are well suited to solve domain adaptation problems (See Chapters 4 and 6). In this subsection we propose to quickly introduce this setting.

Domain adaptation is a special case of transfer learning (Pan and Yang, 2010) where the goal is to adapt a model learned on a source domain to a target domain. Formally we consider that we have access to two domains, the source domain $\mathcal{S}$ defined as the space $\mathcal{Z}^s = \mathcal{X}^s \times \mathcal{Y}^s$ equipped with the probability distribution $\mathcal{D}_\mathcal{S}$ and the target domain $\mathcal{T}$ defined as the space $\mathcal{Z}^t = \mathcal{X}^t \times \mathcal{Y}^t$ equipped with the probability distribution $\mathcal{D}_\mathcal{T}$. These two domains are considered to be different but related. Hence the tasks associated with the two domains are the same but there is a sort of shift between the two distributions. For the adaptation to be possible we further assume that this shift is not prohibitively large. As an illustrative example we consider the Office-Caltech dataset (Gong et al., 2012) which is used as a benchmark in the domain adaptation community. The task consists in classifying the images of 10 kind of objects. The shift comes from the fact that the pictures come from 4 different domains:

- Amazon: the objects are presented on a white background,
- DSLR: the pictures are taken in an office environment with a high-end camera,
- Webcam: the pictures are taken in an office environment with a low resolution webcam,
- Caltech: the pictures of the objects come from the Caltech256 dataset (Griffin et al., 2007).

It defines 12 different tasks where the domain are paired and alternatively used as the source and the target.

In domain adaptation we consider that we have access to two training sets. The first one, S, is labelled and comes from the source domain[4] while the second one, T, comes from the target domain. We then consider two different settings:

- unsupervised domain adaptation: there is no supervision in T,
- semi-supervised domain adaptation: only a small amount of examples are labelled in T.

When solving a domain adaptation problem the goal is to estimate and overcome the shift between the distributions (Ben-David et al., 2010). To this extent several different strategies have been proposed in the literature. Among these we can cite for example the reweighting approaches where the idea is to put more emphasis on the source examples which are mixed with the target examples in the input space (See e.g. Mansour et al. (2009)). We can also cite iterative approaches where the idea is to learn a classifier on the source domain, label the target domain and replace some examples in the source domain by the newly labelled target examples. This process is repeated several times until a convergence criterion is met (See e.g. Bruzzone and Marconcini (2010)). As a last example another strategy consists in learning a common representation space for the source and the target where the shift between the two domains does not exist (See e.g. Gong et al. (2012); Hoffman et al. (2013)). This last strategy is often the motivating idea behind the metric learning methods interested in addressing the domain adaptation problem (See Section 2.5).

1.6 Conclusion

In this chapter we presented several fundamental notions used throughout this thesis. We started by introducing the risk minimization framework which will be used, in its regularized form, in all of our contributions. Then we addressed the problem of deriving generalization bounds. We presented two frameworks respectively based on the uniform stability principle and the notion of Rademacher complexity. Next we proposed a formal definition and some well known examples of the notion of loss function and regularization term. We also clarified the notion of metric as we consider it in this thesis. Lastly we introduced the nearest neighbour algorithm and the domain adaptation setting which will be used to assess the performance of several of our contributions.

In the next chapter we propose a non exhaustive review of the field of metric learning by answering four fundamental questions on the problem.

[4]Sometimes we also consider that we have access to a second non labelled training set from the source domain.

Chapter 2

Metric Learning

Abstract

In this chapter we propose a non exhaustive review of the field of metric learning. In particular we present several methods which are relevant in the context of this thesis. It notably corresponds to approaches that learn the same kind of metrics as we do, consider a similar way to perform the learning step, derive the same kind of generalization bounds or learn a metric to solve the same kind of task.

2.1 Introduction

As mentioned before the idea behind metric learning is to automatically learn, from the data, a metric adapted to the task at hand. This chapter is a non exhaustive review of this field as we put the focus on the most relevant methods for this thesis. We propose to explore the different existing approaches by answering four basic questions on metric learning.

- **What kind of metrics is it possible to learn?** We will see that most of the methods are interested in learning either a Mahalanobis distance or a bilinear similarity. The most common approaches to include some non linearity in the process are to learn multiple metrics across the space or to learn a linear metric in a kernel induced space.

- **How are the metrics effectively learned?** We will see that optimization problems in batch learning setting are widely used in metric learning and that approaches mainly varies in function of the kind of constraints used, the loss function considered and the regularization term. Nevertheless several approaches also proposed to learn a metric in an online fashion.

- **Are there any theoretical guarantees on the learned metrics?** We presented two frameworks used to derive generalization bounds in Chapter 1. We will see that these can be extended to the metric learning setting. Furthermore it is sometimes possible to evaluate the impact of a metric on the subsequent algorithm.

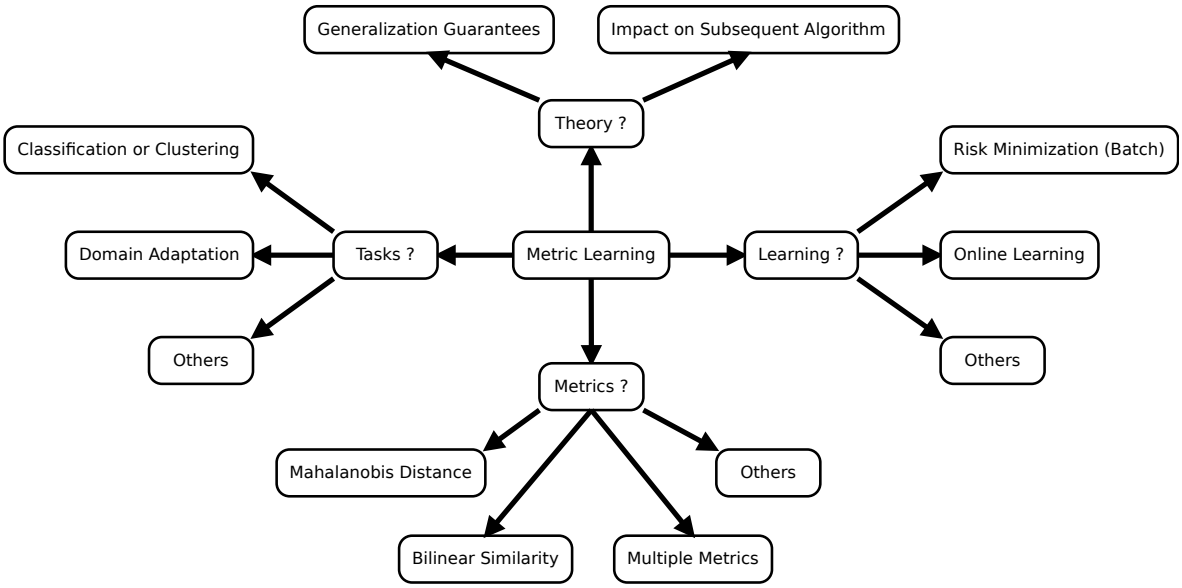


Figure 2.1: Metric learning in four questions.

- **In which tasks are the metrics used?** We will see that many approaches are interested in solving classification or semi-supervised clustering problems. However some works also considered different tasks such as image retrieval, face recognition or domain adaptation.

In Chapter 1 we were mainly focused on supervised learning problems. In this chapter, and unless stated otherwise, we consider the setting consisting in learning a metric for a classification problem. Formally we consider a domain $\mathcal{T}$ which corresponds to the space $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ equipped with the probability distribution $\mathcal{D}_{\mathcal{T}}$. We further consider that $\mathcal{X} \subseteq \mathbb{R}^d$, i.e. we are working with real valued vectors, and that we only have access to a training set $T = \{\mathbf{z}_i = (\mathbf{x}_i, y_i)\}_{i=1}^n$ of n training examples.

In Figure 2.1 we propose a diagram summarizing the outline of this review. In Section 2.2 we present several works whose goal is to learn a specific metric. Next in Section 2.3 we review different learning procedures applicable to metric learning. In Section 2.4 we consider the problem of deriving theoretical guarantees. Finally in Section 2.5 we focus on the applications making use of the learned metrics before concluding in Section 2.6

2.2 Metrics

Several metrics have been considered in the field. We present a short description of the most popular ones here.

2.2.1 Mahalanobis Distance

In their pioneering work Xing et al. (2002) propose to learn the parameter matrix $\mathbf{M}$ of a Mahalanobis distance. Popularized by Large Margin Nearest Neighbour (LMNN) (Weinberger et al., 2005) and Information Theoretic Metric Learning (ITML) (Davis et al., 2007), it is probably the most studied metric in the community. We presented it in Section 1.4 but we recall it here for the sake of readability:

$$d_{\mathbf{M}}(\mathbf{x}, \mathbf{x}') = \sqrt{(\mathbf{x} - \mathbf{x}')^T \mathbf{M} (\mathbf{x} - \mathbf{x}')} \text{ with } \mathbf{x}, \mathbf{x}' \in \mathcal{X}, \mathbf{M} \in \mathbb{S}_+^{d \times d}. \tag{2.1}$$

Note that to avoid the difficulties linked to the use of a square root, i.e. it is concave and only defined on $\mathbb{R}_+$, a lot of approaches have focused on learning the quadratic version of $d_{\mathbf{M}}$, denoted $d^2_{\mathbf{M}}$.

To obtain a proper distance the matrix $\mathbf{M}$ has to be positive semi-definite. This constraint can be hard to satisfy in practice as it often requires some costly projections on the positive semi-definite cone[1] (Jin et al., 2009). However this constraints also provides a nice interpretation of the metric. Indeed using a Cholesky decomposition one can write $\mathbf{M} = \mathbf{L}^T \mathbf{L}$ with $\mathbf{L} \in \mathbb{R}^{d' \times d}$. It implies that the Mahalanobis distance is the standard euclidean distance in a new space linearly dependent on $\mathcal{X}$:

$$d_{\mathbf{L}}(\mathbf{x}, \mathbf{x}') = \sqrt{(\mathbf{L}\mathbf{x} - \mathbf{L}\mathbf{x}')^T (\mathbf{L}\mathbf{x} - \mathbf{L}\mathbf{x}')} \text{ with } \mathbf{x}, \mathbf{x}' \in \mathcal{X}, \mathbf{L} \in \mathbb{R}^{d' \times d}. \tag{2.2}$$

Following this idea and to avoid the positive semi-definite constraint on $\mathbf{M}$ some approaches propose to directly learn the matrix $\mathbf{L}$ (Goldberger et al., 2004).

Another appealing property of the matrix $\mathbf{M}$ stemming from its positive semi-definiteness is that it can be written as a combination of rank 1 matrices:

$$\mathbf{M} = \sum_{i=1}^{k} \mathbf{u}_i \mathbf{u}_i^T \tag{2.3}$$

with the $\mathbf{u}_i \in \mathbb{R}^d$ are linearly independent vectors[2] and k is the rank of $\mathbf{M}$. Using this property several approaches propose to learn either a weighted combination of given rank 1 matrices (Shi et al., 2014) or the matrices themselves (Shen et al., 2009, 2012).

Depending on the form of the matrix $\mathbf{M}$ the Mahalanobis distance can have some appealing properties. For example if this matrix is diagonal the distance can be seen as a reweigthing of the input features (Xing et al., 2002). Similarly when $\mathbf{M}$ is low rank ($k < d$) then in the decomposition presented above the matrix $\mathbf{L}$ is rectangular with $d' < d$. It implies that the examples are projected in a lower dimensional space, i.e. it is equivalent to performing some dimensionality reduction on the data. Following this idea some approaches have then been interested in learning low rank matrices using some sparsity inducing norms such as the

[1] A projection onto the positive semi-definite cone requires an eigenvalue decomposition whose computational cost is roughly in $\mathcal{O}(d^3)$ making it intractable when d becomes large.

[2] These vectors can for example be the eigenvectors of the matrix times the square root of the corresponding eigenvalue.

trace norm (Ying et al., 2009), the capped trace norm (Huo et al., 2016) or a Fantope based norm (Law et al., 2014).

2.2.2 Bilinear Similarity

Apart from the Mahalanobis distance, the second most popular metric is probably the bilinear similarity which is also parametrized by a matrix $\mathbf{M}$. For example Qamar et al. (2008) consider the following similarity:

$$k_{\mathbf{M}}(\mathbf{x}, \mathbf{x}') = \frac{\mathbf{x}^T \mathbf{M} \mathbf{x}'}{N(\mathbf{x}, \mathbf{x}')} \text{ with } \mathbf{x}, \mathbf{x}' \in \mathcal{X}, \mathbf{M} \in \mathbb{R}^{d \times d} \tag{2.4}$$

where $\mathbf{M}$ can be either diagonal, symmetric or simply a square matrix and $N(\mathbf{x}, \mathbf{x}')$ is a normalization parameter.

Following this idea Qamar and Gaussier (2009) propose to use a generalized cosine similarity:

$$k_{\mathbf{M}}(\mathbf{x}, \mathbf{x}') = \frac{\mathbf{x}^T \mathbf{M} \mathbf{x}'}{\sqrt{\mathbf{x}^T \mathbf{M} \mathbf{x}} \sqrt{\mathbf{x}'^T \mathbf{M} \mathbf{x}'}} \text{ with } \mathbf{x}, \mathbf{x}' \in \mathcal{X}, \mathbf{M} \in \mathbb{S}_+^{d \times d}. \tag{2.5}$$

However the positive semi-definite constraint can be too restrictive in practice.

These two similarities can be seen as particular forms of the bilinear similarity presented in Section 1.4 and recalled here for the sake of readability:

$$k_{\mathbf{M}}(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{M} \mathbf{x}' \text{ with } \mathbf{x}, \mathbf{x}' \in \mathcal{X}, \mathbf{M} \in \mathbb{R}_+^{d \times d}. \tag{2.6}$$

This more general form has for example been used in Chechik et al. (2009, 2010); Kulis et al. (2011); Bellet et al. (2012).

2.2.3 Multiple Metrics

The two metrics presented above are linearly dependent on the input space. However it is sometimes not sufficient to capture the idiosyncrasies of the data. Hence learning a non linear metric becomes necessary. One possible approach is then to learn multiple linear metrics across the space. One basic strategy is local metric learning which consists in dividing the input space in K clusters $C_1, \ldots, C_K$ and to learn one metric in each cluster. For example one can obtain K Mahalanobis distances $d_{\mathbf{M}_1}, \ldots, d_{\mathbf{M}_K}$. The distance can then be computed between two examples $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$ as:

$$d_{\mathbf{M}}(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^{K} \mathbf{w}_{\mathbf{x},\mathbf{x}'}(i) d_{\mathbf{M}_i}(\mathbf{x}, \mathbf{x}') \tag{2.7}$$

where $\mathbf{w}_{\mathbf{x},\mathbf{x}'}(i)$ is the weight of the distance $d_{\mathbf{M}_i}$ in the combination when considering the two examples $\mathbf{x}$ and $\mathbf{x}'$. Hence the methods interested in learning multiple metrics mainly vary on the way they cluster the input space, the metric learned and the way they choose the weights of the metrics with respect to the examples.

- In Multi-Metric Large Margin Nearest Neighbour (MM-LMNN) Weinberger and Saul (2008) propose to either set one partition for each class or to use the k-means algorithm. In each partition they propose to use LMNN (Weinberger et al., 2005) to learn a Mahalanobis distance. Finally the distance $d_{\mathbf{M}}(\mathbf{x}, \mathbf{x}')$ between two examples only depends on the cluster in which $\mathbf{x}'$ falls. It implies that the global distance is not symmetric if $\mathbf{x}$ and $\mathbf{x}'$ do not fall in the same cluster.

- Semerci and Alpaydın (2013) propose to learn a Mixture of LMNN (MoLMNN) by alternatively learning the partition of the space and the transformation matrix of a Mahalanobis distance. Furthermore they propose to use a soft partitioning of the space where the transformation of one example depends on several local transformations.

- In Large Margin Local Metric Learning (LMLML) Bohné et al. (2014) propose to learn one Mahalanobis distance for each of the K components of a Gaussian mixture model. For two examples $\mathbf{x}$ and $\mathbf{x}'$ the weight of each metric depends on the degree of membership of the two examples to each component.

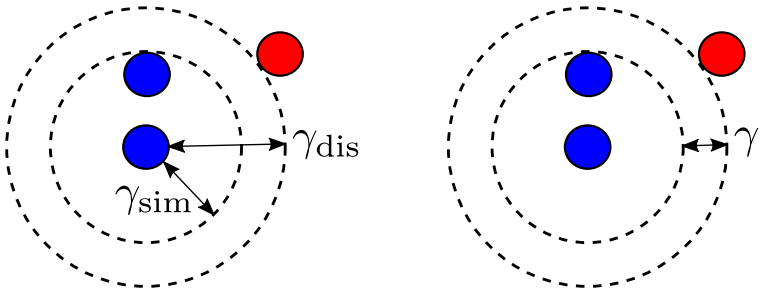
- In Parametric Local Metric Learning (PLML) Wang et al. (2012) propose to select anchor points defined as the means of clusters constructed by the k-means algorithm. Then they express each example in the training set as a weighted combination of the anchor points and they use these weights to learn one basis metric for each anchor point. Note that the global metric is not symmetric.

- Instead of partitioning the input space Chang and Yeung (2004, 2007) propose to learn one linear transformation for each example but to compute their effective transformations as a learned weighted combination of the transformations of their neighbours.

2.2.4 Other Non Linear Metrics

To include some non linearity in the model some approaches propose to consider intrinsically non linear metrics. For example Kedem et al. (2012) proposed to build upon LMNN (Weinberger et al., 2005) to learn two new metrics. The first one, called χ^2-LMNN, is well suited for histogram data. The second one is called GB-LMNN and is based on Gradient Boosted regression trees. As a last example of a method learning a non linear metric Xiong et al. (2012a) propose the Random Forest Distance (RFD). This is a local metric learning method based on random forests classifiers. The idea is to learn a classifier able to predict if two examples are similar or dissimilar, i.e. to predict 1 if two examples are similar and 0 otherwise. Another possible approach to include some non linearity, used for example in Davis et al. (2007), is to consider learning a linear metrics in a space non linearly dependent on the input space using for example a kernel.

In this thesis we consider the problem of learning Mahalanobis distances in Chapters 4, 5 and 6. In Chapter 4 we also consider learning bilinear similarities while in Chapter 3 we propose to learn multiple Mahalanobis distances.



(a) Pair based constraints. (b) Triplet based constraints.

Figure 2.2: Illustration of the notion of margin for pair based constraints 2.2(a) and triplet based constraints 2.2(b).

2.3 Learning Approaches

A classification problem becomes easier to solve when, in the data, the intra class variance is low and the inter class variance is high. In other words it is desirable to have all the examples of the same class close to each other and all the examples of different classes far from each other. Building upon this idea most of the existing works in metric learning try to learn the best metric such that the aforementioned constraints are respected.

2.3.1 Pair Based Constraints

Given a labelled training set T, a first approach consists in considering the examples by pairs and defining similarity and dissimilarity constraints as follows:

- A set of pairs of similar examples: $P_{\text{sim}} = \{(\mathbf{z}, \mathbf{z}') \text{ s.t. } \mathbf{z}, \mathbf{z}' \in T, y = y'\}$,
- A set of pairs of dissimilar examples: $P_{\text{dis}} = \{(\mathbf{z}, \mathbf{z}') \text{ s.t. } \mathbf{z}, \mathbf{z}' \in T, y \neq y'\}$.

Alternatively we can consider a single set of pairs of examples:

- $P_{\text{pair}} = \left\{(\mathbf{z}, \mathbf{z}', \delta_{yy'}) \text{ s.t. } \mathbf{z}, \mathbf{z}' \in T, \delta_{yy'} = 1 \text{ if } y = y', \delta_{yy'} = -1 \text{ if } y \neq y'\right\}$.

A good metric should be able to bring closer to each other all the similar examples while pushing far away all the dissimilar ones. For example, using empirical risk minimization (Section 1.2), learning a Mahalanobis distance could be done by solving one of the following two optimization problems:

$$\begin{aligned}
&\underset{\mathbf{M}\in\mathbb{S}_+^{d\times d}}{\arg\min} \sum_{(\mathbf{z},\mathbf{z}')\in P_{\text{sim}}} \mathbb{1}_{d_{\mathbf{M}}(\mathbf{x},\mathbf{x}')\geq\gamma_{yy'}} + \sum_{(\mathbf{z},\mathbf{z}')\in P_{\text{dis}}} \mathbb{1}_{d_{\mathbf{M}}(\mathbf{x},\mathbf{x}')\leq\gamma_{yy'}} + \lambda \left\|\mathbf{M}\right\| \\
&\underset{\mathbf{M}\in\mathbb{S}_+^{d\times d}}{\arg\min} \sum_{(\mathbf{z},\mathbf{z}',\delta_{yy'})\in P_{\text{pair}}} \mathbb{1}_{\delta_{yy'} d_{\mathbf{M}}(\mathbf{x},\mathbf{x}')\geq\delta_{yy'}\gamma_{yy'}} + \lambda \left\|\mathbf{M}\right\|
\end{aligned}$$

where the margin $\gamma_{yy'}$ is a measure of closeness between the examples and $\mathbb{1}$ is the indicator function whose value is 1 if the condition is true and 0 otherwise[3]. The idea is that similar examples should be at a distance lower than $\gamma_{yy'}$ while dissimilar examples should be at a distance greater than $\gamma_{yy'}$ (See Figure 2.2(a)). Note that $\gamma_{yy'}$ depends on the examples and can thus have a different value for each pair. In practice we often fix a value γ_{sim} when $y = y'$ and a value γ_{dis} when $y \neq y'$. Many approaches in metric learning are based on a similar idea, i.e. they try to obtain a metric that approximates these constraints. However the optimization problem presented here is non convex and non differentiable and is thus hard to optimize in practice. Most of the approaches then consider surrogate losses (Section 1.4) which are easier to handle. They also make use of various regularization terms to enforce different properties on the metrics.

- In their pioneering work (Xing et al., 2002) propose to learn a Mahalanobis distance $d_{\mathbf{M}}$ by bringing similar examples close to each other while keeping dissimilar examples reasonably far away. They use a gradient descent based approach with iterative projections on the constraints to solve the following optimization problem:

$$\underset{\mathbf{M}\in\mathbb{S}_+^{d\times d}}{\arg\min} \sum_{(\mathbf{x},\mathbf{x}')\in P_{\text{sim}}} d^2_{\mathbf{M}}\big(\mathbf{x},\mathbf{x}'\big) \tag{2.8}$$

$$\text{s.t.} \sum_{(\mathbf{x},\mathbf{x}')\in P_{\text{dis}}} d_{\mathbf{M}}\big(\mathbf{x},\mathbf{x}'\big) \geq 1. \tag{2.9}$$

  Here the margin between similar examples is implicitly set to 0 while the margin between dissimilar examples is set to 1.

- Goldberger et al. (2004) proposed Neighbourhood Components Analysis (NCA). It is a method based on a non convex optimization problem where the idea is to directly learn the transformation matrix $\mathbf{L}$ of a Mahalanobis distance. To this extent they first propose to define for each example in the training set T the probability that an example $\mathbf{x}_j$ is in the neighbourhood of an example $\mathbf{x}_i$ as:

$$p_{ij} = \frac{\exp\big(-d^2_{\mathbf{L}}(\mathbf{x}_i,\mathbf{x}_j)\big)}{\sum_{\substack{\mathbf{z}_k\in T\\ \mathbf{z}_k\neq\mathbf{z}_i}} \exp\big(-d^2_{\mathbf{L}}(\mathbf{x}_i,\mathbf{x}_k)\big)}. \tag{2.10}$$

  From this, assuming that $p_{ii} = 0$, they compute the probability that the example $\mathbf{x}_i$ is correctly classified as:

$$p_i = \sum_{\substack{\mathbf{z}_j\in T\\ y_i=y_j}} p_{ij}. \tag{2.11}$$

  They then try to find the metric which maximizes the probability of correctly classifying the examples:

$$\underset{\mathbf{L}\in\mathbb{R}^{d'\times d}}{\arg\max} \sum_{\mathbf{z}_i\in T} p_i. \tag{2.12}$$

[3]It is another way to write the 0/1 loss presented in Section 1.4.

To solve this optimization problem, the authors propose to use a gradient based approach and precise that some care should be taken to avoid local maxima. In this work the authors consider that all the examples of the same class should be similar while all the examples of different classes should be dissimilar. There is no explicit notion of margin.

- Globerson and Roweis (2005) proposed Maximally Collapsing Metric Learning (MCML) where the idea is to learn a Mahalanobis distance able to collapse all the similar examples in a single point and to push the dissimilar examples infinitely far away. To this extent the authors propose a convex optimization problem based on the Kullback-Leibler divergence. As in NCA (Goldberger et al., 2004) they first propose to consider for each example in the training set T the probability that an example $\mathbf{x}_j$ is in the neighbourhood of an example $\mathbf{x}_i$ as:

$$p_{ij} = \frac{\exp\left(-d^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j)\right)}{\sum_{\substack{\mathbf{z}_k \in T \\ \mathbf{z}_k \neq \mathbf{z}_i}} \exp\left(-d^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_k)\right)}. \tag{2.13}$$

They also define the ideal probability that they want to achieve between two examples as:

$$p^*_{ij} \propto \begin{cases} 1 & y_i = y_j \\ 0 & y_i \neq y_j. \end{cases} \tag{2.14}$$

Following this the authors propose to learn a metric minimizing the Kullback-Leibler divergence between the empirical and the ideal probability distributions:

$$\underset{\mathbf{M} \in \mathbb{S}^{d\times d}_+}{\arg\min}\, \mathrm{KL}\left(p_{ij} \,\middle|\, p^*_{ij}\right). \tag{2.15}$$

To solve this convex optimization problem the authors propose a gradient based approach with projections onto the constraints.

- Information-Theoretic Metric Learning (ITML) (Davis et al., 2007) is among the most famous Mahalanobis distance learning approaches. It is based on the log det divergence and the idea is to learn a metric which is close to a known prior metric $\mathbf{M}_{\mathcal{S}}$ using the following optimization problem:

$$\underset{\mathbf{M} \in \mathbb{S}^{d\times d}_+}{\arg\min}\ \mathrm{Tr}\left(\mathbf{M}\mathbf{M}_{\mathcal{S}}^{-1}\right) - \log\det\left(\mathbf{M}\mathbf{M}_{\mathcal{S}}^{-1}\right) - n \tag{2.16}$$

$$\text{s.t. } \mathrm{Tr}\left(\mathbf{M}(\mathbf{x}-\mathbf{x}')(\mathbf{x}-\mathbf{x}')^T\right) \leq \gamma_{\text{sim}} \qquad (\mathbf{x}, \mathbf{x}') \in P_{\text{sim}} \tag{2.17}$$

$$\mathrm{Tr}\left(\mathbf{M}(\mathbf{x}-\mathbf{x}')(\mathbf{x}-\mathbf{x}')^T\right) \geq \gamma_{\text{dis}} \qquad (\mathbf{x}, \mathbf{x}') \in P_{\text{dis}}. \tag{2.18}$$

The log det divergence is a particular Bregman divergence with the nice property that if the divergence is finite and the prior matrix is positive semi-definite then the learned matrix is also guaranteed to be positive semi-definite. It implies that this optimization problem does not require projections on the semi-definite cone.

- Jin et al. (2009) propose to learn a Mahalanobis distance using the following optimization problem:

$$\underset{\mathbf{M}\in\mathbb{S}_+^{d\times d},\mathrm{Tr}(\mathbf{M})\leq\eta(d)}{\arg\min} \frac{2}{n(n-1)} \sum_{\substack{\mathbf{z}_i,\mathbf{z}_j\in T\\ i<j}} \left[\delta_{y_i y_j}\left(1-d^2_{\mathbf{M}}(\mathbf{x}_i,\mathbf{x}_j)\right)\right]_+ + \lambda\frac{1}{2}\left\|\mathbf{M}\right\|^2_{\mathcal{F}} \tag{2.19}$$

  where $\eta(d)$ is shown to be sublinear in d, i.e. $\eta(d)\sim\mathcal{O}\left(d^p\right)$ with $p<1$.

- Log-determinant regularized Distance Metric Learning (L-DML) (Zha et al., 2009) is a Mahalanobis distance learning method which is able to make use of some auxiliary knowledge in the form of given metrics. The idea is to use a variant of ITML (Davis et al., 2007) to accommodate several prior metrics rather than a single one. Hence they consider that they have access to a set of prior matrices $\mathbf{M}_1,\cdots,\mathbf{M}_k\in\mathbb{S}_+^{d\times d}$ and use the following optimization problem:

$$\begin{aligned}\underset{\mathbf{M}\mathbb{S}_+^{d\times d},\boldsymbol{\mu}\geq 0}{\arg\min} & \sum_{i=1}^{k}\mu_i\mathrm{Tr}\left(\mathbf{M}_i^{-1}\mathbf{M}\right)-\log\det(\mathbf{M})\\ & +\lambda_{\mathrm{sim}}\sum_{(\mathbf{z},\mathbf{z}')\in P_{\mathrm{sim}}} d_{\mathbf{M}}\left(\mathbf{x},\mathbf{x}'\right)-\lambda_{\mathrm{dis}}\sum_{(\mathbf{z},\mathbf{z}')\in P_{\mathrm{dis}}} d_{\mathbf{M}}\left(\mathbf{x},\mathbf{x}'\right)+\lambda_{\boldsymbol{\mu}}\left\|\boldsymbol{\mu}\right\|_2^2\\ \text{s.t. } & \sum_{i=1}^{k}\mu_i=1\end{aligned}$$

  where the vector $\boldsymbol{\mu}$ controls the impact of each prior matrix and the λ parameters control the trade-off between the different terms.

2.3.2 Triplet Based Constraints

Sometimes pair based constraints are not sufficient to capture the relationships between the constraints. Another common trend in metric learning is to consider triplet based constraints:

- $P_{\mathrm{tri}}=\{(\mathbf{z},\mathbf{z}',\mathbf{z}'')\text{ s.t. }\mathbf{z},\mathbf{z}',\mathbf{z}''\in T, y=y', y\neq y''\}$.

Using empirical risk minimization (Section 1.2), learning a Mahalanobis distance could be done by solving the following optimization problem:

$$\underset{\mathbf{M}\in\mathbb{S}_+^{d\times d}}{\arg\min} \sum_{(\mathbf{z},\mathbf{z}',vz'')\in P_{\mathrm{tri}}} \mathbb{1}_{d_{\mathbf{M}}(\mathbf{x},\mathbf{x}')\geq d_{\mathbf{M}}(\mathbf{x},\mathbf{x}'')+\gamma}+\lambda\left\|\mathbf{M}\right\| \tag{2.20}$$

where γ is the desired margin between the two distances and $\mathbb{1}$ is the indicator function. In practice this margin is often set to 1 (Weinberger et al., 2005; Ying et al., 2009; Shi et al., 2014). The underlying idea is that similar examples should be closer to each other than dissimilar ones (See Figure 2.2(b)). Once again many approaches in metric learning are based on a similar idea and make use of a variation of the previous optimization problem.

- Large Margin Nearest Neighbours (LMNN) (Weinberger et al., 2005; Weinberger and Saul, 2009) is a popular metric learning method based on triplets constraints whose goal is to learn a Mahalanobis distance specifically tailored to improve k-nearest neighbours classification. The idea is, for a given example $\mathbf{x}$, to learn a metric which brings closer the k nearest neighbours of a similar class (target examples) and tries to push farther away all the examples of different classes which are closer to $\mathbf{x}$ than the target examples (impostors). The authors propose to use the following convex optimization problem:

$$\underset{\mathbf{M}\in\mathbb{S}_+^{d\times d}}{\arg\min}\, \lambda \sum_{(\mathbf{x},\mathbf{x}')\in P_{\text{sim}}} d^2_{\mathbf{M}}(\mathbf{x},\mathbf{x}') + (1-\lambda) \sum_{(\mathbf{x},\mathbf{x}',\mathbf{x}'')\in P_{\text{tri}}} \left[1 + d^2_{\mathbf{M}}(\mathbf{x},\mathbf{x}') - d^2_{\mathbf{M}}(\mathbf{x},\mathbf{x}'')\right]_+ \tag{2.21}$$

  where λ is a parameter controlling the balance between the term which brings closer the target examples and the term which moves the impostors with respect to the target examples. Satisfying the positive semi-definiteness of $\mathbf{M}$ is costly and general solvers do not handle this optimization problem efficiently. Hence the authors propose a gradient based specific solver which makes use of the possible decomposition $\mathbf{M} = \mathbf{L}^T\mathbf{L}$. Several metric learning approaches are based on the same formulation albeit in different contexts such as local metric learning (Weinberger and Saul, 2008; Semerci and Alpaydın, 2013) or learning intrinsically non linear metrics (Kedem et al., 2012).

- Ying et al. (2009) propose to learn a low rank Mahalanobis distance. The idea is to use the trace norm as a regularization term in the following optimization problem:

$$\underset{\mathbf{M}\in\mathbb{S}_+^{d\times d}}{\arg\min} \sum_{(\mathbf{z},\mathbf{z}',\mathbf{z}'')\in P_{\text{tri}}} \left[1 + d^2_{\mathbf{M}}(\mathbf{x},\mathbf{x}') - d^2_{\mathbf{M}}(\mathbf{x},\mathbf{x}'')\right]_+ + \lambda \text{Tr}\left(\mathbf{M}\right)^2. \tag{2.22}$$

  As seen in Section 1.4 the trace norm corresponds to the sum of the eigenvalues of the matrix. It implies that to obtain a small trace norm the eigenvalues should be minimized and thus go to 0, i.e. the matrix becomes low rank.

- Sparse Compositional Metric Learning (SCML) (Shi et al., 2014) is a Mahalanobis distance learning method based on the idea that any positive semi-definite matrix can be decomposed as a set of rank 1 positive semi-definite matrices. To learn a metric the authors consider that they have access to a set of rank 1 matrices, the bases $B = \left\{\mathbf{b}_i\mathbf{b}_i^T \text{ s.t. } \mathbf{b}_i \in \mathbb{R}^d\right\}_{i=1}^m$. This set can, for example, be obtained thanks to a Fisher discriminant analysis. The goal is then to learn the vector $\mathbf{w}$ which combines the bases. In the general case the authors propose to solve the following optimization problem:

$$\underset{\mathbf{w}\geq 0}{\arg\min}\, \frac{1}{n} \sum_{(\mathbf{z},\mathbf{z}',\mathbf{z}'')\in P_{\text{tri}}} \left[1 + d^2_{\mathbf{M}}(\mathbf{x},\mathbf{x}') - d^2_{\mathbf{M}}(\mathbf{x},\mathbf{x}'')\right]_+ + \lambda \left\|\mathbf{w}\right\|_1 \tag{2.23}$$

  where $\mathbf{M} = \sum_{i=1}^m \mathbf{w}(i)\mathbf{b}_i\mathbf{b}_i^T$. The regularization term tends to promote sparse combination vectors in order to minimize the number of bases needed to compute the matrix.

Note that instead of learning the d^2 parameters of the matrix $\mathbf{M}$, this method only requires to learn the sparse vector $\mathbf{w}$ of size m. Hence it greatly reduces the computational cost since m will, most of the time, be smaller than d^2.

2.3.3 Quadruplet Based Constraints

Introduced by Law et al. (2013) in Quadruplet-wise Metric Learning (Qwise) the underlying idea is that in some particular cases pair or triplet based constraints are not sufficient. As a motivating example they propose the problem of smiling faces. They consider 4 examples ordered as follows $\mathbf{z}'' \prec \mathbf{z} \sim \mathbf{z}' \prec \mathbf{z}'''$, i.e. $\mathbf{z}''$ is not smiling at all, $\mathbf{z}'''$ is smiling a lot and $\mathbf{z}$ and $\mathbf{z}'$ both smile a little. In this case pair or triplet based constraints cannot completely capture the relations between the examples since they are not fully determined, e.g. it is unknown if $\mathbf{z}$ is closer to $\mathbf{z}''$ or to $\mathbf{z}'''$. To solve this problem Law et al. (2013) propose to use quadruplet based constraints of the form:

$$P_{\text{quad}} = \left\{(\mathbf{z}, \mathbf{z}', \mathbf{z}'', \mathbf{z}''') \text{ s.t. } \mathbf{z} \text{ and } \mathbf{z}' \text{ are more similar to each other than } \mathbf{z}'' \text{ and } \mathbf{z}'''\right\}.$$

These constraints can also be extended to take into account a margin γ:

$$P_{\text{quad}} = \left\{(\mathbf{z}, \mathbf{z}', \mathbf{z}'', \mathbf{z}''', \gamma) \text{ s.t. } \mathbf{z} \text{ and } \mathbf{z}' \text{ are more similar than } \mathbf{z}'' \text{ and } \mathbf{z}''' \text{ by a margin } \gamma\right\}.$$

Quadruplet based constraints can accommodate the motivating example by considering that $\mathbf{z}$ and $\mathbf{z}'$ should be closer to each other than $\mathbf{z}''$ and $\mathbf{z}'''$ should. These constraints have been used in a several approaches.

- Law et al. (2014) introduce a Fantope regularization for Mahalanobis distance learning. One of the limits of the trace norm regularization is the fact that it tends to reduce all the eigenvalues of the matrix. However reducing the values of the highest eigenvalues does not reduce the rank and might decrease the performance of the metric. Hence Law et al. (2014) propose to consider a regularization of the form $\text{Tr}(\mathbf{WM})$ where $\mathbf{W}$ is in the convex hull of the set of rank k projection matrices, called a Fantope. This matrix can be built by first computing the eigenvalue decomposition $\mathbf{M} = \mathbf{V}^T\mathbf{\Sigma}\mathbf{V}$ with $\mathbf{\Sigma}$ a diagonal matrix containing the eigenvalues and $\mathbf{V}$ containing the eigenvectors and then by setting $\mathbf{W} = \mathbf{V}^T\mathbf{\Sigma}'\mathbf{V}$ where $\mathbf{\Sigma}'$ is obtained from $\mathbf{\Sigma}$ by replacing the k smallest eigenvalues by 1 and the others by 0. In other words, the idea is to consider that only the k smallest eigenvalues should be minimized. Using this idea the authors propose to solve the following optimization problem:

$$\underset{\mathbf{M}\in\mathbb{S}_+^{d\times d}}{\arg\min} \sum_{(\mathbf{z},\mathbf{z}',\mathbf{z}'',\mathbf{z}''',\gamma)\in P_{\text{quad}}} \left[\gamma + d^2_{\mathbf{M}}(\mathbf{x}, \mathbf{x}') - d^2_{\mathbf{M}}(\mathbf{x}'', \mathbf{x}''')\right]_+ + \lambda \text{Tr}(\mathbf{WM}). \tag{2.24}$$

Note that one of the difficulties when optimizing this kind of problem is that $\mathbf{W}$ depends on the current value of $\mathbf{M}$ and as such it should be updated during the optimization process. Hence the authors propose to consider a sub gradient descent approach and to update $\mathbf{W}$ at each iteration.

- Huo et al. (2016) propose a Capped Trace norm regularization term for Mahalanobis metric learning. This regularization term can be written as $\frac{1}{2}\sum_i \min(\boldsymbol{\sigma}(i), C)$ where $\boldsymbol{\sigma}$ is the vector of singular values of $\mathbf{M}$ and C is a constant threshold. The idea is to limit the impact of the highest singular values in the optimization problem and thus to promote the minimization of the smallest singular values. The authors propose to use the following optimization problem:

$$\underset{\mathbf{M}\in\mathbb{S}_+^{d\times d}}{\arg\min} \sum_{(\mathbf{z},\mathbf{z}',\mathbf{z}'',\mathbf{z}''',\gamma)\in P_{\text{quad}}} \left[\gamma + d^2_{\mathbf{M}}(\mathbf{x},\mathbf{x}') - d^2_{\mathbf{M}}(\mathbf{x}'',\mathbf{x}''')\right]_+ + \frac{\lambda}{2}\sum_i \min(\sigma_i, C). \quad (2.25)$$

  In its original form the proposed optimization problem is not convex but the authors tackle this issue by solving an equivalent convex optimization formulation. The Capped Trace norm regularization is close in spirit to the Fantope regularization (Law et al., 2014). However the authors show that it is less sensitive to hyper parameters.

2.3.4 Online Learning

In the previous subsections we considered a batch setting where all the examples are available at the same time. However in some cases the examples arrive in a stream like fashion. Contrary to classification or regression where the examples can be considered independently from each other, in metric learning most of the approaches work with pairs or triplet of examples. Hence a natural assumption is that these pairs or triplets are given one after the other. The goal is then to learn a metric which is able to change when more and more pairs or triplets are available. Formally assume that we have a sequence of pairs $(\mathbf{x}_1,\mathbf{x}'_1),\ldots,(\mathbf{x}_t,\mathbf{x}'_t),(\mathbf{x}_{t+1},\mathbf{x}'_{t+1})$ and that from the first t examples we were able to learn a Mahalanobis distance $d_{\mathbf{M}_t}$. The goal is to learn a distance $d_{\mathbf{M}_{t+1}}$ such that:

$$d_{\mathbf{M}_{t+1}} = f\big(d_{\mathbf{M}_t}, (\mathbf{x}_{t+1},\mathbf{x}'_{t+1})\big) \quad (2.26)$$

where f is a function able to combine the current metric with the new pair to learn a new metric.

- Pseudo-Metric Online Learning Algorithm (POLA) (Shalev-Shwartz et al., 2004) is interested in learning a Mahalanobis distance and a parameter b corresponding to the threshold between similar and dissimilar examples. It is assumed that pairs of examples arrive one after the other. Given a new example $(\mathbf{x}_{t+1},\mathbf{x}'_{t+1},\delta_{y_{t+1}y'_{t+1}})$ they propose to update the matrix $\mathbf{M}_t$ and the threshold b_t successively solving the following two optimization problems:

$$\begin{aligned}
\mathbf{M}_{t+\frac{1}{2}}, b_{t+\frac{1}{2}} = \underset{\mathbf{M}\in\mathbb{R}^{d\times d}, b\in\mathbb{R}}{\arg\min} \; & \|\mathbf{M}_t - \mathbf{M}\|^2_{\mathcal{F}} + (b_t - b)^2 \\
& \text{s.t.} \quad \left[\delta_{y_{t+1}y'_{t+1}}\left(d^2_{\mathbf{M}_t}(\mathbf{x}_{t+1},\mathbf{x}'_{t+1}) - b_t\right) + 1\right]_+ = 0, \\
\mathbf{M}_{t+1}, b_{t+1} = \underset{\mathbf{M}\in\mathbb{S}_+^{d\times d}, b\geq 1}{\arg\min} \; & \left\|\mathbf{M}_{t+\frac{1}{2}} - \mathbf{M}\right\|^2_{\mathcal{F}} + (b_{t+\frac{1}{2}} - b)^2.
\end{aligned}$$

(2.27)

The first step consists in searching for the (matrix, threshold)-pair achieving a loss of 0 on the new pair of examples while staying as close as possible to the current solution. The second step consists in projecting the new solution onto the set of admissible solutions, i.e. positive semi-definite matrix and threshold greater than 1. Note that a kernel version of this approach is also proposed allowing one to learn non-linear metrics in an online fashion.

- LogDet Exact Gradient Optimization (LEGO) (Jain et al., 2009) is a Mahalanobis distance learning approach. For a new pair of examples $(\mathbf{x}_{t+1}, \mathbf{x}'_{t+1}, \delta_{y_{t+1}y'_{t+1}})$ they propose to use a formulation based on ITML (Davis et al., 2007) where they set the prior matrix $\mathbf{M}_{\mathcal{S}}$ to the current matrix $\mathbf{M}_t$ at each iteration. Hence they obtain the matrix $\mathbf{M}_{t+1}$ by solving the following optimization problem:

$$\mathbf{M}_{t+1} = \underset{\mathbf{M}\in\mathbb{S}_+^{d\times d}}{\arg\min}\ \mathrm{Tr}\left(\mathbf{M}\mathbf{M}_t^{-1}\right) - \log\det\left(\mathbf{M}\mathbf{M}_t^{-1}\right) - d + \lambda\left[\delta_{y_{t+1}y'_{t+1}}\left(d_{\mathbf{M}}(\mathbf{x}_{t+1}, \mathbf{x}'_{t+1}) - \gamma_{y_{t+1}y'_{t+1}}\right)\right]_+ .$$

- Chechik et al. (2009, 2010) proposed Online Algorithm for Scalable Image Similarity (OASIS) a bilinear similarity learning approach specifically designed to handle large datasets of images. In this work the authors work with triplet based constraints and given a new triplet $(\mathbf{x}_{t+1}, \mathbf{x}'_{t+1}, \mathbf{x}''_{t+1})$ they propose to update the metric in the following way:

$$\mathbf{M}_{t+1} = \underset{\mathbf{M}\in\mathbb{R}^{d\times d}}{\arg\min}\ \frac{1}{2}\left\|\mathbf{M} - \mathbf{M}_t\right\|_{\mathcal{F}}^2 + \lambda\left[1 - k_{\mathbf{M}_t}\left(\mathbf{x}_{t+1}, \mathbf{x}'_{t+1}\right) + k_{\mathbf{M}_t}\left(\mathbf{x}_{t+1}, \mathbf{x}''_{t+1}\right)\right]_+ . \quad (2.28)$$

The idea is to update the matrix for each new triplet while staying close to the matrix obtained during the previous iteration. This trade-off is controlled by a parameter λ. The initial matrix is selected as the identity matrix $\mathbf{M}_0 = \mathbf{I}$. The matrix $\mathbf{M}$ can be either unconstrained, symmetric or PSD depending on the problem. This method has been shown to be computationally efficient thanks to the specifically developed solver.

2.3.5 Other Approaches

Other approaches than the one presented above have been considered to learn metrics. For example, Shen et al. (2009, 2012) propose to use the theory of boosting to learn a Mahalanobis distance. The main idea is to notice that any positive semi definite matrix can be decomposed as a combination of rank 1 matrices which can be used and combined as weak learners. In a subsequent work Bi et al. (2011) propose a substantial seep-up of the approach. Another approach was proposed by Qamar et al. (2008) who learn a similarity by using a variant of the voted perceptron algorithm (Freund and Schapire, 1999).

In this thesis we are mainly interested in batch optimization problems based on regularized risk minimization. In Chapters 3 and 4 we consider pair based constraints. In Chapters 5 and 6 we propose two approaches which are not based on standard metric learning constraints as they are able to consider each example individually.

2.4 Theoretical Guarantees

Metric learning is most of the time used as a preprocessing step before other algorithms. As such when considering theoretical guarantees for metric learning two questions may arise. On the one hand studying the generalization ability of the metric is crucial to ensure that distances computed between new examples will be correct. On the other hand considering the impact of the metric on the subsequent algorithm is important as it is a way to theoretically show the interest of learning it.

2.4.1 Generalization Bounds for Metric Learning

Generalization bounds for metric learning are harder to derive than in standard approaches. Indeed one of the common assumptions when proving this kind of guarantees is that the examples are drawn i.i.d. from a probability distribution (See Section 1.3). However in metric learning most of the time the loss functions are defined with respect to pairs or triplets of examples as presented above. One of the issue is then that even if the examples are drawn i.i.d. from $\mathcal{D}_\mathcal{T}$ there is no guarantee that this is also the case for the pairs or the triplets. More precisely if the examples are drawn identically and independently from $\mathcal{D}_\mathcal{T}$, one can only assume that the pairs are independent but not that they are identically distributed. Nevertheless the two particular frameworks presented in Section 1.3 are based on the McDiarmid's inequality (McDiarmid, 1989) that only needs to assume that the examples are independent. Using adapted definitions of uniform stability (Jin et al., 2009) and Rademacher complexity (Cao et al., 2016) these two frameworks have been successfully extended to metric learning with pair based constraints. Note that the robustness framework has also been considered for metric learning (Bellet and Habrard, 2015) but we do not present it here.

Uniform stability for metric learning To extend the uniform stability framework to metric learning, Jin et al. (2009) propose to adapt the definition of uniform stability as follows.

Definition 2.1 (Uniform Stability for Metric Learning)**.** *Let $T \sim \mathcal{D}_\mathcal{T}$ be a size n training set and $\mathbf{z} \sim \mathcal{D}_\mathcal{T}$ be an example. Let T^i be the training set obtained by replacing example i in T by $\mathbf{z}$. Let $\mathcal{A}$ be an algorithm which returns a metric h_T when learning with the training set T and h_{T^i} when learning with the training set T^i. An algorithm $\mathcal{A}$ has uniform stability β with respect to its loss function $l(\cdot)$ if the following holds:*

$$\forall i \in \{1, \dots, n\}, \sup_{\mathbf{z}', \mathbf{z}'' \sim \mathcal{D}_\mathcal{T}} \left| l\left(h_T, \mathbf{z}', \mathbf{z}''\right) - l\left(h_{T^i}, \mathbf{z}', \mathbf{z}''\right) \right| \leq \frac{\beta}{n}. \tag{2.29}$$

Given this definition, they show that using the same proof technique that in Bousquet and Elisseeff (2002b) it is possible to obtain a generalization bound for metric learning similar, up to some constants, to the one presented in Theorem 1.1. As an example of a practical use of the framework, Jin et al. (2009) show that their algorithm, presented in Section 2.3, is uniformly stable and thus that the metric learned with their method generalizes well to new pairs of examples.

Rademacher complexity for metric learning To extend the Rademacher complexity framework, Cao et al. (2016) propose a new definition of the Rademacher complexity specifically tailored for metric learning. It corresponds to the expected value over size n training sets of the Rademacher averages of the dual norm (Definition A.4) of the regularization term.

Definition 2.2 (Rademacher Complexity for Metric Learning)**.** *Let $T' \sim \mathcal{D}_{\mathcal{T}}$ be a set of size n such that the pairs $(\mathbf{z}'_i, \mathbf{z}'_{\lfloor \frac{n}{2} \rfloor + i})$ are i.i.d.. Let $\boldsymbol{\sigma}$ be a vector of n Rademacher Variables, i.e. variables which can take a value of either 1 or -1 with probability $\frac{1}{2}$. Let $\|\cdot\|$ be a norm and $\|\cdot\|_*$ its dual norm*[4]*. The Rademacher average is respectively defined for Mahalanobis distance learning and bilinear similarity learning:*

$$\hat{R}_n(\|\cdot\|) = \frac{1}{\lfloor \frac{n}{2} \rfloor} \mathop{\mathbb{E}}_{\boldsymbol{\sigma}} \left\| \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} \sigma_i (\mathbf{x}_i - \mathbf{x}_{\lfloor \frac{n}{2} \rfloor + i})(\mathbf{x}_i - \mathbf{x}_{\lfloor \frac{n}{2} \rfloor + i})^T \right\|_* \tag{2.30}$$

$$\hat{R}_n(\|\cdot\|) = \frac{1}{\lfloor \frac{n}{2} \rfloor} \mathop{\mathbb{E}}_{\boldsymbol{\sigma}} \left\| \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} \sigma_i \mathbf{x}_i \mathbf{x}^T_{\lfloor \frac{n}{2} \rfloor + i} \right\|_* . \tag{2.31}$$

The Rademacher complexity for Metric Learning is then defined as:

$$R(\|\cdot\|) = \mathop{\mathbb{E}}_{T' \sim \mathcal{D}_{\mathcal{T}}} \hat{R}_n(\|\cdot\|) \tag{2.32}$$

where the expectation is taken over size n training sets.

Note that even if the i.i.d. property of the pairs is relaxed when using the McDiarmid concentration inequality, it is still needed when computing the Rademacher complexity. However, using properties of U-statistics (See e.g. Clémençon et al. (2008)), Cao et al. (2016) show that this is not an issue in practice since it is possible, in expectation over all the possible training sets of size n, to reduce a pair based approach to the case of i.i.d. random variable blocks as required in this definition[5]. Using this new definition of Rademacher complexity for metric learning Cao et al. (2016) derive a generalization bound which is close in spirit to the one presented in Theorem 1.2.

[4] See A.4 for a formal definition.

[5] An example of this is given in Section 4.5 where we use the Rademacher complexity to derive a generalization bound for metric hypothesis transfer learning.

2.4.2 Impact on a Subsequent Algorithm

As we have seen in Chapter 1, metrics are used in a wide range of applications and choosing a good metric through metric learning or other means can be seen as a preprocessing step for classic problems such as clustering or classification. Hence the question of the impact of the metric on the algorithm which makes use of it may arise. Such problems have been for example addressed by Balcan et al. (2008) who propose to define the goodness of a metric as its capacity to determine the similarity between the examples and a set of so called reasonable examples. They then show that this can be directly related to the performance of a linear classifier making use of the metric. Building upon this framework, several works propose to learn a good metric (Bellet et al., 2011, 2012) or even to jointly learn the metric and its associated classifier (Nicolae et al., 2015). Using a different approach Guo and Ying (2014) propose to learn a metric specifically designed to improve the performance of a linear SVM (Vapnik, 1998). Building upon the Rademacher Complexity framework they show that the true risk of the classifier is bounded by the empirical risk of the metric. Once again, the problem of considering the impact of the learned metric on an algorithm making use of it is beyond the scope of this thesis.

In this thesis we theoretically justify the approaches presented in Chapters 3, 4 and 5 by deriving generalization bounds based either on the uniform stability or Rademacher complexity frameworks.

2.5 Applications

If many approaches consider learning a metric for a clustering or classification task, some methods are specifically designed to help solve other kind of tasks such as image retrieval or domain adaptation. We provide a quick non exhaustive overview here.

Semi-supervised clustering The idea behind semi-supervised clustering (Xing et al., 2002; Chang and Yeung, 2004) is that instead of having the labels of the examples as in classification, we have only access to similarity and dissimilarity constraints.

Classification A lot of approaches have been interested in learning a metric for classification (Semerci and Alpaydın, 2013; Davis et al., 2007; Nicolae et al., 2015). Similarly, Weinberger et al. (2005); Wang et al. (2012); Goldberger et al. (2004); Qamar et al. (2008) propose to put a special emphasis on learning a metric specifically designed for a nearest neighbour classifier (See Section 1.5). It results in methods where only a subset of the constraints are considered as in LMNN (Weinberger et al., 2005) presented in Section 2.3. Other approaches consider the problem of learning low rank matrices to simplify the subsequent classification algorithm (Ying et al., 2009; Shi et al., 2014) or learning a metric inducing a sparse classifier (Bellet et al., 2012).

Image retrieval Metric learning has also been used to improve image retrieval algorithms where the goal is, given a query image, to retrieve the most similar images in a given set (Chechik et al., 2010; Law et al., 2014; Chang and Yeung, 2007; Huo et al., 2016). Similarly the idea behind face recognition is to be able to recognize a query person in a set of images (Jin et al., 2009; Bohné et al., 2014; Cao et al., 2013b; Zha et al., 2009).

Domain adaptation In domain adaptation we assume that we have access to two domains, the source $\mathcal{S}$ and the target $\mathcal{T}$, and that we want to adapt from the source to the target (See Section 1.5). One way to perform domain adaptation is to bring the two domains closer to each other, i.e. to align the source and target examples such that any classifier leaned on the source can also be applied on the target. Several approaches in metric learning thus propose to learn a metric to bring examples from the source closer to the examples from the target.

- Saenko et al. (2010) propose to use ITML (Davis et al., 2007) presented in Section 2.3 to learn a Mahalanobis distance which brings closer the two domains. They consider a semi-supervised domain adaptation problem, i.e. some of the target examples are labelled. To generate the constraints they propose to randomly select examples from the source and the target and simply consider them as similar if they share the same label and as dissimilar otherwise. Learning a Mahalanobis distance might sometimes not be sufficient to overcome the shift between the two domains. Hence they also consider the kernelized version of ITML in order to obtain a non linear metric.

- In Asymmetric Regularized Cross-domain transformation (ARC-t) Kulis et al. (2011) propose to learn a bilinear similarity between the source and the target domain for a semi-supervised domain adaptation task. The interest is that instead of modifying the source and the target domain at the same time as in Saenko et al. (2010), they simply move one domain closer to the other. They propose to use the following optimization problem:

$$\underset{\mathbf{M}\in\mathbb{R}^{d\times d}}{\arg\min} \sum_{(\mathbf{z}^s,\mathbf{z}^t)\in P_{\text{sim}}} \left[\gamma_{\text{sim}} - k_{\mathbf{M}}\left(\mathbf{x}^s,\mathbf{x}^t\right)\right]_+^2 + \sum_{(\mathbf{z}^s,\mathbf{z}^t)\in P_{\text{dis}}} \left[k_{\mathbf{M}}\left(\mathbf{x}^s,\mathbf{x}^t\right) - \gamma_{\text{dis}}\right]_+^2 + \frac{\lambda}{2}\left\|\mathbf{M}\right\|_{\mathcal{F}}^2$$

  where the two examples from the source and the target are similar if they share the same label and are dissimilar otherwise. Also note that in this work the bilinear similarity is considered to be oriented in the sense that the source example always multiply the matrix $\mathbf{M}$ on the left while the target example always multiply $\mathbf{M}$ on the right. To consider more complex transformations between the source and the target they propose to kernelize their approach.

- Geng et al. (2011) proposed Domain Adaptation Metric Learning (DAML) a Mahalanobis distance learning approach for unsupervised domain adaptation. They propose to learn a metric able to well separate similar and dissimilar examples in the source domain while keeping the source and the target domain close to each other. Hence they

define the set of similar examples P_{sim} as all the pairs of source examples sharing the same label. Similarly they define the set of dissimilar examples P_{dis} as all the pairs of source examples with different labels. To keep the source and the target domain close to each other they propose to use the Maximum Mean Discrepancy (MMD) (Borgwardt et al., 2006) where the idea is that the respective means of the source and target samples should be close even after the projection. They propose to learn the transformation matrix $\mathbf{L}$ which minimizes the following optimization problem:

$$\begin{aligned} \underset{\mathbf{L}\in\mathbb{R}^{d'\times d}}{\arg\min} \quad & \sum_{(\mathbf{z}^s,\mathbf{z}^{s\prime})\in P_{\text{sim}}} d^2_{\mathbf{L}}\big(\mathbf{x}^s,\mathbf{x}^{s\prime}\big) + \left\| \frac{1}{n}\sum_{\mathbf{z}^s\in S}\mathbf{L}\mathbf{z}^s - \frac{1}{m}\sum_{\mathbf{z}^t\in T}\mathbf{L}\mathbf{z}^t \right\|_2^2 \\ \text{s.t.} \quad & \sum_{(\mathbf{z}^s,\mathbf{z}^{s\prime})\in P_{\text{dis}}} d_{\mathbf{L}}\big(\mathbf{x}^s,\mathbf{x}^{s\prime}\big) \geq 1. \end{aligned}$$

They also propose a kernelization of their approach based on a Kernel Principal Component Analysis (KPCA) (Schölkopf et al., 1997) in order to learn a non linear metric.

Other tasks As a last remark note that metric learning has also been used to improve the performance of a kernel (Weinberger and Tesauro, 2007) where the idea is to consider distance based kernels and to optimize the corresponding metric or in a multi-task setting (Parameswaran and Weinberger, 2010) where the idea is to learn one metric for each task under the constraint that all these metrics share a common basis.

In this thesis we demonstrate the interest of our algorithms in a wide range of applications. In Chapter 3 we propose to learn a metric for perceptual color differences and we show the interest of this metric in a segmentation task. In Chapter 4 we consider the problem of learning a metric with auxiliary knowledge and we apply our framework to a semi-supervised domain adaptation task. In Chapter 5 we propose a new framework for machine learning and we demonstrate its good performances for classification problems. Lastly in Chapter 6 the interest of the algorithm is demonstrated on two tasks, namely unsupervised domain adaptation and seamless copy in images.

2.6 Conclusion

In this chapter we proposed a non exhaustive review of metric learning. We chose to consider 4 different questions and to study several approaches proposing different answers to these problems. First we noticed that several kinds of metrics can be learned with metric learning approaches. The most popular one is the Mahalanobis distance while the bilinear similarity has also been widely studied. To include some non linearity several methods propose to learn multiple metrics while others consider either learning a linear metric in a kernel induced space or directly learning an intrinsically non linear metric. Second we presented many approaches interested in learning a metric in a batch setting by using an optimization problem making use

of pair, triplet or quadruplet based constraints. We also considered several methods addressing the problem of learning a metric in an online fashion. Third we introduced two methods to derive generalization guarantees in metric learning. These approaches are respectively based on the uniform stability and the Rademacher complexity frameworks. We also recalled several methods interested in the theoretical impact of a metric on the subsequent algorithm. Fourth we considered different tasks that can be solved with the help of metric learning, namely classification, clustering, image retrieval or domain adaptation.

This chapter concludes the first part of this thesis that was dedicated to the presentation of several notions which will be used throughout our contributions. In the next part we address the problem of controlling the behaviour of a metric such that it either follow or stay close to a reference metric. In our first contribution we propose to learn a metric able to approximate a reference distance from a limited number of examples.

Part II

Metric Learning with a Reference Metric

Chapter 3

Metric Approximation Learning in Perceptual Colour Learning

This chapter is based on the following publication

Michaël Perrot, Amaury Habrard, Damien Muselet, and Marc Sebban. Modeling perceptual color differences by local metric learning. In *European Conference on Computer Vision (ECCV-15)*, pages 96–111. Springer International Publishing, 2014b

---

Abstract

In this chapter we are interested in the problem of estimating an unknown reference metric from a set of pairs of examples. A solution to this problem could be to use metric learning to automatically approximate the values of the reference metric. However most of the algorithms proposed in metric learning are more interested in correctly estimating the relative closeness of the examples rather than the actual distance. In this chapter we propose a new local metric learning algorithm to learn a Mahalanobis distance which correctly approximates a reference metric. Using the uniform stability framework we derive generalization guarantees on the learned model showing that our method is theoretically founded. Furthermore we evaluate our approach in a computer vision problem, namely the computation of perceptual color differences. Having perceptual differences between scene colors is key in many computer vision applications such as image segmentation or visual salient region detection. Nevertheless, most of the times, we only have access to the rendered image colors, without any means to go back to the true scene colors. There are two main existing approaches to tackle this problem. On the one hand, one can compute a complex perceptual distance between rendered image colors. However it makes the distance dependent on the acquisition conditions and thus far from the scene color differences. On the other hand one can estimate the scene colors from the rendered image colors and then evaluate perceptual distances. However it implies the knowledge of the acquisition conditions which is an unreasonable assumption for most of the applications. Our approach allows us to learn a metric which is invariant to the acquisition conditions

and computed only from rendered image colors. Our experimental evaluation shows its great ability (i) to generalize to new colors and devices and (ii) to deal with segmentation tasks.

3.1 Introduction

In recent years, metric learning has mainly been interested in learning metrics able to estimate the relative similarities between examples. This can be attributed to the fact that using similarity and dissimilarity constraints is usually the way to go when learning a metric (See Section 2.3). It implies that automatically learned distances are able to return a small value when comparing similar examples and a large value when comparing dissimilar examples. In this context the exact value is often out of interest. For example when using a nearest neighbour algorithm what really matters is the ordering of the examples rather than the exact distance. However there are some cases where learning a distance able to return a specific value could be of interest. This is for example the case when one wants to approximate an existing distance to simplify its computation or when one has access to pairs of examples alongside their distances but no way to compute the distance between new examples. To sum up here we are more interested in regressing the values of a metric than in learning the best metric for a subsequent algorithm as it is often the case in metric learning (See Section 2.5).

In this chapter we present a new algorithm to deal with the problem of learning a metric able to approximate a reference distance. We propose to learn a Mahalanobis distance which corresponds to a linear transformation of the input space (See Section 2.2). However there is no guarantee that the metric we want to approximate can effectively be embedded in an euclidean space. In other words a linear metric might not be sufficient. Following previous works in local metric learning (Section 2.2) we propose to learn several metrics across the input space. More precisely we consider a hard partitioning of the space (Weinberger and Saul, 2008) and learn one metric for each cluster. To deal with the problem of examples which do not fall in the same cluster we also learn a so called global metric. Moreover we show that our approach is theoretically founded. Indeed we build upon the framework of uniform stability to show that the global and each local metric generalize well. Furthermore, combining these generalization bounds we derive a global bound which holds for the whole model.

We evaluate our method on the computer vision problem of learning perceptual color differences, i.e. differences between colors which are proportional to the color difference perceived by human observers. A metric with such a property is highly desirable for most of computer vision applications and especially for visual saliency detection (Achanta and Süsstrunk, 2010) or image segmentation (Bitsakos et al., 2010). The main drawbacks of existing methods for computing perceptual color differences is that they are either dependent on the acquisition conditions or make unrealistic assumptions to be usable in a practical context. Using our approach we propose to approximate a perceptual color difference. Furthermore, we create a

new dataset specifically dedicated to the task. It allows us to go a step further by learning a metric which is mostly invariant to acquisition conditions. This last point is empirically demonstrated by showing the ability of the learned metric to generalize to new colors and to new cameras, i.e. to new acquisition conditions. Furthermore we illustrate the good behaviour of our distance in a standard segmentation task.

This chapter is organised as follows. First in Section 3.2 we present our local metric learning algorithm. Then in Section 3.3 we derive generalisation bounds which theoretically show the good behaviour of our approach. Section 3.4 is dedicated to the problem of learning color differences in computer vision and to our new dataset created to learn perceptual distances. Finally in Section 3.5 we empirically evaluate our approach before concluding in Section 3.6.

3.2 Regressing the Values of a Reference Metric by Local Metric Learning

In this section we present our metric learning framework whose objective is to approximate a reference distance $\Delta : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^d$. It aims at optimizing K local metrics plus one global metric. Let $\mathcal{T}$ be the domain equipped with the distribution $\mathcal{D}_\mathcal{T}$ over the space $\mathcal{X} \times \mathcal{X} \times \mathcal{R}$ where $\mathcal{X} \in \mathbb{R}^d$ is a vector space and $\mathcal{R} \in \mathbb{R}_+$ is the set of values the reference metric can take. We consider that we have access to a training set of pairs and their distance:

$$T = \left\{(\mathbf{x}_i, \mathbf{x}'_i, \Delta(\mathbf{x}_i, \mathbf{x}'_i))\right\}_{i=1}^{n}. \tag{3.1}$$

For the sake of simplicity, when the examples are clear from the context we replace $\Delta(\mathbf{x}_i, \mathbf{x}'_i)$ by Δ.

To learn a local metric we first divide the space of examples, i.e. $\mathcal{X}$, in K local parts using a clustering algorithm. From this, we deduce $K+1$ regions defining a partition $C_0, C_1, \ldots, C_K$ over the possible pairs of examples, i.e. over $\mathcal{X} \times \mathcal{X}$. A pair $(\mathbf{x}, \mathbf{x}')$ belongs to a region C_j, $1 \leq j \leq K$ if both $\mathbf{x}$ and $\mathbf{x}'$ belong to the same cluster j, otherwise it is assigned to region C_0. In other words, each region C_j corresponds to pairs related to cluster j, while C_0 contains the remaining pairs whose points do not belong to the same cluster. It gives us a finite-size training sample of n_j pairs for each region:

$$T_j = \left\{(\mathbf{x}_i, \mathbf{x}'_i, \Delta)\right\}_{i=1}^{n_j}. \tag{3.2}$$

To approximate Δ we independently learn a Mahalanobis distance in every C_j, $j = 0, 1, \ldots, K$. We define a loss function l on any matrix $\mathbf{M}$ and any pair of examples as:

$$l\left(\mathbf{M}, (\mathbf{x}, \mathbf{x}', \Delta)\right) = \left|(\mathbf{x} - \mathbf{x}')^T \mathbf{M} (\mathbf{x} - \mathbf{x}') - \Delta^2\right|. \tag{3.3}$$

Here we consider the non differentiable absolute loss rather than a more classic loss based on the ℓ_2 norm. It makes the optimization problem harder to solve. However we will see in Section 3.4 that in our application we are mainly interested in having a good metric for small

input : A training set T of n pairs of examples; The number of clusters $K \geq 2$
output: 1 global and K local Mahalanobis distances
begin
  Run K-means to deduce, from T, $K+1$ training subsets $T_j = \{(\mathbf{x}_i, \mathbf{x}'_i, \Delta)\}_{i=1}^{n_j}$, $j = 0, 1 \ldots, K$.
  for $j = 0 \rightarrow K$ **do**
    Learn $\mathbf{M}_{T_j}$ by solving the convex optimization Problem (3.5) using T_j
  end
end

Algorithm 1: Local metric learning

values of the reference distance. In this case the absolute loss is more adapted as it penalizes more small approximation errors which are more likely to happen when dealing with small distances. We denote the empirical error over the set T_j by:

$$\hat{L}_{T_j}(\mathbf{M}) = \frac{1}{n_j} \sum_{(\mathbf{x},\mathbf{x}',\Delta) \in T_j} l\left(\mathbf{M}, (\mathbf{x}, \mathbf{x}', \Delta)\right). \tag{3.4}$$

Finally we suggest to learn the matrix $\mathbf{M}_{T_j}$ minimizing $\hat{L}_{T_j}$ via the following regularized problem:

$$\arg\min_{\mathbf{M} \succeq 0} \hat{L}_{T_j}(\mathbf{M}) + \lambda_j \|\mathbf{M}\|_{\mathcal{F}}^2. \tag{3.5}$$

where $\lambda_j > 0$ is a regularization parameter. It is worth noting that our optimization problem takes the form of a simple regularized least absolute deviation formulation. The interest of using the least absolute deviation, rather than a regularized least square, comes from the fact that it enables accurate estimates of small Δ values.

The pseudo-code of our metric learning algorithm is presented in Algorithm 1. Note that to solve the convex Problem (3.5), we use a classic interior points approach. Moreover, parameter λ_j can be tuned by cross-validation.

3.2.1 Discussion about Local versus Global Metric

Note that in our approach, the metrics learned in the K regions $C_1, \ldots, C_K$ are local metrics while the one learned for region C_0 is rather a global metric considering pairs that do not fall in the same region. Beyond the fact that such a setting will allow us to derive generalization guarantees on our algorithm, it constitutes a straightforward solution to deal with examples at test time that would not be concerned by the same local metric in the input space. In this case, we make use of the matrix $\mathbf{M}_{T_0}$ associated to partition C_0. Another possible solution may consist in resorting to a Gaussian embedding of the local metrics. However, because this solution would imply learning additional parameters, we suggest here to make use of this simple and efficient (parameters-wise) strategy. In the segmentation experiments, we noticed

that $\mathbf{M}_{T_0}$ is used in only ∼20% of the cases. Finally, note that if $K = 1$, this boils down to learning only one global metric over the whole training sample.

In the next section, we prove generalization guarantees for our approach.

3.3 Theoretical Analysis

We now provide a generalization bound justifying that the metrics learned with our approach will generalize well. It is derived by considering (i) a multinomial distribution over the regions, and (ii) per region generalization guarantees that are obtained with the uniform stability framework presented in Section 1.3.

First of all we assume that the training sample $T = \cup_{j=0}^{K} T_j$ is drawn from an unknown distribution $\mathcal{D}_{\mathcal{T}}$ over a domain $\mathcal{T}$ such that for any $(\mathbf{x}, \mathbf{x}', \Delta) \sim \mathcal{D}_{\mathcal{T}}$, $\Delta \leq \Delta_{\max}$, with $\Delta_{\max}$ the maximum distance value used in our context. We assume that given any two examples $\mathbf{x}$ and $\mathbf{x}'$ we have $\|\mathbf{x} - \mathbf{x}'\|_2 \leq 1$, i.e. the examples are normalized[1]. The $K+1$ regions $C_0, \ldots, C_K$ define a partition of the support of $\mathcal{D}_{\mathcal{T}}$ where $\Pr(C_j)$ is the probability that a pair of examples falls in region C_j. In C_j, let $\mathcal{D}_{\mathcal{T}_j}$ be the marginal distribution and $D_j = \max_{(\mathbf{x},\mathbf{x}',\Delta)\sim\mathcal{D}_{\mathcal{T}_j}} \|\mathbf{x} - \mathbf{x}'\|_2$ be the maximum distance between two examples.

Let $\mathbf{M}_T = \{\mathbf{M}_{T_0}, \mathbf{M}_{T_1}, \ldots, \mathbf{M}_{T_K}\}$ be the $K + 1$ matrices learned by Algorithm 1. We define the true error associated to $\mathbf{M}_T$ by:

$$L_{\mathcal{T}}(\mathbf{M}_T) = \sum_{j=0}^{K} L_{\mathcal{T}_j}(\mathbf{M}_{T_j}) \Pr(C_j) \tag{3.6}$$

where

$$L_{\mathcal{T}_j}(\mathbf{M}_{T_j}) = \mathop{\mathbb{E}}_{(\mathbf{x},\mathbf{x}',\Delta)\sim\mathcal{D}_{\mathcal{T}_j}} l\left(\mathbf{M}_{T_j}, (\mathbf{x}, \mathbf{x}', \Delta)\right) \tag{3.7}$$

is the local true risk for region C_j. The empirical error over T of size n is defined as:

$$\hat{L}_T(\mathbf{M}_T) = \frac{1}{n} \sum_{j=0}^{K} n_j \hat{L}_{T_j}(\mathbf{M}_{T_j}) \tag{3.8}$$

where $\hat{L}_{T_j}(\mathbf{M}_{T_j})$, Equation (3.4), is the empirical risk over T_j, i.e. for region C_j.

3.3.1 Generalization Bound per Region C_j

For any learned local matrix $\mathbf{M}_{T_j}$, we provide a bound on its associated local true risk $L_{\mathcal{T}_j}(\mathbf{M}_{T_j})$ in function of the empirical risk $\hat{L}_{T_j}(\mathbf{M}_{T_j})$ over T_j. To this end we use the uniform stability framework presented in Section 1.3. Note that this theoretical analysis is based on the work of (Bousquet and Elisseeff, 2002b). Hence to show a generalization bound in each

[1]Note that in the case of color differences studied in Section 3.4, we work in the RGB cube and any patch belongs to $[0; 255]^3$. It is then easy to normalize each coordinate by $255\sqrt{3}$ to meet the assumption.

region C_j we use the McDiarmid's inequality (Theorem A.1) on the estimation error, i.e. the difference between the true risk and the empirical risk. Before that we need to show that our algorithm to learn a metric in each region is uniformly stable which requires the loss to be bounded and k-lipschitz (Definition A.1).

First of all, our loss function, Equation (3.3), is bounded and k-lipschitz as shown in the two following lemmas.

Lemma 3.1 (Bounded loss function)**.** *For any $0 \leq j \leq K$, let $\mathbf{M}_{T_j}$ be the metric learned for region C_j with the training set T_j, we have that for any example $(\mathbf{x}, \mathbf{x}', \Delta) \sim \mathcal{D}_{\mathcal{T}j}$:*

$$0 \leq l\left(\mathbf{M}_{T_j}, (\mathbf{x}, \mathbf{x}', \Delta)\right) \leq B_j, \tag{3.9}$$

with $B_j = \max\left(\frac{\Delta_{\max}}{\sqrt{\lambda_j}}, \Delta_{\max}^2\right)$.

Proof. The proof of this lemma can be found in Appendix B.1. □

Lemma 3.2 (k-lipschitz continuity)**.** *Let $\mathbf{M}_{T_j}$ and $\mathbf{M}'_{T_j}$ be two matrices for a region C_j and $(\mathbf{x}, \mathbf{x}', \Delta)$ be an example. Our loss $l\left(\mathbf{M}_{T_j}, (\mathbf{x}, \mathbf{x}', \Delta)\right)$ is k-lipschitz continuous with $k = D_j^2$.*

Proof. The proof of this lemma can be found in Appendix B.2. □

We can show that our approach is uniformly stable in the sense of Definition 1.3 for each region C_j.

Lemma 3.3 (Uniform stability per region C_j)**.** *Given two training samples T_j and T_j^i of n_j examples where T_j^i is obtained by replacing example i from T_j by another example drawn independently from $\mathcal{D}_{\mathcal{T}j}$. Let $\mathbf{M}_{T_j}$ and $\mathbf{M}_{T_j^i}$ be the respective optimal solutions of Problem* (3.5) *when learning with T_j and T_j^i. In region C_j our problem is β_j uniformly stable with $\beta_j = \frac{2D_j^4}{\lambda_j}$.*

Proof. The proof of this lemma can be found in Appendix B.3. □

Using Lemma 3.3 about the stability of our algorithm and McDiarmid's inequality (Theorem A.1) we can derive our generalization bound. Let $R_{T_j} = L_{\mathcal{T}_j}(\mathbf{M}_{T_j}) - \hat{L}_{T_j}(\mathbf{M}_{T_j})$ be the estimation error for Problem (3.5) when learning with training set T_j. To apply McDiarmid's inequality we need to bound $\mathbb{E}_{T_j \sim \mathcal{D}_{\mathcal{T}j}}\left[R_{T_j}\right]$ and $\left|R_{T_j} - R_{T_j^i}\right|$. This is done in the two following lemmas.

Lemma 3.4 (Bound on $\mathbb{E}_{T_j \sim \mathcal{D}_{\mathcal{T}j}}\left[R_{T_j}\right]$)**.** *For any β_j uniformly stable learning method of estimation error $R_{T_j} = L_{\mathcal{T}_j}(\mathbf{M}_{T_j}) - \hat{L}_{T_j}(\mathbf{M}_{T_j})$ for a training set T_j, we have:*

$$\mathop{\mathbb{E}}_{T_j \sim \mathcal{D}_{\mathcal{T}j}}\left[R_{T_j}\right] \leq \frac{\beta_j}{n_j}. \tag{3.10}$$

Proof. The proof of this lemma can be found in Appendix B.4. □

Lemma 3.5 (Bound on $\left|R_{T_j} - R_{T_j^i}\right|$)**.** *For any β_j uniformly stable learning method of estimation error $R_{T_j} = L_{\mathcal{T}_j}(\mathbf{M}_{T_j}) - \hat{L}_{T_j}(\mathbf{M}_{T_j})$ for a training set T_j and any B_j bounded loss function we have:*

$$\left|R_{T_j} - R_{T_j^i}\right| \leq \frac{2\beta_j + B_j}{n_j}. \tag{3.11}$$

Proof. The proof of this lemma can be found in Appendix B.5. □

We can now show that Problem (3.5) generalize well for each region C_j.

Lemma 3.6 (Generalization bound per region C_j)**.** *For any matrix $\mathbf{M}_{T_j}$ learned with Problem (3.5) with the training set T_j in region C_j, we have with probability $1-\delta$:*

$$\left|L_{\mathcal{T}_j}(\mathbf{M}_{T_j}) - \hat{L}_{T_j}(\mathbf{M}_{T_j})\right| \leq \frac{2D_j^4}{\lambda_j n_j} + \left(\frac{4D_j^4}{\lambda_j} + B_j\right)\sqrt{\frac{\ln(\frac{2}{\delta})}{2n_j}}. \tag{3.12}$$

Proof. Using the McDiarmid inequality (Theorem A.1) on $R_{T_j} = L_{\mathcal{T}_j}(\mathbf{M}_{T_j}) - \hat{L}_{T_j}(\mathbf{M}_{T_j})$, the estimation error, coupled with Lemma 3.5 for the c_i values we have:

$$\begin{aligned}\Pr\left(\left|R_{T_j} - \mathop{\mathbb{E}}_{T_j\sim\mathcal{D}_{\mathcal{T}_j}}\left[R_{T_j}\right]\right| \geq \epsilon\right) &\leq 2\exp\left(-\frac{2\epsilon^2}{\sum_{i=1}^{n_j}\left(\frac{2\beta_j+B_j}{n_j}\right)^2}\right)\\ &\leq 2\exp\left(-\frac{2\epsilon^2}{\frac{1}{n_j}\left(2\beta_j+B_j\right)^2}\right).\end{aligned}$$

Then, by setting:

$$\delta = 2\exp\left(-\frac{2\epsilon^2}{\frac{1}{n_j}\left(2\beta_j+B_j\right)^2}\right)$$

we obtain:

$$\epsilon = (2\beta_j + B_j)\sqrt{\frac{\ln\left(\frac{2}{\delta}\right)}{2n_j}}$$

and:

$$\Pr\left[\left|R_{T_j} - \mathop{\mathbb{E}}_{T_j\sim\mathcal{D}_{\mathcal{T}_j}}\left[R_{T_j}\right]\right| < \epsilon\right] > 1-\delta.$$

Then, with probability $1-\delta$:

$$\begin{aligned} & R_{T_j} < \mathop{\mathbb{E}}_{T_j\sim\mathcal{D}_{\mathcal{T}_j}}\left[R_{T_j}\right] + \epsilon \\ \Leftrightarrow \quad & L_{\mathcal{T}_j}(\mathbf{M}_{T_j}) - \hat{L}_{T_j}(\mathbf{M}_{T_j}) < \mathop{\mathbb{E}}_{T_j\sim\mathcal{D}_{\mathcal{T}_j}}\left[R_{T_j}\right] + (2\beta_j + B_j)\sqrt{\frac{\ln\left(\frac{2}{\delta}\right)}{2n_j}}\end{aligned}$$

(Lemma 3.4.)

$$\Leftrightarrow \qquad L_{\mathcal{T}_j}(\mathbf{M}_{T_j}) - \hat{L}_{T_j}(\mathbf{M}_{T_j}) < \frac{\beta_j}{n_j} + (2\beta_j + B_j)\sqrt{\frac{\ln\left(\frac{2}{\delta}\right)}{2n_j}}. \tag{3.13}$$

Noting that Lemmas 3.5 and 3.4 also hold for $R'_{T_j} = \hat{L}_{T_j}(\mathbf{M}_{T_j}) - L_{\mathcal{T}_j}(\mathbf{M}_{T_j})$ and using similar arguments than above we obtain with probability $1-\delta$ that:

$$\hat{L}_{T_j}(\mathbf{M}_{T_j}) - L_{\mathcal{T}_j}(\mathbf{M}_{T_j}) < \frac{\beta_j}{n_j} + (2\beta_j + B_j)\sqrt{\frac{\ln\left(\frac{2}{\delta}\right)}{2n_j}}. \tag{3.14}$$

From Equations (3.13) and (3.14) we deduce that with probability $1-\delta$ we have:

$$\left|L_{\mathcal{T}_j}(\mathbf{M}_{T_j}) - \hat{L}_{T_j}(\mathbf{M}_{T_j})\right| < \frac{\beta_j}{n_j} + (2\beta_j + B_j)\sqrt{\frac{\ln\left(\frac{2}{\delta}\right)}{2n_j}}.$$

Replacing β_j by its value gives the lemma. □

This lemma shows that good generalization is achieved in each region with a convergence rate in $\mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$. When the region is compact, the quantity D_j is rather small making the bound tighter. However we will see in the next section that for the generalization of Algorithm 1 there is a trade-off between D_j and the number of regions K.

3.3.2 Generalization Bound for Algorithm 1

The generalization bound of our algorithm is based on the fact that the different marginals $\mathcal{D}_{\mathcal{T}j}$ can be interpreted as the parameters of a multinomial distribution. Then, we have that $(n_0, n_1, \ldots, n_K)$ is an i.i.d. multinomial random variable with parameters $n = \sum_{j=0}^{K} n_j$ and $(\Pr(C_0), \Pr(C_1), \ldots, \Pr(C_K))$. Our result makes use of the Bretagnolle-Huber-Carol concentration inequality for multinomial distributions (Van Der Vaart and Wellner, 1996) which is recalled in Proposition A.1 for the sake of completeness (this result has also been used in other contexts (Xu and Mannor, 2012)).

Theorem 3.1 (Generalization bound for Algorithm 1)**.** *Let $C_0, C_1, \ldots, C_K$ be the regions considered, then for any set of metrics $\mathbf{M}_T = \{\mathbf{M}_{T_0}, \ldots, \mathbf{M}_{T_K}\}$ learned by Algorithm 1 from a data sample T of n triplets, we have with probability at least $1-\delta$ that*

$$\begin{aligned} L_{\mathcal{T}}(\mathbf{M}_T) \leq \hat{L}_T(\mathbf{M}_T) + B\sqrt{\frac{2(K+1)\ln 2 + 2\ln\left(\frac{2}{\delta}\right)}{n}} \\ + \frac{2(KD^4+1)}{\lambda n} + \left(\frac{4(KD^4+1)}{\lambda} + (K+1)B\right)\sqrt{\frac{\ln(\frac{4(K+1)}{\delta})}{2n}} \end{aligned} \tag{3.15}$$

where $B = \max_{0\leq j\leq K} B_j$ is a global bound on the loss function, $D = \max_{1\leq j\leq K} D_j$ is the maximum euclidean distance in a region except C_0 and $\lambda = \min_{0\leq j\leq K} \lambda_j$ is the minimum regularization parameter among the $K+1$ learning problems used in Algorithm 1.

Proof. Let n_j be the number of points of T that fall into the partition C_j. $(n_0, n_1, \dots, n_K)$ is an i.i.d. multinomial random variable with parameters n and $(\Pr(C_0), \Pr(C_1), \dots, \Pr(C_K))$.

$$
\begin{aligned}
&\left|L_{\mathcal{T}}(\mathbf{M}_T) - \hat{L}_T(\mathbf{M}_T)\right| \\
&= \left|\sum_{j=0}^{K} L_{\mathcal{T}_j}(\mathbf{M}_{T_j}) \Pr(C_j) - \hat{L}_T(\mathbf{M}_T)\right| \\
&= \left|\sum_{j=0}^{K} L_{\mathcal{T}_j}(\mathbf{M}_{T_j}) \Pr(C_j) - \sum_{j=0}^{K} L_{\mathcal{T}_j}(\mathbf{M}_{T_j}) \frac{n_j}{n} + \sum_{j=0}^{K} L_{\mathcal{T}_j}(\mathbf{M}_{T_j}) \frac{n_j}{n} - \hat{L}_T(\mathbf{M}_T)\right| \\
&\qquad \text{(Triangle inequality.)} \\
&\leq \left|\sum_{j=0}^{K} L_{\mathcal{T}_j}(\mathbf{M}_{T_j}) \Pr(C_j) - \sum_{j=0}^{K} L_{\mathcal{T}_j}(\mathbf{M}_{T_j}) \frac{n_j}{n}\right| + \left|\sum_{j=0}^{K} L_{\mathcal{T}_j}(\mathbf{M}_{T_j}) \frac{n_j}{n} - \hat{L}_T(\mathbf{M}_T)\right| \\
&\qquad \text{(Triangle inequality.)} \\
&\leq \sum_{j=0}^{K} L_{\mathcal{T}_j}(\mathbf{M}_{T_j}) \left|\Pr(C_j) - \frac{n_j}{n}\right| + \left|\sum_{j=0}^{K} L_{\mathcal{T}_j}(\mathbf{M}_{T_j}) \frac{n_j}{n} - \sum_{j=0}^{K} \frac{n_j}{n} \hat{L}_{T_j}(\mathbf{M}_{T_j})\right| \\
&\qquad \text{(Lemma 3.1 coupled with the definition of } B.) \\
&\leq \sum_{j=0}^{K} B \left|\Pr(C_j) - \frac{n_j}{n}\right| + \left|\sum_{j=0}^{K} \frac{n_j}{n} \left[L_{\mathcal{T}_j}(\mathbf{M}_{T_j}) - \hat{L}_{T_j}(\mathbf{M}_{T_j})\right]\right| \\
&\qquad \text{(Proposition A.1 with probability } 1 - \tfrac{\delta}{2}.) \\
&\leq B\sqrt{\frac{2(K+1)\ln 2 + 2\ln\left(\frac{2}{\delta}\right)}{n}} + \sum_{j=0}^{K} \frac{n_j}{n} \left|L_{\mathcal{T}_j}(\mathbf{M}_{T_j}) - \hat{L}_{T_j}(\mathbf{M}_{T_j})\right| \\
&\qquad \text{(Lemma 3.6 with probability } 1 - \tfrac{\delta}{2(K+1)} \text{ in each region.)} \\
&\leq B\sqrt{\frac{2(K+1)\ln 2 + 2\ln\left(\frac{2}{\delta}\right)}{n}} + \sum_{j=0}^{K} \frac{n_j}{n} \left(\frac{2D_j^4}{\lambda_j n_j} + \left(\frac{4D_j^4}{\lambda_j} + B_j\right)\sqrt{\frac{\ln(\frac{4(K+1)}{\delta})}{2n_j}}\right) \\
&\qquad \text{(Definition of } B,\ D,\ \lambda,\ D_0 = 1 \text{ and noting that } \sqrt{n_j} \leq \sqrt{n}.) \\
&\leq B\sqrt{\frac{2(K+1)\ln 2 + 2\ln\left(\frac{2}{\delta}\right)}{n}} \\
&\quad + \frac{2(KD^4+1)}{\lambda n} + \left(\frac{4(KD^4+1)}{\lambda} + (K+1)B\right)\sqrt{\frac{\ln(\frac{4(K+1)}{\delta})}{2n}}.
\end{aligned}
$$

Finally, the union bound (Theorem A.2) gives the theorem with probability $1 - \delta$. □

This result justifies that good generalization is achieved globally with a standard convergence rate in $\mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$. We can remark that if the local regions $C_1, \dots, C_K$ are rather small (i.e. D is significantly smaller than 1), then the last part of the bound will not suffer too

much on the number of regions. On the other hand, there is also a trade-off between the number/size of regions considered and the number of instances falling in each region. It is important to have enough examples to learn good models.

3.4 Learning Perceptual Color Differences

In computer vision, the evaluation of color differences is required for many applications. For example, in image segmentation, the basic idea is to merge two neighbour pixels in the same region if the difference between their colors is "small" and to split them into different regions otherwise (Bitsakos et al., 2010). Likewise, for visual salient region detection, the color difference between one pixel and its neighbourhood is also the main used information (Achanta and Süsstrunk, 2010), as well as for edge and corner detection (Van de Weijer et al., 2006, 2005). Similarly, in order to evaluate the quality of color images, (Xue et al., 2013) have shown that the pixel-wise mean square difference between the original and distorted image provides very good results. As a last example, the orientation of gradient which is the most widely used feature for image description (SIFT (Lowe, 2004), HOG (Dalal and Triggs, 2005)) is evaluated as the ratio between vertical and horizontal differences.

Depending on the application requirement, the used color difference may have different properties. For material edge detection, it has to be robust to local photometric variations such as highlights or shadows (Van de Weijer et al., 2005). For gradient-based color descriptors, it has to be robust to acquisition condition variations (Burghouts and Geusebroek, 2009; Van De Sande et al., 2010) or discriminative (Van de Weijer et al., 2006). For most applications and especially for visual saliency detection (Achanta and Süsstrunk, 2010), image segmentation (Bitsakos et al., 2010) or image quality assessment (Xue et al., 2013), the color difference has to be above all perceptual, i.e. proportional to the color difference perceived by human observers. As such a large amount of work has been done by color scientists around perceptual color differences (Wyszecki and Stiles, 2000; Huang et al., 2012; Sharma et al., 2005), where the required inputs of the proposed distances are either *reflectance spectra* or the *device-independent color components* CIE XYZ (Wyszecki and Stiles, 2000). These features are obtained with particular devices such as spectrophotometer or photoelectric colorimeter (Wyszecki and Stiles, 2000). It is known that neither the euclidean distance between reflectance spectra nor the euclidean distance between XYZ vectors are perceptual, i.e. these distances can be higher for two colors that look similar than for two colors that look different. Consequently, some color spaces such as CIELAB or CIELUV have been designed to be more perceptually uniform. In those spaces, specific color difference equations have been proposed to further improve perceptual uniformity over the simple euclidean distance (Huang et al., 2012). The ΔE_{00} (Sharma et al., 2005) distance is one nice example of such a distance. It corresponds to the difference perceived by a human looking at the two considered colors under standard viewing conditions recommended by the CIE (illuminant D65, illuminance of 1000 lx, etc.).

However, it is worth noting that in most of the computer vision applications, the available information does not take the form of a reflectance spectra or some device-independent components, as assumed above. Indeed, the classical acquisition devices are cameras that use iterative complex transforms from the irradiance (amount of light) collected by each CCD sensor cell to the pixel intensity of the output image (Kim et al., 2012b). These device-dependent transforms are color filtering, white-balancing, gamma correction, demosaicing, compression, etc. (Xiong et al., 2012b) which are designed to provide pleasant images and not to accurately measure colors. Consequently, the available RGB components in color images do not allow us to get back to the original spectra or XYZ components. To overcome this limitation, two main strategies have been suggested in the literature: either by applying a default transformation from RGB components to $L^*a^*b^*$ (CIELAB space) or $L^*u^*v^*$ (CIELUV space) assuming a given configuration, or by learning a coordinate transform to actual $L^*a^*b^*$ components under particular conditions.

Using default transformations A classic strategy consists in using a default transformation from the available RGB components to XYZ and then to $L^*a^*b^*$ or $L^*u^*v^*$ (Achanta and Süsstrunk, 2010; Arbelaez et al., 2011; Bitsakos et al., 2010; Khan et al., 2013; Mojsilovic, 2005). This default transformation assumes an average gamma correction of 2.2 (Stokes et al., 1996), color primaries close to ITU-R BT.709 (Union, 2000) and D65 illuminant (Daylight). Finally, from the estimated $L^*a^*b^*$ or $L^*u^*v^*$ (denoted $\widehat{L^*a^*b^*}$ and $\widehat{L^*u^*v^*}$ respectively) of two pixels, one can make use of the euclidean distance. In the case of $L^*a^*b^*$, one can use $\widehat{L^*a^*b^*}$ to estimate more complex and accurate distances such as ΔE_{00} via its estimate $\widehat{\Delta E_{00}}$ (Sharma et al., 2005), that will be used in our experimental study as a baseline. This default approach provides a perceptual distance between the colors in the rendered image (called image-wise color distance) and not between the colors as they appear to a human observer looking at the real scene (called scene-wise color distance). For some applications such as image quality assessment, it is required to use the image-wise color distances since only the rendered image colors need to be compared, whatever the scene colors. But for a lot of other applications such as image segmentation or saliency detection, we claim that a scene-wise perceptual color distance should be used. Indeed, in these cases, the aim is to be able to evaluate distances as they would have been perceived by a human observing the scene and not after the camera transformations. Note that some solutions exist (Kim et al., 2012a) to get back to scene colors from RGB camera outputs, thus avoiding using a default transformation, but they require calibrated acquisition conditions (known illumination, known sensor sensitivities, RAW data available, ...).

Learning coordinate transforms to $L^*a^*b^*$ For applications requiring the distances between the colors in the scene, the acquisition conditions are calibrated first and then the images are acquired under these particular conditions (Larraín et al., 2008; Leon et al., 2006). Therefore, the camera position and the light color, intensity and positions are fixed and a set of images of different color patches are acquired. Meanwhile, under the same exact condi-

tions, a colorimeter measures the actual $L^*a^*b^*$ components (in the scene) for each of these patches. Leon et al. (2006) learn then the best transform from camera RGB to actual $L^*a^*b^*$ components with a neural network. Larraín et al. (2008) first apply the default transform presented before from camera RGB to $\widehat{L^*a^*b^*}$ and then learn a polynomial regression (until quadratic term) from the $\widehat{L^*a^*b^*}$ to the true $L^*a^*b^*$. However, it is worth mentioning that in both cases the learned transforms are accurate only under these acquisition conditions. Thus, these approaches can not be applied on most of the computer vision applications where such an information is unavailable.

Using the metric learning method presented in Section 3.2 we propose to estimate scene-wise color distances from non calibrated rendered image colors. Furthermore, we go a step further towards an invariant color distance. This invariance property means that, considering one image representing two color patches, the distance is predicting how much difference would have perceived a human observer looking at the two real patches under standard fixed viewing conditions, such as the ones recommended by the CIE (Commission Internationale de l'Eclairage) in the context of color difference assessment (Sharma et al., 2005). In other words, whatever the acquisition device or the illuminant, an invariant scene-wise distance should return stable values. To the best of our knowledge, no previous work has both underlined and answered the problem of the approximations that are made during the estimation of perceptual color differences in the very frequent case of non calibrated acquisitions. It implies that no suitable dataset exists for the problem at hand. Hence we propose a new dataset specifically designed to learn a perceptual distance which is invariant across acquisition conditions.

3.4.1 Creating the Dataset

Given two color patches, we want to design a perceptual distance not disturbed by the acquisition conditions. So we propose to use pairs of patches for which we can measure the true perceptual distance under standard viewing conditions and to image them under different other conditions.

The choice of the patches is key in this work since all the distances will be learned from these pairs. Consequently, the colors of the patches have to be well distributed in the RGB cube in order to be able to well approximate the color distance between two new pairs that have not been seen in the training set. Moreover, as we would like to learn a local perceptual distance, we need pairs of patches whose colors are close from each other. According to Sharma et al. (2005), ΔE_{00} seems to be a good candidate for that because it is designed to compare similar colors. Finally, since hue, chroma and luminance differences impact the perceptual color difference (Sharma et al., 2005), the patches have to be chosen so that all these three variations are represented among the pairs.

Given these three requirements, we propose to use two different well-known sets of patches, namely the Farnsworth-Munsell 100 hue test and the Munsell atlas (see Figure 3.1). The Farnsworth-Munsell 100 hue test is one of the most famous color vision tests which consists

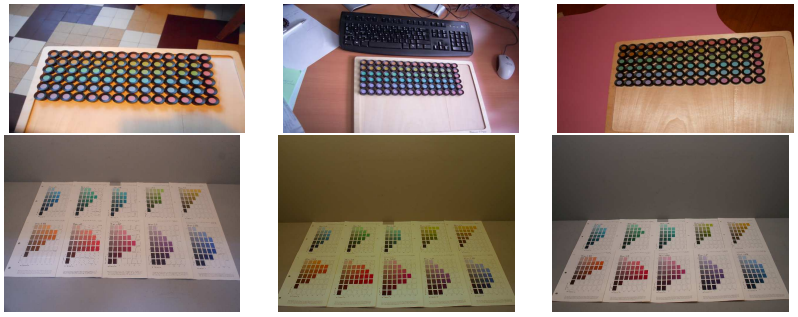


Figure 3.1: Some images from our dataset showing (first row) the 84 Farnsworth-Munsell patches and (second row) the 238 Munsell patches under different conditions.

in ordering 84 patches in the correct order and any misplacement can point to some sort of color vision deficiency. Since these 84 patches are well distributed on the hue wheel, their colors will cover a large area of the RGB cube when imaging them under an important range of acquisition conditions. Furthermore, consecutive patches are known to have very small color differences and then, learning perceptual distances from such pairs is a good purpose. This set is constituting the main part of our dataset. However, the colors of these patches first, are not highly saturated and second, they mostly exhibit hue variations and relatively small luminance and chroma differences. In order to cope with these weaknesses, we add to this dataset the 238 patches constituting the Munsell Student Color Set (Munsell, 1912). These patches are characterized by more saturated colors and the pairs of similar patches mostly exhibit luminance and chroma variations (since only the 5 principal and 5 intermediate hues are provided in this student set).

To build the dataset, we first use a spectroradiometer (Minolta CS 1000) in order to measure the spectra of each color patch of the Farnsworth set, the spectra of the Munsell atlas patches being available online [2]. Five measurements have been done in our light cabinet and the final spectra are the average of each measurement. From these spectra, we evaluate the $L^*a^*b^*$ coordinates of each patch under D65 illuminant. Then, we evaluate the distance ΔE_{00} between all the pairs of color patches (Sharma et al., 2005). Since we need patch pairs whose colors are similar, following the CIE recommendations (CIE Standard DS 014-6/E:2012), we select among the $C_{84}^2 + C_{238}^2$ available pairs only the 223 that are characterized by a Euclidean distance in the CIELAB space (denoted ΔE_{ab}) less than 5.

Note that the available ΔE_{00} have been evaluated in the standard viewing conditions recommended by the CIE for color difference assessment and we would like to obtain these reference distances whatever the acquisition conditions. Consequently, we propose to use 4 different cameras, namely Kodak DCS Pro 14n, Konica Minolta Dimage Z3, Nikon Coolpix S6150 and Sony DCR-SR32 and a large variety of lights, viewpoints and backgrounds (since background also perturbs the colors of the patches). For each camera, we acquire 50 images

[2] https://www.uef.fi/spectral/spectral-database

of each Farnsworth pair and 15 of each Munsell pair (overall, 41,800 imaged pairs). Finally, after all these measurements and acquisitions, we have for each image of a pair, two image rendered RGB vectors and one reference distance ΔE_{00}.

In the next section, using this dataset, we evaluate the approach presented in Section 3.2.

3.5 Experiments

Evaluating the interest of a metric can be done in two ways:

- assessing the quality of the metric itself,
- measuring its impact once plugged in an application.

In the following, we evaluate the generalization ability of the learned metric on our dataset and we measure its contribution in a color segmentation application but first we give a brief overview of how we learn a metric on our dataset.

3.5.1 Learning the Metric

From our dataset of 41,800 pairs and their reference distance ΔE_{00} we can draw training sets T of varying size depending on our needs. Given T we learn a set of $K+1$ local distances $\mathbf{M}_T = \{\mathbf{M}_{T_0}, \mathbf{M}_{T_1}, \ldots, \mathbf{M}_{T_K}\}$. Note that the local metrics are relatively simple since they correspond to 3×3 matrices. Furthermore, in all our experiments we consider a large amount of the training pairs. It makes our algorithm rather insensible to the choice of λ. Therefore, we chose to fix $\lambda = 1$.

The learned metric for a given training set T is denoted by Δ_T. For two examples $\mathbf{x}, \mathbf{x}'$ it is computed as follows:

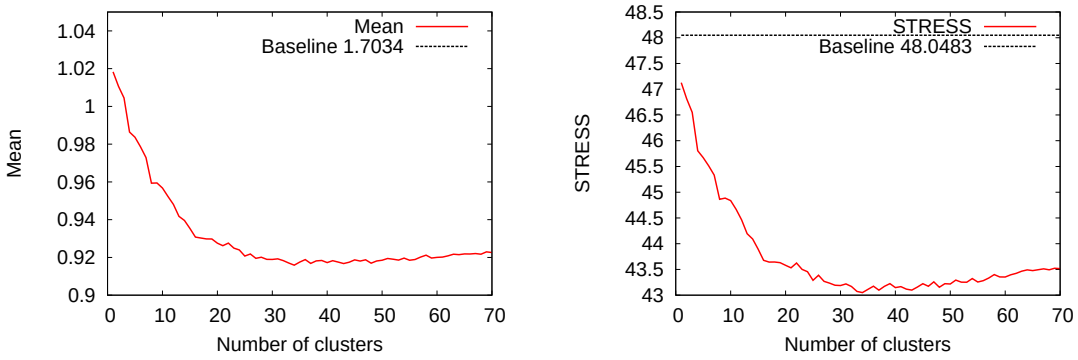
$$\Delta_T(\mathbf{x}, \mathbf{x}') = \begin{cases} (\mathbf{x}-\mathbf{x}')^T \mathbf{M}_{T_j} (\mathbf{x}-\mathbf{x}') & \text{if } \mathbf{x} \text{ and } \mathbf{x}' \text{ fall in the same cluster } C_j,\ 1 \leq j \leq K, \\ (\mathbf{x}-\mathbf{x}')^T \mathbf{M}_{T_0} (\mathbf{x}-\mathbf{x}') & \text{otherwise.} \end{cases} \tag{3.16}$$

3.5.2 Evaluation on our Dataset

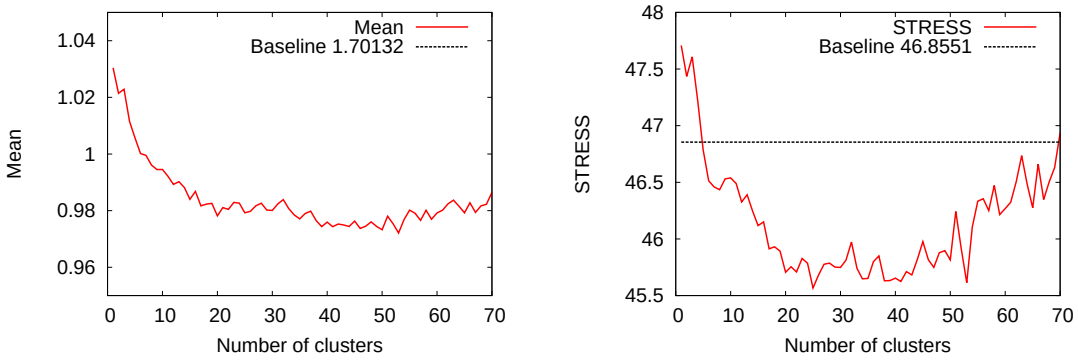
To empirically evaluate the generalization ability of the metric, we conduct two experiments. On the one hand we assess the behaviour of our approach when it is applied to new unseen colors. On the other hand we consider the problem of patches coming from a new unseen camera, i.e. of new acquisition conditions. All the results presented are averaged over 5 runs.

To estimate the performance of our metric we use two criteria that we want to make as small as possible. These two criteria are computed over a test set $T' = \{(\mathbf{x}_i, \mathbf{x}'_i, \Delta E_{00})\}_{i=1}^{n'}$ independent from the training set T. The first criterion is the mean absolute difference between the learned metric Δ_T and the reference metric ΔE_{00}:

$$\text{mean} = \frac{1}{n} \sum_{(\mathbf{x},\mathbf{x}',\Delta E_{00}) \in T'} \left| \Delta_T - \Delta E_{00} \right|. \tag{3.17}$$



(a) Generalization to new colors.



(b) Generalization to new cameras.

Figure 3.2: 3.2(a) Generalization of the learned metrics to new colors; 3.2(b) Generalization of the learned metrics to new cameras. For 3.2(a) and 3.2(b), we plotted the Mean and STRESS values as a function of the number of clusters. The horizontal dashed line represents the STRESS baseline of $\widehat{\Delta E_{00}}$. For the sake of readability we have not plotted the mean baseline of $\widehat{\Delta E_{00}}$ at 1.70.

As a second criterion, we use the STRESS[3] measure (Melgosa et al., 2008) which is widely used by the computer vision community as a way to compare color differences. It is defined as follows:

$$\text{STRESS} = 100\sqrt{\left(\frac{\sum_{(\mathbf{x},\mathbf{x}',\Delta E_{00})\in T'}(\Delta E_{00} - F\Delta_T)^2}{\sum_{(\mathbf{x},\mathbf{x}',\Delta E_{00})\in T'} F^2\Delta_T^2}\right)} \text{ with } F = \frac{\sum_{(\mathbf{x},\mathbf{x}',\Delta E_{00})\in T'} \Delta E_{00}^2}{\sum_{(\mathbf{x},\mathbf{x}',\Delta E_{00})\in T'} \Delta E_{00}\Delta_T}. \tag{3.18}$$

Roughly speaking the STRESS evaluates quadratic differences between the learned metric Δ_T and the reference Δ. We compare our approach to the state of the art where Δ_T is replaced by $\widehat{\Delta E_{00}}$ (Sharma et al., 2005) in both criteria, i.e. transforming from rendered image RGB to $\widehat{L^*a^*b^*}$ and computing the $\widehat{\Delta E_{00}}$ distance.

[3]STandardized REsidual Sum of Squares.

Generalization to Unseen Colors

In this experiment, we perform a 6-fold cross validation procedure over the set of *patches*. Thus we obtain, on average, 27927 training pairs and 13873 testing pairs. The results are shown on Figure 3.2(a) according to an increasing number of clusters (from 1 to 70). We can see that using our learned metric Δ_T instead of the state of the art estimate $\widehat{\Delta E_{00}}$ (Sharma et al., 2005) enables significant improvements according to both criteria (the baselines are 1.70 for the mean and 48.05 for the STRESS). Note that from 50 clusters onward, the quality of the learned metric declines slightly while remaining much better than $\widehat{\Delta E_{00}}$. Figure 3.2(a) shows that $K = 20$ seems to be a good compromise between a high algorithmic complexity (the higher K, the larger the number of learned metrics) and good performances of the models. When $K = 20$, using a Student's t test over the mean absolute differences and a Fisher test over the STRESS, our method is significantly better than the state of the art with a p-value $< 1^{-10}$. Figure 3.2(a) also emphasizes the interest of learning several local metrics. Indeed, optimizing 20 local metrics rather than only one is significantly better with a p-value smaller than 0.001 for both criteria.

Generalization to Unseen Cameras

In this experiment, our model is learned according to a 4-fold cross validation procedure such that each fold corresponds to the pairs coming from a given camera. Thus we learn the metric on a set of 31350 pairs and test it on a set of 10450 pairs. This task is more complicated than generalizing to unseen colors. Indeed when generalizing to unseen colours even if the metric has never seen a given colour before it has been learned on similar examples. Contrarily the acquisition conditions highly depend on the kind of camera used and can vastly differ from one camera to another (Ilie and Welch, 2005). Given that we use a limited number of cameras there is no guarantee that similar acquisition conditions have been seen before. The results are presented in Figure 3.2(b). We can note that our approach always outperforms the state of the art for the mean criterion (of baseline 1.70). Regarding the STRESS, we are on average better when using between 5 to 60 clusters. Beyond 65 clusters, the performances decrease significantly. This behaviour likely describes an overfitting phenomenon due to the fact that a lot of local metrics have been learned that are more and more specialized for 3 out of 4 cameras, and unable to generalize well to the fourth one. For this series of experiments, $K = 20$ is still a good value to deal with the trade-off between complexity and efficiency. Using a Student's t test over the mean absolute differences and a Fisher test over the STRESS, our method is significantly better with p-values respectively $< 1^{-10}$ and < 0.006. The interest of learning several local metrics rather than only one is still confirmed. Applying statistical comparison tests between $K = 20$ and $K = 1$ leads to small p-values < 0.001.

Thus for both series of experiments, $K = 20$ appears to be a good number of clusters and allows significant improvements. Therefore, we suggest to take this value in the next section to tackle a segmentation problem. Before that, let us finish this section by geometrically showing

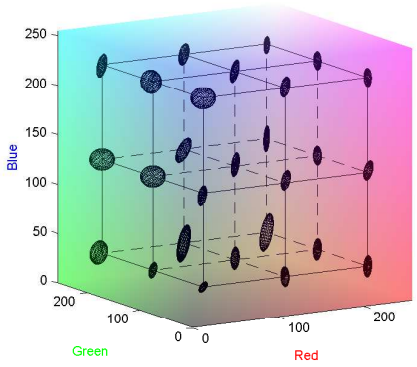


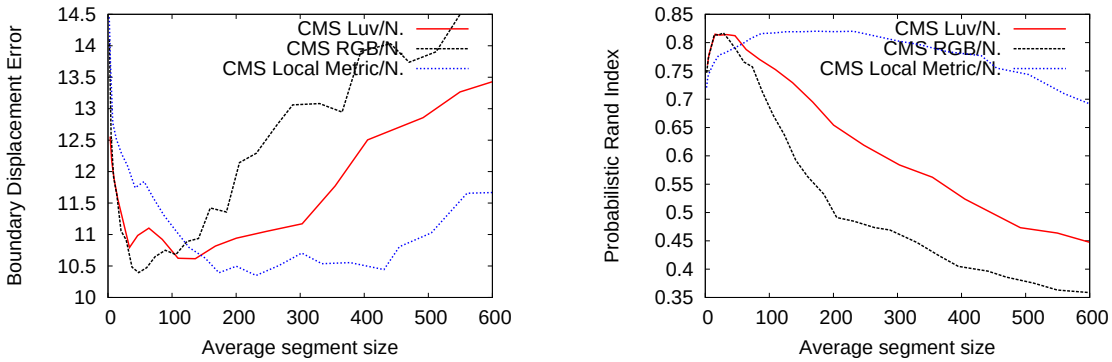
Figure 3.3: Interest of learning local metrics. We took 27 points uniformly distributed on the RGB cube. Around each point we plotted an ellipsoid where the surface corresponds to the RGB colors lying at a learned distance of 1. In this case we used the metric learned by our algorithm using $K = 20$.

the interest of learning local metrics. Figure 3.3 shows ellipsoids uniformly distributed in the RGB space whose surface corresponds to the RGB colors lying at the corresponding learned local distance of 1 from the center of the ellipsoid. It is worth noting that the variability of the shapes and orientations of the ellipsoids is high, meaning that each local metric could capture local specificities of the color space. The experimental results presented in the next section will prove this claim.

3.5.3 Application to Image Segmentation

In this experiment, we evaluate the performance of our approach in a color based image segmentation application. We propose to use the approach proposed by Bitsakos et al. (2010) that suggests a nice extension of the classic mean-shift algorithm (Fukunaga and Hostetler, 1975) by accounting for color information. Furthermore, the authors show that the more perceptual the used distance, the better the results. Especially, by using the default transform from the available camera RGB to the $\widehat{L^*u^*v^*}$ space, they significantly improve the segmentation results over the simple RGB coordinates. Our aim is not to propose a new segmentation algorithm but to use the exact algorithm proposed by Bitsakos et al. (2010) working in the RGB space and to replace in their code (publicly available) the distance between two colors with our learned color distance Δ_T. This way, we can compare the perceptual property of our distance with this of the recommended default approach (euclidean distance in the $\widehat{L^*u^*v^*}$ space).

Therefore, we take exactly the same protocol as Bitsakos et al. (2010). We use the same 200 images taken from the well-known Berkeley dataset and the associated ground-truth that is constituted by 1087 segmented images provided by humans. In order to assess the quality of the segmentation, as recommended by Bitsakos et al. (2010), we use the average Boundary



(a) Boundary Displacement Error.

(b) Probabilistic Rand Index.

Figure 3.4: 3.4(a) Boundary Displacement Error (lower is better) versus the average segment size. 3.4(b) Probabilistic Rand Index (higher is better) versus the average segment size.

Displacement Error (BDE) and the Probabilistic Rand Index (PRI). Note that the better the quality of the segmentation, the lower the BDE and the higher the PRI. The segmentation algorithm proposed in Bitsakos et al. (2010) has one main parameter which is the color distance threshold under which two neighbour pixels (or sets of pixels) have to be merged in the same segment. As in Bitsakos et al. (2010), we plot the evolution of the quality criteria versus the average segment size (see Figures 3.4(a) and 3.4(b)). For comparison, we have run the code from Bitsakos et al. (2010) for the parameters providing the best results in their paper, namely "CMS Luv/N.", corresponding to their color mean-shift (CMS) applied in the $\widehat{L^*u^*v^*}$ color space. The results of CMS applied in the RGB color space with the classical euclidean distance are plotted as "CMS RGB/N." and those of CMS applied with our color distance in the RGB color space are plotted as "CMS Local Metric/N.".

For both criteria, we can see that our learned color distance significantly improves the quality of the results over the two other approaches, i.e. it provides a segmentation that is closer to the one computed by humans. This is truer when the segment size is increasing (right part of the plots). It is important to understand that increasing the average segment size (moving to the right on the plots) is like merging neighbour segments in the images. So by analysing the curves, we can see that for the classic approaches ("CMS Luv/N." and "CMS RGB/N."), it seems that the segments that are merged together when moving to the right on the plot are not the ones that would be merged by humans. That is why both criteria are worst (BDE increases and PRI decreases) on the right for these methods. On the other hand, it seems that our distance is more accurate when merging neighbour segments since for high average segment sizes, our results are much better. This point can be observed in Figure 3.5, where the segment size is high, i.e. when the number of clusters is low (50), the segmentation provided by RGB or $\widehat{L^*u^*v^*}$ are far from the ground truth, unlike our approach which provides nice results. To get the same perceptual result, both methods require about 500 clusters.

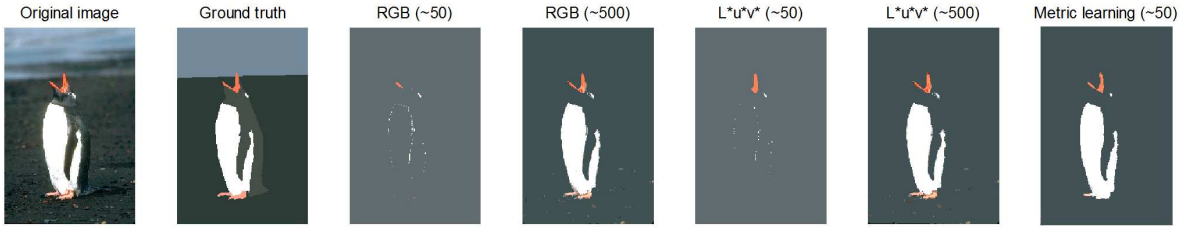


Figure 3.5: Segmentation illustration. When the number of clusters is low (around 50), the segmentation provided by RGB or $\widehat{L^*u^*v^*}$ are far from the ground truth, unlike our approach which provides nice results. To get the same perceptual result, both methods require about 500 clusters.

We further illustrate the performance of our method in Figure 3.6. As explained before, the number of segments in the resulting images is not a parameter of the algorithm, as a consequence it is not easy to obtain images with the same number of segments for the three algorithms (RGB, $L^*u^*v^*$ and Metric learning). Thus, given an image, by modifying the color distance threshold, we tried to obtain the same segment numbers as the corresponding ground truth for the three algorithms. However, the color mean-shift algorithm provides some very small segments, specially for the RGB and $L^*u^*v^*$ color spaces. Consequently, for each test, in Figure 3.6, we have mentioned between brackets, first, the number of segments, and second, the number of segments whose size is more than 150 pixels. For a fair comparison, we use this last number as reference for each image, i.e. this number is almost constant and close to the ground truth for each row. It is worth mentioning that the ground truth segmentation has always very few segments. Thus, starting from a large number of small segments, the algorithm is grouping them by considering their color differences. Consequently, the used color distance is crucial when we want to obtain small number of segments as provided by the ground truth. We can see in Figure 3.6 that when working in the RGB or $L^*u^*v^*$ color spaces, some segments that are perceptually different are merged while some other similar segments are not. Most of the time, the color mean-shift is working well when using our distance.

3.6 Conclusion

In this chapter, we presented a new local metric learning approach able to approximate a reference distance. Based on a hard clustering of the space, the main idea is to minimize the absolute difference between the learned and the reference distances. Building upon the uniform stability framework we proved that this method is theoretically founded. It is guaranteed to generalize well if a sufficient number of examples is used. We have applied our framework to the problem of perceptual color differences where the idea is to have a metric which is invariant to acquisition conditions but also which is proportional to the human perception of color differences. To this end we proposed a new dataset specifically tailored for

this problem. We have shown that with a sufficient number of clusters our approach allows us to learn a metric which is substantially better than the state of the art distances. Similarly we have demonstrated the interest of the learned metric in a segmentation application.

Even though Figure 3.3 shows ellipsoids that tend to be locally regular leading to a certain spatial continuity, our model does not explicitly deal with this issue. Hence one of the main drawback of the proposed approach is that the learned metric might not be smooth across the space. To deal with this problem we could consider a soft clustering of the space (Semerci and Alpaydın, 2013) where each example is associated to several local metrics depending on its degree of membership to each cluster. It would reduce the risk of having discontinuities in the values of the metric across the space, notably near the borders of the clusters. Another approach, explored for example in Zantedeschi et al. (2016), would consist in learning a smooth combination of the local metrics as an independent post processing step.

In our framework we only considered the Frobenius norm as a regularization term. One interesting perspective would be to consider other regularization terms such as the mixed norm or the nuclear norm. Indeed learning low rank matrices could reduce the computational cost of the metrics, especially in case of a very high dimensional input spaces. One of the drawbacks is that, as we have seen in Section 1.3, algorithms which make use of sparse regularization terms are not stable. It implies that our theoretical analysis would not hold and that other frameworks, such as the Rademacher complexity one, would have to be considered.

In this chapter we have studied the problem of learning an approximation of a reference metric. We assumed to have only access to this reference through its values over a limited number of examples. In the next chapter we consider to have fully access to the reference metric, i.e. we have the parameter matrix of a Mahalanobis distance or a Bilinear Similarity, and we want to use this reference to help learning another metric. More precisely we consider a transfer learning problem where the reference metric is either given or learned on a source domain, i.e. a source metric, and we want to learn a metric on a different but related domain, i.e. a target metric.

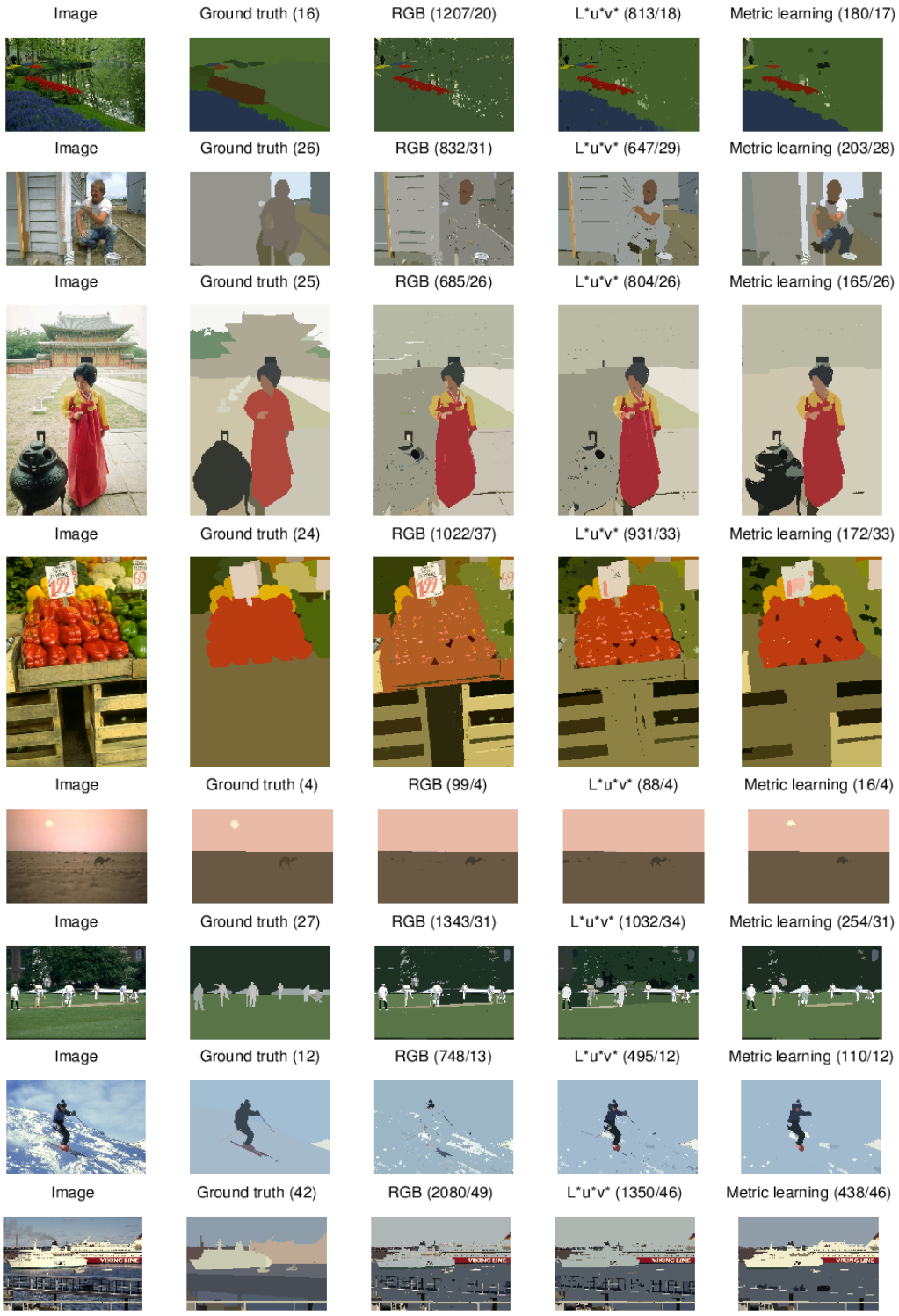


Figure 3.6: Illustration of segmentation provided by the color mean-shift algorithm applied in the RGB components (third column), on $L^*u^*v^*$ components (fourth column) and by using our learned distance directly in the RGB components (fifth column). First column represents the original image and the second one the ground truth.

Chapter 4

Metric Hypothesis Transfer Learning

This chapter is based on the following publication

Michaël Perrot and Amaury Habrard. A theoretical analysis of metric hypothesis transfer learning. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 1708–1717, 2015d

---

Abstract

We consider the problem of transferring some a priori knowledge in the context of supervised metric learning approaches. More precisely we consider biased optimization problems which make use of a source metric coming from a different but related problem to help learning in the presence of few data. While this setting has been shown to be empirically successful, no theoretical evidence exists to justify it. In this chapter we propose to close this gap by providing a theoretical analysis of this framework based on three different approaches. First we propose an on-average-replace-two-stability model allowing us to prove on average fast generalization rates when an auxiliary source metric is used to bias the regularization term. Second we consider a notion of algorithmic stability adapted to the regularized metric learning setting and we prove probabilistic generalization bounds which show the interest of considering biased weighted regularized formulations. We also provide a solution to estimate the associated weight that we evaluate in two experimental tasks (i) standard metric learning and (ii) transfer learning with few labelled target data. Third we derive a generalization bound related to the Rademacher complexity of the metric class taking into account the source metric considered by the algorithm. This vanishing bound underlines the interest of using a good source metric by showing that, when the source metric perfectly solves the problem, learning is no longer a necessity. To justify the interest of the framework we also provide several examples of loss functions and regularization terms which fall under one or more of our theoretical analyses.

4.1 Introduction

Recently, there is a growing interest for methods able to take into account some background knowledge when learning a metric (Parameswaran and Weinberger, 2010; Cao et al., 2013a; Bohné et al., 2014). This is in particular the case for supervised regularized metric learning approaches where the regularization term is biased with respect to an auxiliary metric given under the form of a matrix. The main objective here is to make use of this a priori knowledge in a setting where only few labelled data are available to help learning. For example, in the context of learning a PSD matrix $\mathbf{M}$ plugged into a Mahalanobis-like distance, let $\mathbf{I}$ be the identity matrix used as an auxiliary knowledge, $\|\mathbf{M} - \mathbf{I}\|$ is a biased regularization often considered. This regularization can be interpreted as follows: learn $\mathbf{M}$ while trying to stay close to the Euclidean distance, or from another standpoint try to learn a matrix $\mathbf{M}$ which performs better than $\mathbf{I}$. Other standard matrices can be used such as $\mathbf{\Sigma}^{-1}$ the inverse of the variance-covariance matrix (Mahalanobis, 1936). If we take the $\mathbf{0}$ matrix, we retrieve the classic unbiased regularization setting.

Another useful setting comes when $\mathbf{I}$ is replaced by any auxiliary matrix $\mathbf{M}_{\mathcal{S}}$ learned from another task. It then corresponds to a transfer learning approach (See Section 1.5) where the biased regularization can be interpreted as transferring the knowledge brought by $\mathbf{M}_{\mathcal{S}}$ to help learning $\mathbf{M}$. This setting is appropriate when the distributions over training and testing domains are different but related. Domain adaptation strategies (Ben-David et al., 2010) propose to make use of the relationship between the training examples, called the source domain, and the testing instances, called the target domain to infer a model. However, it is sometimes not possible to have access to all the training examples, for example when some new domains are acquired incrementally. In this context, transferring the information directly from the model learned from the source domain without any other access to the source domain is of crucial importance. We call this setting Metric Hypothesis Transfer Learning in reference to the Hypothesis Transfer Learning model introduced in (Kuzborskij and Orabona, 2013) in the context of classic supervised learning.

If metric hypothesis transfer learning has been shown to work well empirically, it has, to the best of our knowledge, never been investigated from a theoretical standpoint. In this chapter, we propose to bridge this gap by providing a theoretical analysis showing that supervised regularized metric learning approaches using a biased regularization are well-founded. This analysis is based on three different theoretical frameworks which allows us to underline the different properties of biased regularization based algorithms and to derive three measures of goodness of the source metric. The latter quantities are important in the sense that they give a founded way of estimating the interest of a source metric for a particular problem.

- On average stability: The first theoretical framework that we consider is derived from a notion of stability called on-average-replace-one-stability (Ben-David and Urner, 2013). As in other stability frameworks the idea is to verify that small changes in the training does not significantly change the output of the algorithm. The main difference is that

this property is considered on average over all the size n training sets. This approach allows us to derive a bound showing that on average the metric learned with a biased regularization will be better than the source metric with a convergence rate in $\mathcal{O}\left(\frac{1}{n}\right)$. It also implies a first theoretical notion of goodness of the source metric.

- Uniform stability for metric learning: The second theoretical framework that we consider has been proposed by Jin et al. (2009). It corresponds to the uniform stability framework presented in Section 1.3 but adapted to the case of metric learning as shown in Section 2.4. It allows us to derive a probabilistic generalization bound. The main interest of this bound is that, in some cases, it involves an empirical quantity related to the source metric. It implies a natural notion of goodness of the source metric which can be optimized. We empirically confirm the interest of this measure in two experiments.

- Rademacher complexity for metric learning: The third theoretical framework that we consider is a slight adaptation of the one proposed by Cao et al. (2016) (See Section 2.4). The latter is in itself an adaptation to metric learning of the Rademacher complexity framework presented in Section 1.3. It allows us to derive a vanishing bound with respect to the source metric. It means that if the source metric is a perfect fit for the problem at hand the bound shows that learning is no longer necessary. It also gives a theoretical measure of goodness of the source metric.

This chapter is organized as follows. First we present the metric hypothesis transfer learning setting considered here in Section 4.2. Then we present our theoretical analysis based on three frameworks in Sections 4.3, 4.4 and 4.5. We summarize the different bounds in Section 4.6. In Section 4.7 we propose several examples of loss functions and regularization terms that can be used in our framework. Next, in Section 4.8, we empirically demonstrate the interest of using a good source metric as defined in Section 4.4. We conclude in Section 4.9.

4.2 Metric Hypothesis Transfer Learning with Biased Regularization

In this section we present the framework of metric hypothesis transfer learning considered in this chapter.

First of all, let $\mathcal{T}$ be a domain equipped with a probability distribution $\mathcal{D}_{\mathcal{T}}$ defined over $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X} \subseteq \mathbb{R}^d$ and $\mathcal{Y}$ is the label set. Our goal is to learn a metric (as considered in Section 1.4) parametrized by a $d \times d$ matrix $\mathbf{M}$. Let $\mathcal{M}$ be a metric class. Using a slight abuse of notations we denote the fact that a metric is in $\mathcal{M}$ by $\mathbf{M} \in \mathcal{M}$. Here, $\mathcal{M}$ can simply be the set of real matrices of dimension $d \times d$ or can be more restrictive and only consider symmetric positive semi-definite matrices. To learn we consider that we have access to $T = \{\mathbf{z}_i = (\mathbf{x}_i, y_i)\}_{i=1}^n$ a set of n examples drawn i.i.d. from $\mathcal{D}_{\mathcal{T}}$. We also assume that we have access to some additional information under the form of an auxiliary $d \times d$ matrix $\mathbf{M}_{\mathcal{S}}$.

We call this additional information source metric or source hypothesis to denote the fact that, in a transfer learning setting, this metric can come from a different but related source domain $\mathcal{S}$.

We consider all the algorithms learning $\mathbf{M}$ by solving the following optimization problem:

$$\arg\min_{\mathbf{M}\in\mathcal{M}} \hat{L}_T(\mathbf{M}) + \lambda \left\|\mathbf{M} - \mathbf{M}_{\mathcal{S}}\right\|^2 \tag{4.1}$$

where $\left\|\mathbf{M} - \mathbf{M}_{\mathcal{S}}\right\|^2$ is a biased regularization term which ensures that there is a transfer of informations between $\mathbf{M}$ and $\mathbf{M}_{\mathcal{S}}$. In Section 4.7 we consider several regularization terms allowing the transfer of different properties of the metric. λ is a trade-off parameter between risk and regularization. The empirical risk of a metric $\mathbf{M}$ over a training set T is:

$$\hat{L}_T(\mathbf{M}) = \frac{1}{n(n-1)} \sum_{\mathbf{z}\in T} \sum_{\substack{\mathbf{z}'\in T \\ \mathbf{z}\neq\mathbf{z}'}} l\left(\mathbf{M}, \mathbf{z}, \mathbf{z}'\right) \tag{4.2}$$

where $l\left(\mathbf{M}, \mathbf{z}, \mathbf{z}'\right)$ is a pairwise loss quantifying the error of the metric $\mathbf{M}$ when presented with the examples $\mathbf{z}$ and $\mathbf{z}'$. In Section 4.7 we present several loss functions which can be used in metric hypothesis transfer learning. The true risk of $\mathbf{M}$ over the distribution $\mathcal{D}_{\mathcal{T}}$ is:

$$L_{\mathcal{T}}(\mathbf{M}) = \mathop{\mathbb{E}}_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_{\mathcal{T}}} l\left(\mathbf{M}, \mathbf{z}, \mathbf{z}'\right). \tag{4.3}$$

In this chapter we consider a theoretical analysis of this framework of metric hypothesis transfer learning using three different theoretical approaches. As mentioned before several loss functions and regularization terms can be considered. However depending on the theoretical framework considered some assumptions have to be made on these and may differ from one approach to another. Similarly here we only considered a general framework able to handle any metric parametrized by a matrix $\mathbf{M}$ but it might sometimes be necessary to further restrain the range of metrics considered. For the sake of readability we postpone the definition of these different constraints to the beginning of each section.

4.3 On Average Stability Analysis

To derive our first bound for metric hypothesis transfer learning we propose a new notion of stability which is an adaptation to metric learning of the notion of on-average-replace-one-stability (Shalev-Shwartz and Ben-David, 2014a) that we recall in Definition A.5 for the sake of completeness.

Assumptions In this section we make the following assumptions. We only consider metrics as Mahalanobis distances parametrized by a matrix $\mathbf{M}$ positive semi definite. The loss function has to be positive, convex in $\mathbf{M}$ (Definition A.6) and k-lipschitz (Definition A.1). As a last constraint we consider that the regularization term is the Frobenius norm (Section 1.4).

We now turn our attention to the derivation of the bound. First of all we introduce our new notion of on-average-replace-two-stability. Indeed Definition A.5 allows one to perform an on average analysis over the expected loss, however its formulation is not tailored to metric learning approaches that work with pairs of examples. Thus we propose an adaptation allowing us to derive sharp bounds for metric learning.

Definition 4.1 (On-average-replace-two-stability)**.** *Let n be the number of examples considered during the learning step. Let $\epsilon : \mathbb{N} \to \mathbb{R}$ be monotonically decreasing with respect to n and let $U(1,n)$ be the uniform distribution over $1,\dots,n$. A metric learning algorithm is on-average-replace-two-stable with rate $\epsilon(n)$ if for any distribution $\mathcal{D}_\mathcal{T}$:*

$$\mathop{\mathbb{E}}_{\substack{T\sim\mathcal{D}_\mathcal{T}\\ i,j\sim U(1,n)\\ \mathbf{z},\mathbf{z}'\sim\mathcal{D}_\mathcal{T}}} \left[l\left(\mathbf{M}_{T^{ij}},\mathbf{z}_i,\mathbf{z}_j\right) - l\left(\mathbf{M}_T,\mathbf{z}_i,\mathbf{z}_j\right)\right] \leq \epsilon(n) \tag{4.4}$$

where $\mathbf{M}_T$, respectively $\mathbf{M}_{T^{ij}}$, is the optimal solution when learning with the training set T, respectively T^{ij}. T^{ij} is obtained by replacing $\mathbf{z}_i$, the i^{th} example of T, by $\mathbf{z}$ to get a training set T^i and then by replacing $\mathbf{z}_j$, the j^{th} example of T^i, by $\mathbf{z}'$.

This property ensures that, given two examples, learning with or without them will not imply a big change in the hypothesis prediction. Note that the property is required to be true on average over all the possible training sets of size n. Furthermore note that when this definition holds, it implies $\mathbb{E}_{T\sim\mathcal{D}_\mathcal{T}}\left[L_\mathcal{T}(\mathbf{M}_T) - \hat{L}_T(\mathbf{M}_T)\right] \leq \epsilon(n)$. Using this definition we derive a bound on the expected true risk of our algorithm. Before that we show, in the next theorem, that our algorithm is on-average-replace-two-stable.

Lemma 4.1 (On-average-replace-two-stability)**.** *Given n the number of training examples, drawn i.i.d. from $\mathcal{D}_\mathcal{T}$, considered and a k-lipschitz loss function, any algorithm solving Problem* (4.1) *is on-average-replace-two-stable with $\epsilon(n) = \frac{8k^2}{\lambda n}$.*

Proof. The proof of this lemma can be found in Appendix C.1. □

We can now bound the expected true risk of our algorithm.

Theorem 4.1 (On average bound)**.** *For any positive, convex, k-lipschitz loss and for $\mathbf{M}_T$ optimal solution of Problem* (4.1) *when learning with the training set T, we have:*

$$\mathop{\mathbb{E}}_{T\sim\mathcal{D}_\mathcal{T}}\left[L_\mathcal{T}(\mathbf{M}_T)\right] \leq L_\mathcal{T}(\mathbf{M}_\mathcal{S}) + \frac{8k^2}{\lambda n} \tag{4.5}$$

where the expected value is taken over size-n training sets.

Proof. Let T be any training set of size n, we have:

$$\begin{aligned}\mathop{\mathbb{E}}_{T\sim\mathcal{D}_\mathcal{T}}\left[L_\mathcal{T}(\mathbf{M}_T)\right] &= \mathop{\mathbb{E}}_{T\sim\mathcal{D}_\mathcal{T}}\left[L_\mathcal{T}(\mathbf{M}_T)\right] + \mathop{\mathbb{E}}_{T}\left[\hat{L}_T(\mathbf{M}_T)\right] - \mathop{\mathbb{E}}_{T}\left[\hat{L}_T(\mathbf{M}_T)\right]\\ &= \mathop{\mathbb{E}}_{T\sim\mathcal{D}_\mathcal{T}}\left[\hat{L}_T(\mathbf{M}_T)\right] + \mathop{\mathbb{E}}_{T}\left[L_\mathcal{T}(\mathbf{M}_T) - \hat{L}_T(\mathbf{M}_T)\right]\end{aligned}$$

$$\left(\mathbb{E}_{T\sim\mathcal{D}_\mathcal{T}}\left[L_\mathcal{T}(\mathbf{M}_T) - \hat{L}_T(\mathbf{M}_T)\right] \leq \tfrac{8k^2}{\lambda n} \text{ (Lemma 4.1).}\right)$$

$$\leq \mathop{\mathbb{E}}_{T\sim\mathcal{D}_\mathcal{T}}\left[\hat{L}_T(\mathbf{M}_T)\right] + \frac{8k^2}{\lambda n}$$

$$\left(\hat{L}_T(\mathbf{M}_T) \leq \hat{L}_T(\mathbf{M}_T) + \lambda\left\|\mathbf{M}_T - \mathbf{M}_\mathcal{S}\right\|_\mathcal{F}^2 \leq \hat{L}_T(\mathbf{M}_\mathcal{S}) + \lambda\left\|\mathbf{M}_\mathcal{S} - \mathbf{M}_\mathcal{S}\right\|_\mathcal{F}^2.\right)$$

$$\leq \mathop{\mathbb{E}}_{T\sim\mathcal{D}_\mathcal{T}}\left[\hat{L}_T(\mathbf{M}_\mathcal{S})\right] + \frac{8k^2}{\lambda n}.$$

Noting that $\mathbb{E}_{T\sim\mathcal{D}_\mathcal{T}}\left[\hat{L}_T(\mathbf{M}_\mathcal{S})\right] = L_\mathcal{T}(\mathbf{M}_\mathcal{S})$ gives the theorem. □

This bound shows that with a sufficient but rather reasonable number of examples, i.e. with a convergence rate in $\mathcal{O}(1/n)$, we will on average obtain a metric which is at least as good as the source hypothesis. It underlines the interest of having a good source metric. However this notion of goodness depends on the true risk of the source metric on the target domain:

$$G_1(\mathbf{M}_\mathcal{S}) \doteq L_\mathcal{T}(\mathbf{M}_\mathcal{S}).$$

This quantity can not be computed as it depends of the unknown distribution $\mathcal{D}_\mathcal{T}$ and thus it cannot be used to explicitly choose a good metric. Furthermore the stability condition considered here is in expectation over all the possible training sets. It implies that it will probably not be possible to obtain an empirical measure of goodness in this particular setting. In the next section we propose to address this problem and we consider the different but related framework of uniform stability to derive a generalization bound where the goodness of the source metric is empirical and thus can be estimated.

4.4 Uniform Stability Analysis

The second framework that we propose to use to analyse metric hypothesis transfer learning is the uniform stability framework adapted to metric learning presented by Jin et al. (2009) (Section 2.4). In this section we will show that this framework allows us to derive a probabilistic generalization bound where, depending on the loss function, the goodness of the source metric can be empirically estimated.

Assumptions We make the following assumptions. First we only consider metrics as Mahalanobis distances parametrized by a metric $\mathbf{M}$ positive semi definite. Second the loss function has to be convex in $\mathbf{M}$ (Definition A.6), positive, k-lipschitz (Definition A.1) and (σ, m)-admissible (Definition A.2). Third the regularization term is the Frobenius norm (Section 1.4).

We can now present our generalization bound. We divide this section as follows. First we present the bound for general losses. Then we consider a particular example where we show that the goodness of the source metric can be empirically estimated and we deduce

an approach to weight the importance of the source hypothesis in order to obtain a tighter generalization bound.

4.4.1 Generalization Bound for General Loss Functions

We now derive our generalization bound for general loss functions based on the work of (Jin et al., 2009). To this extent we use the McDiarmid's inequality (Theorem A.1) on the estimation error, i.e. the difference between the true risk and the empirical risk. Before that we show that our algorithm is uniformly stable with respect to Definition 2.1 in the next lemma.

Lemma 4.2 (Uniform stability)**.** *Given a positive, convex, k-lipschitz loss and a training sample T of n examples drawn i.i.d. from $\mathcal{D}_{\mathcal{T}}$, an algorithm solving Problem* (4.1) *has a uniform stability in $\beta = \frac{4k^2}{\lambda}$.*

Proof. The proof of this lemma can be found in Appendix C.2. □

Using Lemma 4.2 about the stability of our algorithm and McDiarmid's inequality (Theorem A.1) we can derive our generalization bound. Let $R_T = L_{\mathcal{T}}(\mathbf{M}_T) - \hat{L}_T(\mathbf{M}_T)$ be the estimation error for Problem (4.1) when learning with training set T. To apply McDiarmid's inequality we need to bound $\mathbb{E}_{T\sim\mathcal{D}_{\mathcal{T}}}[R_T]$ and $|R_T - R_{T^i}|$. This is done in the two following lemmas.

Lemma 4.3 (Bound on $\mathbb{E}_{T\sim\mathcal{D}_{\mathcal{T}}}[R_T]$)**.** *For any β uniformly stable learning method of estimation error $R_T = L_{\mathcal{T}}(\mathbf{M}_T) - \hat{L}_T(\mathbf{M}_T)$ and any training set T, we have:*

$$\mathop{\mathbb{E}}_{T\sim\mathcal{D}_{\mathcal{T}}}[R_T] \leq \frac{2\beta}{n}. \tag{4.6}$$

Proof. The proof of this lemma can be found in Appendix C.3. □

This lemma shows that the expected value of the estimation error over all the possible training sets of size n is bounded. In the next lemma we show that the difference in estimation error between two training sets which only vary by one example is also bounded.

Lemma 4.4 (Bound on $|R_T - R_{T^i}|$)**.** *For any β uniformly stable learning method of estimation error $R_T = L_{\mathcal{T}}(\mathbf{M}_T) - \hat{L}_T(\mathbf{M}_T)$, for any training set T and any (σ, m)-admissible loss function we have:*

$$|R_T - R_{T^i}| \leq \frac{2\beta + 4\sigma + 2m}{n}. \tag{4.7}$$

Proof. The proof of this lemma can be found in Appendix C.4. □

Using the fact that our algorithm is uniformly stable, we can now derive generalization guarantees as stated in the following theorem.

Theorem 4.2 (Generalization bound)**.** *For any matrix* $\mathbf{M}_T$ *learned with Problem* (4.1) *with the training set* T *and any positive, convex, k-lipschitz and* (σ, m)*-admissible loss function, we have with probability* $1 - \delta$*:*

$$L_{\mathcal{T}}(\mathbf{M}_T) \leq \hat{L}_T(\mathbf{M}_T) + \frac{2\beta}{n} + (2\beta + 4\sigma + 2m)\sqrt{\frac{\ln\left(\frac{2}{\delta}\right)}{2n}}. \tag{4.8}$$

Proof. Using McDiarmid's inequality (Theorem A.1) on the estimation error $R_T = L_{\mathcal{T}}(\mathbf{M}_T) - \hat{L}_T(\mathbf{M}_T)$ coupled with Lemma 4.4 for the estimation of the constants c_i we have:

$$\begin{aligned}\Pr\left(\left|R_T - \mathop{\mathbb{E}}_{T\sim\mathcal{D}_{\mathcal{T}}}[R_T]\right| \geq \epsilon\right) &\leq 2\exp\left(-\frac{2\epsilon^2}{\sum_{i=1}^{n}\left(\frac{2\beta+4\sigma+2m}{n}\right)^2}\right) \\ &\leq 2\exp\left(-\frac{2\epsilon^2}{\frac{1}{n}(2\beta+4\sigma+2m)^2}\right).\end{aligned}$$

Then, by setting:

$$\delta = 2\exp\left(-\frac{2\epsilon^2}{\frac{1}{n}(2\beta+4\sigma+2m)^2}\right)$$

we obtain:

$$\epsilon = (2\beta + 4\sigma + 2m)\sqrt{\frac{\ln\left(\frac{2}{\delta}\right)}{2n}}$$

and thus:

$$\Pr\left(\left|R_T - \mathop{\mathbb{E}}_{T\sim\mathcal{D}_{\mathcal{T}}}[R_T]\right| < \epsilon\right) > 1 - \delta.$$

Then, with probability $1 - \delta$:

$$R_T < \mathop{\mathbb{E}}_{T\sim\mathcal{D}_{\mathcal{T}}}[R_T] + \epsilon \tag{4.9}$$

$$\Leftrightarrow \quad L_{\mathcal{T}}(\mathbf{M}_T) - \hat{L}_T(\mathbf{M}_T) < \mathop{\mathbb{E}}_{T\sim\mathcal{D}_{\mathcal{T}}}[R_T] + \epsilon \tag{4.10}$$

(Lemma 4.3.)

$$\Rightarrow \quad L_{\mathcal{T}}(\mathbf{M}_T) - \hat{L}_T(\mathbf{M}_T) < \frac{2\beta}{n} + (2\beta + 4\sigma + 2m)\sqrt{\frac{\ln\left(\frac{2}{\delta}\right)}{2n}}. \tag{4.11}$$

□

This bound shows that with a convergence rate in $\mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$ the true risk of our algorithm is bounded above by its empirical risk. In the next section, we consider a particular example of loss function where we show that the goodness of the source metric can be empirically estimated. This extension allows us to introduce a natural weighting of the source metric in order to improve the proposed bound.

4.4.2 Refinement with Weighted Source Hypothesis

In this section we propose to study the problem of weighting the source metric to improve the generalization bound. However in its current form the generalization bound does not explicitly include any information about the source metric. In the following we notice that, for a particular loss, the goodness of the source metric appears in the (σ, m)-admissibility of this loss.

First of all we consider the loss presented in Example 4.1) instantiated with the hinge function. For any metric $\mathbf{M}$ and any two labelled examples $\mathbf{z}, \mathbf{z}' \sim \mathcal{D}_\mathcal{T}$ we have:

$$l\left(\mathbf{M}, \mathbf{z}, \mathbf{z}'\right) = \left[\delta_{yy'}\left((\mathbf{x}-\mathbf{x}')^T\mathbf{M}(\mathbf{x}-\mathbf{x}') - \gamma_{yy'}\right)\right]_+ \tag{4.12}$$

where $\delta_{yy'} = 1$ if $y = y'$ and -1 otherwise and $\gamma_{yy'}$ is the desired margin between the examples. In Example 4.1, given that the hinge loss is 1-lipschitz, we show that this loss is positive, convex, k-lipschitz with:

$$k = \sup_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_\mathcal{T}} \left\|\mathbf{x}-\mathbf{x}'\right\|_2^2$$

and (σ, m)-admissible with:

$$\sigma = \sup_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_\mathcal{T}} \gamma_{yy'},$$

$$m = 2 \sup_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_\mathcal{T}} \left\|\mathbf{x}-\mathbf{x}'\right\|_2^2 \left(\sqrt{\frac{\hat{L}_T(\mathbf{M}_\mathcal{S})}{\lambda}} + \|\mathbf{M}_\mathcal{S}\|_\mathcal{F}\right).$$

Using this we can now apply Theorem 4.2, to obtain a generalization bound which includes a measure of the goodness of the source metric.

Theorem 4.3 (Generalization bound with Loss (4.12))**.** *For any matrix $\mathbf{M}_T$ learned with Problem* (4.1) *with the training set T and Loss* (4.12)*, we have with probability $1-\delta$:*

$$L_\mathcal{T}(\mathbf{M}_T) \leq \hat{L}_T(\mathbf{M}_T) + \frac{8D^2}{\lambda n} + \left(\frac{8D^2}{\lambda} + 4\sup_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_\mathcal{T}} \gamma_{yy'} + 4D\left(\sqrt{\frac{\hat{L}_T(\mathbf{M}_\mathcal{S})}{\lambda}} + \|\mathbf{M}_\mathcal{S}\|_\mathcal{F}\right)\right)\sqrt{\frac{\ln\left(\frac{2}{\delta}\right)}{2n}}.$$

where $D = \sup_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_\mathcal{T}} \|\mathbf{x}-\mathbf{x}'\|_2^2$.

Proof. This theorem comes from the application of Theorem 4.2 with specific values of k, σ and m. □

This theorem is a refinement of Theorem 4.2 in the case of a specific loss. Hence the convergence rate is still in $\mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$ and one of the most important difference is the presence of the term:

$$G_2(\mathbf{M}_\mathcal{S}) \doteq \left(\sqrt{\frac{\hat{L}_T(\mathbf{M}_\mathcal{S})}{\lambda}} + \|\mathbf{M}_\mathcal{S}\|_\mathcal{F}\right).$$

This term can be seen as a measure of the goodness of the source metric. It mainly depends on the quality of the source hypothesis $\mathbf{M}_{\mathcal{S}}$. The product $G_2(\mathbf{M}_{\mathcal{S}})\sqrt{\frac{\ln\left(\frac{2}{\delta}\right)}{2n}}$ which appears in the bound implies that as the number of examples available for learning increases, the quality of the source metric is of decreasing importance. A similar result has already been stated in domain adaptation or transfer learning in Ben-David et al. (2010); Kuzborskij and Orabona (2013) where they show that as the number of target examples increases, the necessity of having source examples decreases.

Given a source hypothesis $\mathbf{M}_{\mathcal{S}}$, it is possible to optimize it with respect to the bound derived in Theorem 4.3. Indeed, note that $G_2(\mathbf{M}_{\mathcal{S}})$ corresponds to a trade-off between the complexity of the source metric and its performance on the training set. The lower the value of this term, the tighter the bound.

Following this, we propose a way to minimize the right hand side of the generalization bound and more specifically $G_2(\mathbf{M}_{\mathcal{S}})$ by adding a weighting parameter $0 \leq \omega_T \leq 1$ on the source metric $\mathbf{M}_{\mathcal{S}}$. This parameter is a simple way to control the trade-off between complexity and performance of the source metric thanks to a reweighting. It can be assessed by means of the following optimization problem:

$$\omega_T = \underset{0\leq\omega\leq 1}{\arg\min}\, C(\omega\mathbf{M}_{\mathcal{S}}) \tag{4.13}$$

Note that the bound derived in Theorem 4.3 holds whatever the value of $\mathbf{M}_{\mathcal{S}}$. Thus replacing it with $\omega_T\mathbf{M}_{\mathcal{S}}$ does not impact the theoretical study proposed in this section.

Interpretation of the value of ω_T We can distinguish three main cases. First if the source hypothesis performs poorly on the training set at hand we expect ω_T to be as small as possible to reduce the importance of $\mathbf{M}_{\mathcal{S}}$. In a sense, we tend to go back to the classical case were $\mathbf{M}_{\mathcal{S}} = \mathbf{0}$. Second if the source hypothesis is complex and performs well, we expect ω_T to be rather small to reduce the complexity of the hypothesis while keeping a good performance on the training set. Third if the source hypothesis is simple and performs well, we expect ω_T to be closer to one since $\mathbf{M}_{\mathcal{S}}$ is already a good choice. Note that we choose $\omega_T \leq 1$ to limit the potential increase in complexity of the learned matrix.

Learning ω_T Problem (4.13) is highly non differentiable[1] and non convex. However, it remains simple in the sense that we have only one parameter to estimate and we used a classical sub-gradient descent to solve it. Even if it is not convex, our empirical study (Section 4.8 shows no need to perform many restarts to output a good solution: we always found almost the same solution. As a consequence, we applied only one optimization procedure in our experiments. Note that ω_T is influenced by both the values of the margin and the regularization parameter and thus should be tuned accordingly each time. However, computing ω_T by solving Problem (4.13) is not too costly. The process can even be sped up by computing the value of the source metric between the examples beforehand.

[1]To avoid this problem, we can use the classic relaxation with slack variables.

In this section, using the uniform stability framework for metric learning we have shown that our approach generalizes well with a convergence rate in $\mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$. Furthermore, given a specific loss, we have shown that the use of a weighting parameter to control the importance of the source metric is theoretically founded. Indeed it boils down to optimizing a notion of goodness of the source metric for the problem at hand. However the right hand side of the bound derived in Theorem 4.3 does not go to 0 when the source metric is a perfect fit for the problem at hand, i.e. it is not a vanishing bound. It implies that even with the perfect source hypothesis we might learn a metric which performs slightly worse than the source metric in terms of generalization to new examples. In the next section we consider the Rademacher complexity framework to derive a vanishing generalization bound when the source metric is a perfect fit.

4.5 Rademacher Complexity Analysis

The third theoretical framework that we consider in this chapter is the Rademacher complexity framework adapted to metric learning by Cao et al. (2016). More precisely we further adapt this approach to take into account the source metric. It allows us to obtain a vanishing generalization bound which implies that when the source metric is a perfect fit, learning is no longer necessary.

Assumptions In this section instead of only considering the Mahalanobis distance we consider all the metrics parametrized by a matrix $\mathbf{M} \in \mathcal{M}$, and denoted by the function $k_{\mathbf{M}}$, such that given two vectors $\mathbf{x}$ and $\mathbf{x}'$ the metric can be written as:

$$k_{\mathbf{M}}(\mathbf{x}, \mathbf{x}') = \langle g(\mathbf{x}, \mathbf{x}'), \mathbf{M} \rangle \tag{4.14}$$

where g is a function over a pair of examples and $\langle \cdot, \cdot \rangle$ is the Frobenius product[2]. If this definition seems restrictive, it is in fact fulfilled by the main metrics used in Metric Learning (See Section 2.1) such as the squared Mahalanobis distance:

$$(\mathbf{x} - \mathbf{x}')^T \mathbf{M} (\mathbf{x} - \mathbf{x}') = \left\langle (\mathbf{x} - \mathbf{x}')(\mathbf{x} - \mathbf{x}')^T, \mathbf{M} \right\rangle, \tag{4.15}$$

or the bilinear similarity:

$$\mathbf{x}^T \mathbf{M} \mathbf{x}' = \left\langle \mathbf{x}\mathbf{x}'^T, \mathbf{M} \right\rangle. \tag{4.16}$$

The loss function has to be positive, convex (Definition A.6) and k-lipschitz with respect to the metric (Definition 4.2) which is a slight adaptation of the k-lipschitzness presented in Definition A.1. The regularization term has to be convex and must have a well defined dual norm (Definition A.4).

We can now derive our generalization bound based on the Rademacher complexity framework. First of all we present our notion of k-lipschitz continuity with respect to the metric.

[2] $\langle \mathbf{A}, \mathbf{B} \rangle = \mathrm{Tr}(\mathbf{A}^T \mathbf{B})$

Then we present the notion of Rademacher complexity considered here along with a refinement of the metric class $\mathcal{M}$ considered. These definitions allow us to state our generalization bound linked to the Rademacher complexity of the metric class. Next we bound this Rademacher complexity showing that the bound depends on the source metric. We then discuss the implications of the bound and we show that if the source metric $\mathbf{M}_{\mathcal{S}}$ is a good fit then the rate of convergence is improved.

Definition 4.2 (k-lipschitz continuity with respect to the metric)**.** *A loss function $l\left(\mathbf{M}, \mathbf{z}, \mathbf{z}'\right)$ is k-lipschitz continuous with respect to the metric if for any two matrices $\mathbf{M}, \mathbf{M}' \in \mathcal{M}$ and any two examples $\mathbf{z}, \mathbf{z}'$ there exists $k \geq 0$ such that:*

$$\left|l\left(\mathbf{M}, \mathbf{z}, \mathbf{z}'\right) - l\left(\mathbf{M}', \mathbf{z}, \mathbf{z}'\right)\right| \leq k\left|k_{\mathbf{M}}\left(\mathbf{x}, \mathbf{x}'\right) - k_{\mathbf{M}'}\left(\mathbf{x}, \mathbf{x}'\right)\right| \tag{4.17}$$

This k-lipschitz continuity property ensures that given two metrics the difference in losses is bounded by the difference between the metrics. If the loss is differentiable with respect to the metric it can also be seen as a bound on the magnitude of the first derivative Srebro et al. (2010a). Note that the k-lipschitz property is usually considered with respect to the parameters of the metric rather than the metric as it is the case here. However our definition lipschitzness implies the standard definition when metrics of the form presented in Equation (4.14) are considered.

The Rademacher complexity used here is an adaptation of the definition given in Lei and Ying (2015) to the case of metric hypothesis transfer learning. The idea is to take into account the source metric.

Definition 4.3 (Rademacher Complexity)**.** *Let $\mathcal{M}$ be a metric class and $\left\{\sigma_i : i = 1, \cdots, \left\lfloor \frac{n}{2} \right\rfloor\right\}$ be a sequence of independent Rademacher variables, that is, $\Pr(\sigma_i = +1) = \Pr(\sigma_i = -1) = \frac{1}{2}$. Let $\{\mathbf{x}_i : i = 1, \cdots, n\}$ be an i.i.d. sequence of examples. Then the empirical Rademacher complexity of $\mathcal{M}$ is defined as:*

$$\hat{R}_n(\mathcal{M}) \doteq \frac{1}{\left\lfloor \frac{n}{2} \right\rfloor} \mathop{\mathbb{E}}_{\sigma} \sup_{\mathbf{M} \in \mathcal{M}} \sum_{i=1}^{\left\lfloor \frac{n}{2} \right\rfloor} \sigma_i k_{\mathbf{M}-\mathbf{M}_{\mathcal{S}}}\left(\mathbf{x}_i, \mathbf{x}_{\left\lfloor \frac{n}{2} \right\rfloor + i}\right) \tag{4.18}$$

and the Rademacher Complexity of $\mathcal{M}$ as:

$$R_n(\mathcal{M}) = \mathop{\mathbb{E}}_{T \sim \mathcal{D}_{\mathcal{T}}} \hat{R}_n(\mathcal{M}). \tag{4.19}$$

Instead of considering the complexity of a metric class with respect to its ability to fit random noise Lei and Ying (2015), this definition measures the complexity of the metric class with respect to its ability to differ from the source metric.

We now define formally the refinement of the metric class considered in our analysis and taking into account the source metric.

Definition 4.4 (Metric class and source metric)**.** *We define a metric class dependent on $\mathbf{M}_{\mathcal{S}}$ as follows:*

$$\mathcal{M}_{\mathcal{S}} = \left\{\mathbf{M} \in \mathcal{M} : \|\mathbf{M} - \mathbf{M}_{\mathcal{S}}\| \leq \sqrt{\frac{G_3(\mathbf{M}_{\mathcal{S}})}{\lambda}}\right\} \tag{4.20}$$

where $G_3(\mathbf{M}_\mathcal{S}) = \sup_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_\mathcal{T}} l\left(\mathbf{M}_\mathcal{S}, \mathbf{z}, \mathbf{z}'\right)$ *and* $\mathcal{M}$ *is the metric class used in Problem* (4.1).

We now prove that any metric $\mathbf{M}_T$ learned by Problem (4.1) belongs to $\mathcal{M}_\mathcal{S}$.

Lemma 4.5 (Metric class and optimal solution)**.** *Let* $\mathbf{M}_T$ *be the optimal solution returned by Problem* (4.1) *with training set* T*. We have* $\mathbf{M}_T \in \mathcal{M}_\mathcal{S}$ *where* $\mathcal{M}_\mathcal{S}$ *is defined as in Definition 4.4.*

Proof. By the convexity of the loss and the optimality of $\mathbf{M}_T$ we have:

$$\hat{L}_T(\mathbf{M}_T) + \lambda \left\|\mathbf{M}_T - \mathbf{M}_\mathcal{S}\right\|^2 \leq \hat{L}_T(\mathbf{M}_\mathcal{S}) \qquad \text{(Positive loss.)}$$

$$\Rightarrow \quad \lambda \left\|\mathbf{M}_T - \mathbf{M}_\mathcal{S}\right\|^2 \leq \hat{L}_T(\mathbf{M}_\mathcal{S})$$

$$\Rightarrow \quad \left\|\mathbf{M}_T - \mathbf{M}_\mathcal{S}\right\| \leq \sqrt{\frac{\hat{L}_T(\mathbf{M}_\mathcal{S})}{\lambda}} \qquad (\hat{L}_T(\mathbf{M}_\mathcal{S}) \leq \sup_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_\mathcal{T}} l\left(\mathbf{M}_\mathcal{S}, \mathbf{z}, \mathbf{z}'\right).)$$

$$\Rightarrow \quad \left\|\mathbf{M}_T - \mathbf{M}_\mathcal{S}\right\| \leq \sqrt{\frac{G_3(\mathbf{M}_\mathcal{S})}{\lambda}}$$

with $G_3(\mathbf{M}_\mathcal{S}) = \sup_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_\mathcal{T}} l\left(\mathbf{M}_\mathcal{S}, \mathbf{z}, \mathbf{z}'\right)$. Noting that $\mathbf{M}_T \in \mathcal{M}$ gives the lemma. □

To prove our generalization bound based on the Rademacher complexity of the source metric, we use the McDiarmid's inequality (Theorem A.1) and follow a similar strategy as in Cao et al. (2016). Let $R_T = \sup_{\mathbf{M}\in\mathcal{M}_\mathcal{S}} \left[L_\mathcal{T}(\mathbf{M}) - \hat{L}_T(\mathbf{M})\right]$ be the estimation error for Problem (4.1) when learning with training set T. Note that this estimation error is different from the one used in the previous sections. Indeed here we consider the worst error over the whole metric class rather than the error of the learned hypothesis. To apply McDiarmid's inequality we need to bound $\mathbb{E}_{T\sim\mathcal{D}_\mathcal{T}}\left[R_T\right]$ and $|R_T - R_{T^i}|$. This is done in the two following lemmas.

Lemma 4.6 (Bound on $\mathbb{E}_{T\sim\mathcal{D}_\mathcal{T}}\left[R_T\right]$)**.** *For any positive, convex and k-lipschitz (Definition 4.2) loss function and any algorithm with estimation error* $R_T = \sup\limits_{\mathbf{M}\in\mathcal{M}_\mathcal{S}} \left[L_\mathcal{T}(\mathbf{M}) - \hat{L}_T(\mathbf{M})\right]$ *we have:*

$$\mathop{\mathbb{E}}_{T\sim\mathcal{D}_\mathcal{T}}\left[R_T\right] \leq 2kR_n(\mathcal{M}_\mathcal{S}).$$

Proof. The proof of this lemma can be found in Appendix C.5. □

This lemma shows that the expected value of the estimation error over all the possible training sets of size n is bounded. In the next lemma we show that the difference in estimation error between two training sets which only vary by one example is also bounded.

Lemma 4.7 (Bound on $|R_T - R_{T^i}|$)**.** *For any positive, convex and k-lipschitz continuous (Definition 4.2) loss function, any metric satisfying Equation* (4.14) *and any algorithm of estimation error* $R_T = \sup_{\mathbf{M}\in\mathcal{M}_\mathcal{S}} \left[L_\mathcal{T}(\mathbf{M}) - \hat{L}_T(\mathbf{M})\right]$ *we have:*

$$|R_T - R_{T^i}| \leq \frac{2G_3(\mathbf{M}_\mathcal{S}) + 2\sup_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_\mathcal{T}} \left[k \left\|g(\mathbf{x},\mathbf{x}')\right\|_* \sqrt{\frac{G_3(\mathbf{M}_\mathcal{S})}{\lambda}}\right]}{n}$$

where $\|\cdot\|_*$ *is the dual norm of the regularization term (Definition A.4).*

Proof. The proof of this lemma can be found in Appendix C.6. □

We can now present our generalization bound.

Theorem 4.4 (Generalization bound)**.** *With probability* $1-\delta$*, for any matrix* $\mathbf{M}_T$ *learned with Problem* (4.1)*, for any positive, convex, and k-lipschitz continuous (Definition 4.2) loss function and any metric satisfying Equation* (4.14) *we have:*

$$L_\mathcal{T}(\mathbf{M}_T) \leq \hat{L}_T(\mathbf{M}_T) + 2kR_n(\mathcal{M}_\mathcal{S}) + \left(2G_3(\mathbf{M}_\mathcal{S}) + 2\sup_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_\mathcal{T}} \left[k \left\|g(\mathbf{x},\mathbf{x}')\right\|_* \sqrt{\frac{G_3(\mathbf{M}_\mathcal{S})}{\lambda}}\right]\right)\sqrt{\frac{\ln\frac{2}{\delta}}{2n}}.$$

Proof. Coupling Lemma 4.7 with McDiarmid's inequality (Theorem A.1) applied on the estimation error $R_T = \sup_{\mathbf{M}\in\mathcal{M}_\mathcal{S}} \left[L_\mathcal{T}(\mathbf{M}) - \hat{L}_T(\mathbf{M})\right]$ we have with probability $1-\delta$:

$$R_T \leq \mathop{\mathbb{E}}_{T\sim\mathcal{D}_\mathcal{T}}[R_T] + \left(2G_3(\mathbf{M}_\mathcal{S}) + 2\sup_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_\mathcal{T}} \left[k \left\|g(\mathbf{x},\mathbf{x}')\right\|_* \sqrt{\frac{G_3(\mathbf{M}_\mathcal{S})}{\lambda}}\right]\right)\sqrt{\frac{\ln\frac{2}{\delta}}{2n}}$$

(Lemma 4.5.)

$$L_\mathcal{T}(\mathbf{M}_T) - \hat{L}_T(\mathbf{M}_T) \leq \mathop{\mathbb{E}}_{T\sim\mathcal{D}_\mathcal{T}}[R_T] + \left(2G_3(\mathbf{M}_\mathcal{S}) + 2\sup_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_\mathcal{T}} \left[k \left\|g(\mathbf{x},\mathbf{x}')\right\|_* \sqrt{\frac{G_3(\mathbf{M}_\mathcal{S})}{\lambda}}\right]\right)\sqrt{\frac{\ln\frac{2}{\delta}}{2n}}$$

(Lemma 4.6.)

$$L_\mathcal{T}(\mathbf{M}_T) - \hat{L}_T(\mathbf{M}_T) \leq 2kR_n(\mathcal{M}_\mathcal{S}) + \left(2G_3(\mathbf{M}_\mathcal{S}) + 2\sup_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_\mathcal{T}} \left[k \left\|g(\mathbf{x},\mathbf{x}')\right\|_* \sqrt{\frac{G_3(\mathbf{M}_\mathcal{S})}{\lambda}}\right]\right)\sqrt{\frac{\ln\frac{2}{\delta}}{2n}}.$$

□

This generalization bound shows that the generalization ability of a metric learned with Problem (4.1) depends on the Rademacher complexity of the source metric class and on $G_3(\mathbf{M}_\mathcal{S})$, its worst possible error over the distribution. In the next subsection we show that the Rademacher complexity of the source metric class also depends on $G_3(\mathbf{M}_\mathcal{S})$. This value is thus a good candidate to measure the goodness of the source metric.

4.5.1 Rademacher Complexity and Source Metric

One of the critical quantities in the bound presented in Theorem 4.4 is the Rademacher complexity of the source metric class $\mathcal{M}_\mathcal{S}$ which depends on the source metric $\mathbf{M}_\mathcal{S}$. In the next lemma we show that the Rademacher complexity of the source metric class depends on $G_3(\mathbf{M}_\mathcal{S})$.

Lemma 4.8 (Bounding the Rademacher complexity of $\mathcal{M}_{\mathcal{S}}$)**.** *Let $\mathcal{M}_{\mathcal{S}}$ be a metric class which depends on a source metric $\mathbf{M}_{\mathcal{S}}$ as in Definition 4.4 then we have that:*

$$R_n(\mathcal{M}_{\mathcal{S}}) \leq \sqrt{\frac{G_3(\mathbf{M}_{\mathcal{S}})}{\lambda}} R_n(\|\cdot\|_*)$$

with:

$$R_n(\|\cdot\|_*) = \mathop{\mathbb{E}}_{T\sim\mathcal{D}_{\mathcal{T}}} \frac{1}{\lfloor \frac{n}{2} \rfloor} \mathop{\mathbb{E}}_{\boldsymbol{\sigma}} \left\| \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} \sigma_i g(\mathbf{x}_i, \mathbf{x}_{\lfloor \frac{n}{2} \rfloor + i}) \right\|_* \tag{4.21}$$

where $\|\cdot\|_*$ is the dual norm of the regularization term (Definition A.4).

Proof. We consider the empirical Rademacher complexity of the metric class $\mathcal{M}_{\mathcal{S}}$:

$$\begin{aligned}
\hat{R}_n(\mathcal{M}_{\mathcal{S}}) &= \frac{1}{\lfloor \frac{n}{2} \rfloor} \mathop{\mathbb{E}}_{\sigma} \sup_{\mathbf{M}\in\mathcal{M}_{\mathcal{S}}} \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} \sigma_i k_{\mathbf{M}-\mathbf{M}_{\mathcal{S}}} \left(\mathbf{x}_i, \mathbf{x}_{\lfloor \frac{n}{2} \rfloor + i} \right) \\
&\qquad \text{(Equation (4.14).)} \\
&\leq \frac{1}{\lfloor \frac{n}{2} \rfloor} \mathop{\mathbb{E}}_{\sigma} \sup_{\mathbf{M}\in\mathcal{M}_{\mathcal{S}}} \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} \sigma_i \left\langle g(\mathbf{x}_i, \mathbf{x}_{\lfloor \frac{n}{2} \rfloor + i}), \mathbf{M} - \mathbf{M}_{\mathcal{S}} \right\rangle \\
&\qquad \text{(Trace linearity.)} \\
&\leq \frac{1}{\lfloor \frac{n}{2} \rfloor} \mathop{\mathbb{E}}_{\sigma} \sup_{\mathbf{M}\in\mathcal{M}_{\mathcal{S}}} \left\langle \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} \sigma_i g(\mathbf{x}_i, \mathbf{x}_{\lfloor \frac{n}{2} \rfloor + i}), \mathbf{M} - \mathbf{M}_{\mathcal{S}} \right\rangle \\
&\qquad \text{(Cauchy-Schwarz's inequality (Theorem A.3).)} \\
&\leq \frac{1}{\lfloor \frac{n}{2} \rfloor} \mathop{\mathbb{E}}_{\sigma} \sup_{\mathbf{M}\in\mathcal{M}_{\mathcal{S}}} \left\| \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} \sigma_i g(\mathbf{x}_i, \mathbf{x}_{\lfloor \frac{n}{2} \rfloor + i}) \right\|_* \|\mathbf{M} - \mathbf{M}_{\mathcal{S}}\| \\
&\qquad (\mathbf{M} \in \mathcal{M}_{\mathcal{S}} \text{ (Lemma 4.4).)} \\
&\leq \sqrt{\frac{G_3(\mathbf{M}_{\mathcal{S}})}{\lambda}} \frac{1}{\lfloor \frac{n}{2} \rfloor} \mathop{\mathbb{E}}_{\sigma} \left\| \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} \sigma_i g(\mathbf{x}_i, \mathbf{x}_{\lfloor \frac{n}{2} \rfloor + i}) \right\|_*
\end{aligned}$$

Taking the expectation over all size n training sets on both sides of the last inequality gives:

$$R_n(\mathcal{M}_{\mathcal{S}}) \leq \sqrt{\frac{G_3(\mathbf{M}_{\mathcal{S}})}{\lambda}} \mathop{\mathbb{E}}_{T\sim\mathcal{D}_{\mathcal{T}}} \frac{1}{\lfloor \frac{n}{2} \rfloor} \mathop{\mathbb{E}}_{\sigma} \left\| \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} \sigma_i g(\mathbf{x}_i, \mathbf{x}_{\lfloor \frac{n}{2} \rfloor + i}) \right\|_* .$$

Setting $R_n(\|\cdot\|_*) = \mathop{\mathbb{E}}_{T\sim\mathcal{D}_{\mathcal{T}}} \frac{1}{\lfloor \frac{n}{2} \rfloor} \mathop{\mathbb{E}}_{\sigma} \left\| \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} \sigma_i g(\mathbf{x}_i, \mathbf{x}_{\lfloor \frac{n}{2} \rfloor + i}) \right\|_*$ gives the lemma. □

Rademacher complexity and source metric $\mathbf{M}_\mathcal{S}$. We have that $\mathbf{M}_\mathcal{S} \in \mathcal{M}_\mathcal{S}$ and the metric class is centred[3] around $\mathbf{M}_\mathcal{S}$ with the radius being dependent on the worst case performance of $\mathbf{M}_\mathcal{S}$, i.e. $G_3(\mathbf{M}_\mathcal{S})$. It means that changing $\mathbf{M}_\mathcal{S}$ will impact the metric class and, consequently, its Rademacher complexity. On the one hand if the source metric is good, i.e. if $G_3(\mathbf{M}_\mathcal{S})$ is low, then the considered metrics cannot go too far away from the source and thus the Rademacher complexity will be small. On the other hand if the source metric is bad, i.e. if $G_3(\mathbf{M}_\mathcal{S})$ is high, then the considered metrics can go far away from the source and thus the Rademacher complexity will be higher.

We have shown how the source metric impacts the Rademacher complexity of the metric class and that $G_3(\mathbf{M}_\mathcal{S})$ is a measure of goodness of the source metric. We now study the impact of the source metric on the rate of convergence of the bound presented in Theorem 4.4.

4.5.2 Goodness of $\mathbf{M}_\mathcal{S}$

In this subsection we propose to study the bound presented in Theorem 4.4 with respect to the source metric $\mathbf{M}_\mathcal{S}$. We show that when this source metric is good then it has a positive impact on the rate of convergence of the bound. In some case it might result in a faster rate of convergence in $\mathcal{O}\left(\frac{1}{n}\right)$. We also study the impact of the source metric on the number of examples needed to obtain a true risk at most equal to the empirical risk plus ϵ.

First of all note that the key quantity related to the source metric in the bound is:

$$G_3(\mathbf{M}_\mathcal{S}) \doteq \sup_{\mathbf{z},\mathbf{z}' \sim \mathcal{D}_\mathcal{T}} l(\mathbf{M}_\mathcal{S}, \mathbf{z}, \mathbf{z}').$$

We will use this quantity to define the goodness of a source metric. If it is small, respectively large, then the metric is good, respectively bad, for the problem at hand. This quantity represents the worst case loss of the source metric over all the examples.

Source metric and ϵ convergence. We consider the case where we want $L_\mathcal{T}(\mathbf{M}_T) \leq \hat{L}_T(\mathbf{M}_T) + \epsilon$ and we seek n the number of examples needed to obtain such an ϵ convergence. Let $m \geq 1$ such that $G_3(\mathbf{M}_\mathcal{S}) \leq \frac{1}{m}$ and define $\alpha \in \mathbb{R}$ such that $R_n(\|\cdot\|_*) \leq \frac{\alpha}{\sqrt{n}}$ then the bound in Theorem 4.4 gives:

$$\epsilon = \frac{1}{\sqrt{nm}} \left(\frac{2k\alpha}{\sqrt{\lambda}} + \frac{2\sqrt{\ln \frac{2}{\delta}}}{\sqrt{2}} \left(1 + \sup_{\mathbf{z},\mathbf{z}' \sim \mathcal{D}_\mathcal{T}} \frac{2k \left\| g(\mathbf{x}, \mathbf{x}') \right\|_*}{\sqrt{\lambda}} \right) \right) \tag{4.22}$$

which implies:

$$nm = \frac{1}{\epsilon^2} \left(\frac{2k\alpha}{\sqrt{\lambda}} + \frac{2\sqrt{\ln \frac{2}{\delta}}}{\sqrt{2}} \left(1 + \sup_{\mathbf{z},\mathbf{z}' \sim \mathcal{D}_\mathcal{T}} \frac{2k \left\| g(\mathbf{x}, \mathbf{x}') \right\|_*}{\sqrt{\lambda}} \right) \right)^2. \tag{4.23}$$

[3] With respect to the norm considered.

The quantity in the out-most brackets is a constant with respect to n, m and ϵ. It shows that a smaller ϵ can be obtained by increasing the number of examples or by using a better source metric. We can then consider several cases:

- $m \geq n$: In this case the source metric is a good fit since $G_3(\mathbf{M}_\mathcal{S}) \leq \frac{1}{n}$ and the bound exhibits a fast rate of convergence with $\epsilon \leq \mathcal{O}\left(\frac{1}{n}\right)$. However this result is not fully informative in the sense that the constraint imposed on the source metric can be stronger than the bound, i.e. it might be better to directly use the source metric than to learn a new metric.

- $m \to \infty$: In this case the source metric is a perfect fit, i.e. $G_3(\mathbf{M}_\mathcal{S}) = 0$ and by convexity of the loss, the metric learned by Problem (4.1) is the source metric. It implies that the empirical risk is equal to 0 and since $\epsilon \to 0$ the right hand side of the bound also tends to 0 which implies that the bound is vanishing and translates the fact that no learning is necessary.

- $m < n$: In this case we have $G_3(\mathbf{M}_\mathcal{S}) > \frac{1}{n}$ the bound is not worse than classic generalization bounds for metric learning as it exhibits a convergence rate ϵ in $\mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$.

Note that here we chose to bound the goodness of the source metric by a quantity which depends on the number of examples. It might be surprising in the sense that this measure is a constant with respect to $\mathbf{M}_\mathcal{S}$. However it reflects the fact that when one has access to a sufficient number of examples, it is harder to obtain a meaningful source metric.

Comparison with similar bounds. Albeit not in the context of metric learning, the work presented in Kuzborskij and Orabona (2014) presents a generalization bound which is close to ours[4]. However, the condition obtained to derive a fast rate is different. In this work, they propose to bound by $\mathcal{O}\left(\frac{1}{n}\right)$ the true risk of the source hypothesis rather than the worst case loss. It might seem less restrictive as it considers the whole distribution and, thus, makes it easier to have a good source hypothesis. However it is important to note that, besides the fast rate, our condition allows us to guarantee that the empirical risk of the learned metric will be small. Indeed our condition implies that the error of the source metric will be low on the training set, then by convexity of the loss, the learned metric will have a better performance than the source metric. Such an analysis is not possible if we link the goodness of the source hypothesis to its true risk as it might happen that we have access to a training set where the empirical risk of the source hypothesis is greater than its true risk. Note also that their framework is more restrictive than ours since we allow to deal with possibly non smooth lispschitz functions and relaxes the strong convexity. While their result uses standard supervised learning losses, one question is to know if we can derive similar results for metric learning with pairwise losses. We cannot provide a clear answer but this issue is

[4]To the best of our knowledge it is the only work with a fast rate in an Hypothesis Transfer Learning setting.

Table 4.1: Summary of the different bounds.

Bound	On Average Stability (Section 4.3)	Uniform Stability (Section 4.4)	Rademacher Complexity (Section 4.5)
Nature of the bound	Exact	Probabilistic $(1-\delta)$	Probabilistic $(1-\delta)$
Impact of the complexity of the metric class	$\frac{8k^2}{\lambda n}$	$\frac{8k^2}{\lambda n}$	$2kR_n(\mathcal{M}_\mathcal{S}) \leq \mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$
Convergence rate	$\mathcal{O}\left(\frac{1}{n}\right)$	$\mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$	$\mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$
Goodness	Theoretical	Empirical	Theoretical
$G_i(\mathbf{M}_\mathcal{S})$	$L_\mathcal{T}(\mathbf{M}_\mathcal{S})$	$\sqrt{\frac{\hat{L}_T(\mathbf{M}_\mathcal{S})}{\lambda}} + \|\mathbf{M}_\mathcal{S}\|_\mathcal{F}$	$\sup_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_\mathcal{T}} l(\mathbf{M}_\mathcal{S}, \mathbf{z}, \mathbf{z}')$

not straightforward. Recall that if the examples are drawn i.i.d. from a distribution, pairs of examples are not i.i.d. and as such the proof techniques have to be adapted to take this problem into account. Their analysis is based on a generalization of Bennett's concentration inequality Bousquet (2002) which requires to have some informations about the variance of the studied random variable. Its application to pairwise losses seems difficult and would require a specific study. In our framework, we used the McDiarmid's concentration inequality which does not require any information about the variance of the studied random variable but rather about the impact of a small change in the training set that is easier to consider in a metric learning context.

In this section we have presented a vanishing generalization bound showing that a good source metric is beneficial and can significantly increase the rate of convergence of the bound. Furthermore when the metric is a perfect fit the bound shows that learning is not necessary. In the next section we propose a comparison of the bounds and a discussion on their implications.

4.6 Summary of the Bounds

In this section we propose a summary of the bounds derived in the previous sections. We recall their main characteristics in Table 4.1.

Nature of the bound In the average stability case the bound obtained is an exact one since its derivation does not rely on any concentration inequality. In the uniform stability and Rademacher complexity cases the bounds are probabilistic since they come from the application of the McDiarmid inequality (Theorem A.1).

Impact of the complexity of the metric class In the Rademacher complexity case the impact of the complexity of the metric class is in $\mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$ while this impact is improved to $\mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$ for the two stability based bounds. It can be explained by the fact that the Rademacher complexity bound measures the complexity of the metric class with respect to all the possible metrics in the class, even the ones which might never be learned by the algorithm. Hence it is a worst case result. On the other hand the stability based approaches only consider the metrics which can be learned given a training set. Hence it is closer to an exact result.

Convergence rate In the average stability case the goal is to bound the expected true risk of the algorithm over all the possible training sets while in the uniform stability and Rademacher complexity case the goal is to bound the true risk of a metric learned on any particular training set, in other words these bounds should also hold in the worst case scenario. It explains why the convergence rate is better in the average stability case.

Goodness of the source metric On the one hand in the average stability and the Rademacher complexity cases the measure of goodness of the source metric is theoretical and cannot be computed in practice making it unfit to derive an algorithm to choose a good source metric. On the other hand in the uniform stability case, the quantity involving the source metric is empirical. Following this we derived an algorithm to weight the importance of the source metric with respect to its goodness.

Applicability of the bound The three bounds require different assumptions on the metric, the loss function and the regularization. The Rademacher complexity bound is the least constrained of the three approaches as it is applicable to several kind of regularization terms and can be used with different kind of metrics. The two stability bounds are more constrained as they only hold for a Mahalanobis distance learned with a Frobenius norm regularization term.

In the next section we propose several examples of loss functions and regularization terms which can be used in one or several of our frameworks.

4.7 Examples

In this section we present several examples of popular loss functions and regularization terms in metric learning. We show that each example falls in one or more of the theoretical frameworks presented here. It demonstrates the wide range of applicability of the metric hypothesis transfer learning framework proposed here.

First of all note that in Sections 4.3 and 4.4 we only consider learning a Mahalanobis distance and this distance respects the assumption made on the metric of Section 4.5 (Equa-

tion (4.14)). Hence in this section we only consider metrics which satisfy this property. Furthermore we always consider positive and convex losses. It implies a bound on the regularization term presented in the next lemma.

Lemma 4.9 (Bounded regularization)**.** *Let $\mathbf{M}_T$ be the optimal solution returned by Problem* (4.1) *with training set T and a positive and convex loss. We have:*

$$\|\mathbf{M}_T - \mathbf{M}_{\mathcal{S}}\| \leq \sqrt{\frac{\hat{L}_T(\mathbf{M}_{\mathcal{S}})}{\lambda}}.$$

Proof. The proof of this lemma can be found in Appendix C.7. □

4.7.1 Examples of Loss Functions

Overall we successively considered the following properties for the loss function:

- Positive: On Average Stability Analysis, Uniform Stability Analysis, Rademacher Complexity Analysis
- Convex (Definition A.6): On Average Stability Analysis, Uniform Stability Analysis, Rademacher Complexity Analysis
- k-lipschitz continuous with respect to the metric (Definition 4.2): Rademacher Complexity Analysis
- k-lipschitz continuous (Definition A.1): On Average Stability Analysis, Uniform Stability Analysis
- (σ, m)-admissible (Definition A.2): Uniform Stability Analysis

First we propose to consider L-lipschitz functions for dissimilarity and similarity learning. For each example we prove that all the previous properties hold.

Example 4.1 (Positive, convex, L-lipschitz functions for dissimilarity learning)**.** *Let $f(a)$ be a positive, convex, L-lipschitz function. Given a dissimilarity (Definition 1.8) $k_{\mathbf{M}}$ parametrized by $\mathbf{M} \in \mathcal{M}$ and any two examples $\mathbf{z}, \mathbf{z}' \sim \mathcal{D}_{\mathcal{T}}$ we define a loss as:*

$$l\left(\mathbf{M}, \mathbf{z}, \mathbf{z}'\right) = f\left(\delta_{yy'}\left[k_{\mathbf{M}}\left(\mathbf{x}, \mathbf{x}'\right) - \gamma_{yy'}\right]\right) \tag{4.24}$$

where $\delta_{yy'} = 1$ if $y = y'$ and -1 otherwise and $\gamma_{yy'}$ is the desired margin between examples. This loss is:

- *Positive,*
- *Convex,*
- *k-lipschitz continuous with respect to the metric with $k = L$,*

- *k-lipschitz continuous with* $k = L \sup_{\mathbf{z},\mathbf{z}' \sim \mathcal{D}_\mathcal{T}} \|g(\mathbf{x}, \mathbf{x}')\|_*$,
- (σ, m)*-admissible with* $\begin{cases} \sigma = \sup_{\mathbf{z},\mathbf{z}' \sim \mathcal{D}_\mathcal{T}} \gamma_{yy'} \\ m = 2L \sup_{\mathbf{z},\mathbf{z}' \sim \mathcal{D}_\mathcal{T}} \|g(\mathbf{x}, \mathbf{x}')\|_* \left(\sqrt{\frac{\hat{L}_T(\mathbf{M}_\mathcal{S})}{\lambda}} + \|\mathbf{M}_\mathcal{S}\| \right) \end{cases}$.

Proof. The proof of this example can be found in Appendix C.8. □

This loss has, for example, been successfully used in Jin et al. (2009). Similarly if $f(a) = [a]_+$ then it corresponds to the loss used in Section 4.4[5]. As a last example if $f(a) = |a|$ we retrieve a loss function close to the one used in Chapter 3[6].

Example 4.2 (Positive, convex, L-lipschitz functions for similarity learning)**.** *Let* $f(a)$ *be a positive, convex, L-lipschitz function. Given a similarity (Definition 1.8)* $k_\mathbf{M}$ *parametrized by* $\mathbf{M} \in \mathcal{M}$ *and any two examples* $\mathbf{z}, \mathbf{z}' \sim \mathcal{D}_\mathcal{T}$ *we define a loss as:*

$$l\left(\mathbf{M}, \mathbf{z}, \mathbf{z}'\right) = f\left(1 - \delta_{yy'} \frac{k_\mathbf{M}(\mathbf{x}, \mathbf{x}')}{\gamma_{yy'}}\right) \tag{4.25}$$

where $\delta_{yy'} = 1$ *if* $y = y'$ *and* -1 *otherwise and* $\gamma_{yy'}$ *is the desired margin between examples. This loss is:*

- *Positive,*
- *Convex,*
- *k-lipschitz continuous with respect to the metric with* $k = \frac{L}{\inf_{\mathbf{z},\mathbf{z}' \sim \mathcal{D}_\mathcal{T}} |\gamma_{yy'}|}$,
- *k-lipschitz continuous with* $k = \frac{L}{\inf_{\mathbf{z},\mathbf{z}' \sim \mathcal{D}_\mathcal{T}} |\gamma_{yy'}|} \sup_{\mathbf{z},\mathbf{z}' \sim \mathcal{D}_\mathcal{T}} \|g(\mathbf{x}, \mathbf{x}')\|_*$,
- (σ, m)*-admissible with* $\begin{cases} \sigma = 0 \\ m = 2\frac{L}{\inf_{\mathbf{z},\mathbf{z}' \sim \mathcal{D}_\mathcal{T}} |\gamma_{yy'}|} \sup_{\mathbf{z},\mathbf{z}' \sim \mathcal{D}_\mathcal{T}} \|g(\mathbf{x}, \mathbf{x}')\|_* \left(\sqrt{\frac{\hat{L}_T(\mathbf{M}_\mathcal{S})}{\lambda}} + \|\mathbf{M}_\mathcal{S}\| \right) \end{cases}$.

Proof. The proof of this example can be found in Appendix C.9. □

This loss has, for example, been successfully used before with $f(a) = [a]_+$ in Bellet et al. (2012); Nicolae et al. (2015) albeit in a slightly different context.

We now turn our interest to H-smooth (Definition A.3), B-bounded functions as defined in Srebro et al. (2010b). Note that the results are close in spirit to the one obtained for L-lipschitz functions.

[5]The hinge loss is positive, convex and 1-lipschitz.
[6]The absolute value is positive, convex and 1-lipschitz.

Example 4.3 (Positive, convex, H-smooth, B-bounded functions for dissimilarity learning)**.** *Let $f(a)$ be a positive, convex, H-smooth, B-bounded function. Given a dissimilarity (Definition 1.8) $k_{\mathbf{M}}$ parametrized by $\mathbf{M} \in \mathcal{M}$ and any two examples $\mathbf{z}, \mathbf{z}' \sim \mathcal{D}_{\mathcal{T}}$ we define a loss as:*

$$l\left(\mathbf{M}, \mathbf{z}, \mathbf{z}'\right) = f\left(\delta_{yy'}\left[k_{\mathbf{M}}(\mathbf{x}, \mathbf{x}') - \gamma_{yy'}\right]\right) \tag{4.26}$$

where $\delta_{yy'} = 1$ if $y = y'$ and -1 otherwise and $\gamma_{yy'}$ is the desired margin between examples. This loss is:

- *Positive,*
- *Convex,*
- *k-lipschitz continuous with respect to the metric with $k = \sqrt{12HB}$,*
- *k-lipschitz continuous with $k = \sqrt{12HB}\sup_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_{\mathcal{T}}} \|g(\mathbf{x}, \mathbf{x}')\|_*$,*
- *(σ, m)-admissible with $\begin{cases}\sigma = 0 \\ m = B\end{cases}$.*

Proof. The proof of this example can be found in Appendix C.10. □

Example 4.4 (Positive, convex, H-smooth, B-bounded functions for similarity learning)**.** *Let $f(a)$ be a positive, convex, H-smooth, B-bounded function. Given a similarity (Definition 1.8) $k_{\mathbf{M}}$ parametrized by $\mathbf{M} \in \mathcal{M}$ and any two examples $\mathbf{z}, \mathbf{z}' \sim \mathcal{D}_{\mathcal{T}}$ we define a loss as:*

$$l\left(\mathbf{M}, \mathbf{z}, \mathbf{z}'\right) = f\left(1 - \delta_{yy'}\frac{k_{\mathbf{M}}(\mathbf{x}, \mathbf{x}')}{\gamma_{yy'}}\right) \tag{4.27}$$

where $\delta_{yy'} = 1$ if $y = y'$ and -1 otherwise and $\gamma_{yy'}$ is the desired margin between examples. This loss is:

- *Positive,*
- *Convex,*
- *k-lipschitz continuous with respect to the metric with $k = \frac{\sqrt{12HB}}{\inf_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_{\mathcal{T}}}|\gamma_{yy'}|}$,*
- *k-lipschitz continuous with $k = \frac{\sqrt{12HB}}{\inf_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_{\mathcal{T}}}|\gamma_{yy'}|}\sup_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_{\mathcal{T}}} \|g(\mathbf{x}, \mathbf{x}')\|_*$,*
- *(σ, m)-admissible with $\begin{cases}\sigma = 0 \\ m = B\end{cases}$.*

Proof. The proof of this example can be found in Appendix C.11. □

Smooth losses have, for example, been successfully used with $f(a) = (a)^2$ in Srebro et al. (2010a); Kuzborskij and Orabona (2014) albeit in the different context of classification.

We have presented several loss functions which can be used in our framework. In the next subsection we turn our attention toward several regularization terms.

4.7.2 Examples of Regularizations

Due to several technical issues we have to use the Frobenius norm when considering the two stability frameworks. However in the Rademacher complexity framework the only constraint is that the dual norm of the regularization should be well defined, i.e. its Rademacher average should be bounded above by a term which decreases in $\mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$. Cao et al. (2016) have shown that this condition is fulfilled by several norms that we recall in Table 4.2[7]. These norms have been successfully used before as non biased regularization terms (Jin et al., 2009; Ying et al., 2009; Bellet et al., 2012; Shen et al., 2012; Nicolae et al., 2015) but also as biased regularization terms (Parameswaran and Weinberger, 2010; Cao et al., 2013a; Bohné et al., 2014).

We now propose to discuss the impact of the regularization term on the transfer taking place between the source metric $\mathbf{M}_{\mathcal{S}}$ and the learned metric $\mathbf{M}_T$. First of all note that this differ with the kind of metric considered as the interpretation of the values of the matrix may change. Here we consider that we are learning a Mahalanobis distance or a Bilinear similarity where each entry of the matrix can be seen as a measure of the importance of the relation between two features. Hence it gives us the following possible interpretations for the different norms.

- Frobenius norm: The Frobenius norm will encourage small element-wise changes in the values of the matrix $\mathbf{M}_T$ with respect to the matrix $\mathbf{M}_{\mathcal{S}}$. It implies that the relations between the features will keep the same order of magnitude and only slightly change.

- ℓ_1-norm: The ℓ_1-norm is an element-wise sparsity inducing norm. It implies that from the source metric to the learned metric only a limited number of entries in the matrix will change. It means that some of the relations between features will be kept the same while the other relations will be able to change more than with a Frobenius norm.

- $\ell_{2,1}$-norm: The $\ell_{2,1}$-norm is a column wise sparsity inducing norm. It implies that this norm will encourage the learned metric $\mathbf{M}_T$ to keep intact whole rows (and columns if the matrix is symmetric) of $\mathbf{M}_{\mathcal{S}}$. In other words, for some features, the relationships between this feature and the others as encoded by the source metric will be kept in the learned metric.

- Trace norm: The interest of using a trace norm is mainly to obtain low rank matrices. In the case of biased regularization it translates into obtaining a low rank difference between $\mathbf{M}_T$ and $\mathbf{M}_{\mathcal{S}}$. However it does not always imply that the rank difference between the learned metric and the source metric will be small. It seems that, in a biased regularization case, the trace norm is less interesting than the other norms.

[7] Note that Cao et al. (2016) also proved tighter results (with respect to the constants) than the one presented here but this is beyond the scope of this analysis.

Table 4.2: Examples of regularization terms.

Norm $\|\cdot\|$	Dual Norm $\|\cdot\|_*$	Rademacher Average $R_n(\|\cdot\|_*)$ (See Appendix C.12 for a proof)
$\|\cdot\|_{\mathcal{F}}$	$\|\cdot\|_{\mathcal{F}}$	$R_n(\|\cdot\|_*) \leq \frac{2\sup_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_{\mathcal{T}}}\|g(\mathbf{x},\mathbf{x}')\|_{\mathcal{F}}^2}{\sqrt{n}}$
$\|\cdot\|_1$	$\|\cdot\|_\infty$	
$\|\cdot\|_{2,1}$	$\|\cdot\|_{2,\infty}$	
$\|\cdot\|_{\mathrm{Tr}}$	$\|\cdot\|_{\mathrm{Spec}}$	

In this section we have presented several examples of loss functions and regularization terms which can be used in our framework. In the next section we propose to empirically study it in several experiments. On the one hand we propose to evaluate the interest of optimizing the goodness of the metric as defined in Section 4.4. On the other hand we demonstrate that using our approach in a semi-supervised domain adaptation task leads to state of the art results.

4.8 Experiments

In Section 4.4, instantiating Problem (4.1) with the hinge loss (Example 4.1) and the Frobenius norm, we have derived an empirical measure of the goodness of a metric. It leads to the development of an optimization problem to learn the weight of the source metric (Equation (4.13)). In this section we consider two empirical studies depending on the choice of the source metric. First, using some well-known distances as a source metric, we show that our framework performs well on classic supervised metric learning tasks of the UCI database and we empirically demonstrate the interest of learning the ω parameter. Second, we apply our framework with weighted source metric in a semi-supervised Domain Adaptation task. We show that, using only source information through a learned metric, our method is able to compete with state of the art algorithms which have access to the examples of the source dataset.

Setup In all our experiments we use limited training dataset, making it difficult to apply any kind of cross-validation to set the parameters. Thus we propose to fix them as follows. First the positive and negative margin are respectively set to the 5^{th} and 95^{th} percentile of the training set possible distances computed with the source metric as proposed in Davis et al. (2007). Next we set λ such that the two terms of Equation (4.13) are equals, i.e. we balance the complexity and performance importance with respect to the source metric. The ω parameter is then learned using Problem (4.13). In all the experiments we plug our metric in a 1-nearest neighbour classifier to classify the examples of the test set. Furthermore, the significance of the results is assessed with a paired samples t-test considering that an approach is significantly better when the p-value is lower than 0.05.

Table 4.3: Results of the experiments conducted on the UCI datasets. Each value corresponds to the mean and standard deviation over 10 runs. For each dataset we highlight the best result using a bold font. Approaches with the suffix -ω_1 do not learn ω but fix it to 1.

	Baselines			Our approach			
Dataset	1-NN	ITML	LMNN	IDENTITY	IDENTITY-ω_1	MAHALANOBIS	MAHALANOBIS-ω_1
Breast	95.31 ± 1.11	95.40 ± 1.37	95.60 ± 0.92	**96.06 ± 0.77**	95.75 ± 0.87	95.71 ± 0.84	94.76 ± 1.38
Pima	67.92 ± 1.95	68.13 ± 1.86	67.90 ± 2.05	67.87 ± 1.57	67.54 ± 1.99	**68.37 ± 2.00**	66.31 ± 2.37
Scale	78.73 ± 1.69	**87.31 ± 2.35**	86.20 ± 2.83	80.98 ± 1.51	80.82 ± 1.27	81.35 ± 1.17	80.88 ± 1.43
Wine	93.40 ± 2.70	93.82 ± 2.63	93.47 ± 1.80	**95.42 ± 1.71**	95.07 ± 1.68	94.31 ± 2.01	80.56 ± 5.75

4.8.1 Classic Supervised Metric Learning

First we start by conducting experiments on several UCI datasets Lichman (2013), namely breast, pima, scale and wine. We propose to consider three source metrics:

- **Zero**: No source hypothesis,
- **Identity**: Euclidean distance,
- **Mahalanobis**: Inverse of the variance-covariance matrix computed on the training set (Mahalanobis, 1936).

For the last two source metric we propose two experiments, one where we set $\omega = 1$ and one where we learn ω using Problem (4.13). The goal of this experiment is to show the interest of automatically setting ω. We consider a 1-nearest neighbour (1-NN) classifier using the Euclidean Distance as the baseline and also report the results of two well known metric learning algorithms, namely ITML (Davis et al., 2007) and LMNN (Weinberger et al., 2005). The results averaged over 10 runs are reported in Table 4.3. For each run we randomly draw a training set containing 20% of the data available for each class and we test the metric on the remaining 80% of data.

These experiments highlight the interest of learning the ω parameter. When we consider the performance of our approach with and without learning ω, we mainly notice the following facts. First, learning ω always leads to an improvement on all the datasets and the final result is better than the 1-NN classifier. Second, learning ω when considering the identity matrix as the source metric seems to be of limited interest as the differences in accuracy are only significant for the wine dataset. This can be justified by the fact that, in this case, it only consists of a rescaling of the diagonal of the matrix and it does not change much the behaviour of the distance. Finally, learning ω when considering the variance-covariance matrix as the source metric leads to a significant improvement of the performance of the metric except on the breast dataset. This is particularly true for the wine dataset with a gain of nearly 14% in accuracy. It can be explained by the fact that, for this dataset, we are learning with less than 40 examples. Thus the original Mahalanobis distance does not carry as much information as in the other datasets and is thus of a lower quality. Learning ω allows us to compensate this drawback and to obtain results which are even better than ITML or LMNN.

4.8.2 Metric Learning for Semi-Supervised Domain Adaptation

In this section we consider a semi-supervised domain adaptation task with the Office-Caltech dataset. This dataset consists of four domains: Amazon (A), Caltech (C), DSLR (D) and Webcam (W) for which 10 classes are considered. This leads to consider 12 different adaptation problems when we alternatively take each domain as the source or the target dataset. The results are averaged over 20 runs. In each run the training set is composed of 8 labelled source examples (20 if the source is Amazon) and 3 labelled target examples for each class. The testing set corresponds to the remaining target examples. In these experiments we use the same splits as the ones considered in Hoffman et al. (2013) since they are freely available from the authors website and we follow their experimental setup. The data is normalized thanks to the zscore and the dimensionality of the examples is reduced to 20 thanks to a simple PCA. The results are presented in Table 4.4[8] where we compare the performance of our method to 6 baselines:

- 1-NN$_{\mathcal{S}}$: a 1-NN using the source examples,
- 1-NN$_{\mathcal{T}}$: a 1-NN using the target examples,
- LMNN$_{\mathcal{T}}$: a 1-NN on the target examples using the metric learned by LMNN on the source examples,
- ITML$_{\mathcal{T}}$: a 1-NN on the target examples using the metric learned by ITML on the source examples,
- MMDT: a domain adaptation method Hoffman et al. (2013),
- GFK: another DA approach Gong et al. (2012).

The last two methods need the source sample while in our case we only use a source metric learned from the source instances. For our biased regularization framework we consider 3 possible metrics learned on the source examples, namely Mahalanobis, ITML and LMNN. These source metrics are weighted by ω_T which is learned using Problem (4.13).

These results show that metric hypothesis transfer learning can perform well in a semi-supervised domain adaptation setting. Indeed, we perform better than directly plugging the metrics learned by LMNN and ITML in a 1-NN classifier. Moreover, we obtain accuracies which are competitive with state of the art approaches like MMDT or GFK while using less information. If we compare our approach using LMNN as the source metric with MMDT, we note that MMDT is significantly better than our approach on 4 out of 12 tasks while we are significantly better on 3 and 5 end as a draw. Hence we can conclude that our method presents a similar level of performance than MMDT. Similarly, if we compare our approach using LMNN as the source metric with GFK, we obtain that GFK is significantly better than

[8]Note that we also report the mean accuracy over the 12 tasks. Even if we are conscious that the problems are different, it gives a rough idea of the global performance of the compared approaches.

Table 4.4: Metric Learning for Semi-Supervised Domain Adaptation. For the sake of readability we design the considered domains by their initials. $\mathcal{S} \to \mathcal{T}$ stands for adaptation from the source domain to the target domain. Each time we consider the mean and standard deviation over 20 runs. For each task, the best result is highlighted with a bold font.

	Baselines						Our approach		
Task	1-NN$_\mathcal{S}$	1-NN$_\mathcal{T}$	LMNN$_\mathcal{T}$	ITML$_\mathcal{T}$	MMDT	GFK	MAHALANOBIS	ITML	LMNN
A → C	35.95 ± 1.30	31.92 ± 3.24	32.42 ± 3.03	32.56 ± 4.17	**39.76 ± 2.25**	37.81 ± 1.85	32.65 ± 3.76	32.93 ± 4.60	34.66 ± 3.66
A → D	33.58 ± 4.37	53.31 ± 4.31	49.96 ± 3.53	44.33 ± 8.18	54.25 ± 4.32	51.54 ± 3.55	54.69 ± 3.96	51.54 ± 4.03	**54.72 ± 5.00**
A → W	33.68 ± 3.60	66.25 ± 3.87	62.62 ± 4.49	58.17 ± 10.63	64.91 ± 5.71	59.36 ± 4.30	67.11 ± 5.11	64.09 ± 5.20	**67.62 ± 5.18**
C → A	37.37 ± 2.95	47.28 ± 4.15	42.97 ± 3.76	45.16 ± 7.60	**51.05 ± 3.38**	46.36 ± 2.94	50.15 ± 4.87	49.89 ± 5.25	50.36 ± 4.67
C → D	31.89 ± 5.77	54.17 ± 4.76	46.02 ± 6.54	48.07 ± 8.98	52.80 ± 4.84	**58.07 ± 3.90**	56.77 ± 4.63	53.78 ± 7.23	57.44 ± 4.48
C → W	28.60 ± 6.13	65.06 ± 6.27	55.79 ± 5.09	59.21 ± 9.71	62.75 ± 5.19	63.26 ± 5.89	64.64 ± 6.44	64.00 ± 6.08	**65.11 ± 5.25**
D → A	33.59 ± 1.77	47.81 ± 3.56	40.57 ± 3.79	45.06 ± 6.78	**50.39 ± 3.40**	40.77 ± 2.55	49.48 ± 4.41	49.11 ± 4.09	49.67 ± 4.00
D → C	31.16 ± 1.19	32.22 ± 2.98	27.96 ± 3.03	29.93 ± 4.84	**35.70 ± 3.25**	30.64 ± 1.98	32.90 ± 3.14	32.99 ± 3.58	33.84 ± 2.99
D → W	**76.92 ± 2.18**	66.19 ± 4.60	65.36 ± 3.82	66.74 ± 7.16	74.43 ± 3.10	74.98 ± 2.89	65.57 ± 4.52	66.38 ± 6.04	69.72 ± 3.78
W → A	32.19 ± 3.04	48.25 ± 3.52	41.69 ± 3.71	45.11 ± 5.72	50.56 ± 3.66	43.26 ± 2.34	50.80 ± 3.63	50.16 ± 4.32	**50.92 ± 4.00**
W → C	27.67 ± 2.58	30.74 ± 3.92	28.60 ± 3.41	28.99 ± 4.31	**34.86 ± 3.62**	29.95 ± 3.05	31.54 ± 3.60	31.40 ± 4.29	32.64 ± 3.52
W → D	64.61 ± 4.30	54.84 ± 5.17	56.89 ± 5.06	57.76 ± 7.03	62.52 ± 4.40	**71.93 ± 4.07**	57.17 ± 6.50	56.85 ± 5.51	61.14 ± 5.78
Mean	38.93 ± 3.26	49.84 ± 4.20	45.90 ± 4.11	46.76 ± 7.09	**52.83 ± 3.93**	50.66 ± 3.28	51.12 ± 4.55	50.26 ± 5.02	52.32 ± 4.36

our approach on 3 tasks, we are significantly better on 7 and 2 lead to a draw. Hence, we can conclude that our approach performs better than GFK.

If we compare the performances of both ITML and LMNN as metrics used directly in a nearest neighbour classifier one can intuitively expect ITML to be a better source hypothesis than LMNN since its results are better. However, in practice, using the metric learned by LMNN as the source hypothesis yields better results. This suggests that using a learned source model that tends to over-fit reasonably the learning source sample can be of potential interest in a transfer learning context. Indeed LMNN does not use a regularization term in its formulation is thus prone to over-fitting. Since the parameter ω penalizes the source metric with respect to its complexity it may limit the impact of the source metric to what is needed for the transfer. Nevertheless, this point deserves further investigation.

4.9 Conclusion

In this chapter we formalised and theoretically analysed the metric hypothesis transfer learning framework. This framework takes into account a source hypothesis information to help learning by means of a biased regularization. This regularization can be interpreted into two ways: (i) when the source metric is an a priori known metric such as the identity matrix, the objective is to infer a new metric that performs better than the source metric, (ii) when the source metric has been learned from another domain, the formulation allows one to transfer the knowledge from the source metric to the new domain. This last interpretation refers to a transfer learning setting where the learner does not have access to source examples and can only make use of the source model in the presence of few labelled data.

In our theoretical analysis we considered three different frameworks. First the on average stability framework allowed us to derive an exact bound showing that with a convergence rate in $\mathcal{O}\left(\frac{1}{n}\right)$ the learned metric will, on average over all the size n training sets, be as good as the

source metric. Second the uniform stability framework leads to a probabilistic generalization bound where, given a specific loss, an empirical measure of the goodness of the source metric can be obtained. From this we proposed an algorithm to optimally weight the source metric in order to optimize the bound in a theoretically sound way. Third we used the Rademacher complexity framework to address both problems of considering different regularization terms and obtaining a vanishing bound, i.e. a bound which implies that learning is no longer necessary when the source metric is the perfect fit.

To further demonstrate the interest of metric hypothesis transfer learning we proposed several examples of loss functions and regularization terms which can be used with our theoretical analysis. We also discussed the impact of the different regularization terms on the transfer of informations between the learned metric and the source metric. Lastly we proposed an empirical evaluation of our source metric weighting method. On the one hand we considered a classic metric learning task where we showed that weighting the source metric to minimize the bound is indeed beneficial. On the other hand, in a semi-supervised domain adaptation task we demonstrated the good behaviour of the metric hypothesis transfer learning framework. Indeed we obtained results comparable to state of the art approaches which fully make use of the source examples while we only have access to the source metric.

As stated in Kuzborskij and Orabona (2014) in another context, our results stress the importance of choosing a good source hypothesis. Perspectives of this work include further empirical investigations on the interest of using metric hypothesis transfer learning. In particular empirically studying the impact of the regularization on the transferred informations between the source and learned metric could be of interest. Here we considered the case where we have a single source metric. One interesting perspective could be to consider more complex strategies to learn ω and, for example, to study the multi-source case where several source metrics are available at once. Note that one of the main limitations of this work is the fact that we did not manage to obtain a vanishing bound highlighting an empirical goodness criterion for the source metric. Indeed in Section 4.4 we obtained an empirical criterion but the bound is not a vanishing one while in Section 4.5 we obtained a vanishing bound but the associated goodness criterion is theoretical in the sense that it cannot be computed. Hence a perspective could be to develop a vanishing bound with empirical goodness. It would imply obtaining a theoretically sound way to choose the source metric. Furthermore it would probably give further insights on the trade-off between goodness of the source metric and number of examples.

In the first part of this thesis we have been interested in learning in the presence of an auxiliary metric serving as a reference during the learning process. On the one hand in Chapter 3 we studied the case where this metric is only available through its values and we want to learn a good approximation. On the other hand in Chapter 4 we considered that this metric is accessible through its parameters matrix and that it can help us learning a better metric for a given task. In other words in this first part we constrained the learned metric to have a behaviour similar to the reference metric. Hence we implicitly, through the reference

metric, controlled the behaviour of the learned metric. In the next part of this thesis we propose to address the problem of explicitly controlling the behaviour of a metric. To this end in Chapter 5 we propose a new metric learning framework based on virtual points. In other words instead of learning a metric based on similarity constraints between examples we propose to explicitly control the target position of the examples by bringing them closer to these virtual points.

Part III

Metric Learning with Controlled Behaviour

Chapter 5

Regressive Virtual Metric Learning

This chapter is based on the following publication

Michaël Perrot and Amaury Habrard. Regressive virtual metric learning. In *Advances in Neural Information Processing Systems (NIPS-15)*, pages 1810–1818, 2015c

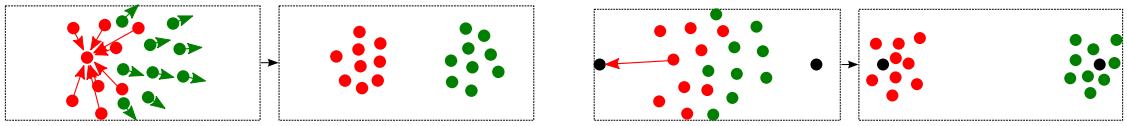
---

Abstract

In this chapter we are interested in supervised metric learning of Mahalanobis like distances. Existing approaches mainly focus on learning a new distance using similarity and dissimilarity constraints between examples. Here instead of bringing closer examples of the same class and pushing far away examples of different classes we propose to move the examples with respect to virtual points. Hence, each example is brought closer to an a priori defined virtual point reducing the number of constraints to satisfy and explicitly controlling the expected behaviour of the metric for each example. We show that our problem admits a closed form solution which can be kernelized. We provide a theoretical analysis proving the generalization ability of the metric learned with our approach and establishing some links with other classic metric learning methods. Furthermore we propose an efficient solution to the difficult problem of selecting virtual points based in part on recent works in optimal transport. Lastly, we evaluate our approach on several state of the art datasets.

5.1 Introduction

Most of the existing approaches in metric learning use constraints of type must-link and cannot-link between learning examples (See Section 2.3). For example, in a supervised classification task, the goal is to bring closer examples of the same class and to push far away examples of different classes. The idea is that the learned metric should affect a high value to dissimilar examples and a low value to similar examples. Then, this new distance can be used in a classification algorithm like a nearest neighbour classifier. Note that in this case the



(a) classic must-link cannot-link approach. (b) Our virtual point-based regression formulation.

Figure 5.1: Arrows denote the constraints used by each approach for one particular example in a binary classification task. The classic metric learning approach in Figure 5.1(a) uses $\mathcal{O}(n^2)$ constraints bringing closer examples of the same class and pushing far away examples of different classes. On the contrary, our approach presented in Figure 5.1(b) moves the examples to the neighborhood of their corresponding virtual point, in black, using only $\mathcal{O}(n)$ constraints.

set of constraints is quadratic in the number of examples which can be prohibitive when the number of examples increases. One heuristic is then to select only a subset of the constraints but selecting such a subset is not trivial.

In this chapter, we propose to consider a new kind of constraints where each example is associated with an a priori defined virtual point. It allows us to consider the metric learning problem as a simple regression where we try to minimize the differences between learning examples and virtual points. Figure 5.1 illustrates the differences between our approach and a classic metric learning approach. It can be noticed that our algorithm only uses a linear number of constraints. However defining these constraints by hand can be tedious and difficult. To overcome this problem, we present two approaches to automatically define them. The first one is based on some recent advances in the field of optimal transport while the second one uses a class-based representation space.

Moreover, thanks to its regression-based formulation, our approach can be easily kernelized allowing us to deal efficiently with non linear transformations which is a nice advantage in comparison to some metric learning methods. We also provide a theoretical analysis showing the generalization ability of the metrics learned with our approach and establishing some relationships with a classic metric learning formulation.

This chapter is organised as follows. In Section 5.2 we present our framework to learn a metric when the virtual points are known. Then in Section 5.3 we address the problem of selecting these virtual points. In Section 5.4 we theoretically analyse our approach by deriving a generalization bound and by showing some links with a classic metric learning approach. In Section 5.5 we empirically demonstrate the interest of our approach. We conclude in Section 5.6.

5.2 Learning a Metric Using Virtual Points

The main idea behind our algorithm is to bring closer the learning examples to a set of virtual points. Here we assume that we have access to a set of n learning pairs $(\mathbf{x},\mathbf{v})$ where $\mathbf{x}$ is a

learning example and $\mathbf{v}$ is a virtual point associated to $\mathbf{x}$. We present both the linear and kernelized formulations of our approach called Regressive Virtual Metric Learning (RVML). It boils down to solve a regression in closed form, the main originality being the introduction of virtual points.

Given a probability distribution $\mathcal{D}_\mathcal{T}$ defined over $\mathcal{X} \times \mathcal{Y}$ where $\mathcal{X} \subseteq \mathbb{R}^d$ and $\mathcal{Y}$ is a finite label set, let $T = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ be a set of examples drawn i.i.d. from $\mathcal{D}_\mathcal{T}$. Let $f_\mathcal{V} : \mathcal{X} \times \mathcal{Y} \to \mathcal{V}$, where $\mathcal{V} \subseteq \mathbb{R}^{d'}$ is the space of virtual points, be the function which associates each example to a virtual point. We consider the learning set $V = \{(\mathbf{x}_i, \mathbf{v}_i)\}_{i=1}^n$ where $\mathbf{v}_i = f_\mathcal{V}(\mathbf{x}_i, y_i)$. We denote by $\mathcal{D}_\mathcal{V}$ the probability distribution defined on $\mathcal{X} \times \mathcal{V}$ obtained from the distribution $\mathcal{D}_\mathcal{T}$ after applying $f_\mathcal{V}$, i.e. $\Pr_{\mathcal{D}_\mathcal{V}}(\mathbf{x}, \mathbf{v}) = \Pr_{\mathcal{D}_\mathcal{T}}(\mathbf{x}, y | \mathbf{v} = f_\mathcal{V}(\mathbf{x}, y))$. Thus it is equivalent to obtain the set of examples $V = \{(\mathbf{x}_i, \mathbf{v}_i)\}_{i=1}^n$ from T after applying $f_\mathcal{V}$ and to draw V i.i.d. from $\mathcal{D}_\mathcal{V}$. For the sake of simplicity denote by $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^T$ and $\mathbf{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_n)^T$ the matrices containing respectively one example and its associated virtual point on each line.

We consider that the function $f_\mathcal{V}$ is known. We come back to its definition in Section 5.3. Our goal is to learn a Mahalanobis distance through its linear transformation interpretation (Section 1.4). More precisely we want to learn the linear transformation matrix $\mathbf{L}$ such that $\mathbf{M} = \mathbf{L}\mathbf{L}^T$. We consider the following optimisation problem:

$$\underset{\mathbf{L} \in \mathbb{R}^{d \times d'}}{\arg\min}\, \hat{L}_V(\mathbf{L}) + \lambda \left\|\mathbf{L}\right\|_\mathcal{F}^2 \tag{5.1}$$

where $\hat{L}_V(\mathbf{L})$ is the empirical risk defined as follows:

$$\hat{L}_V(\mathbf{L}) = \frac{1}{n} \sum_{(\mathbf{x},\mathbf{v}) \in V} l\left(\mathbf{L}, (\mathbf{x}, \mathbf{v})\right)$$

with $l\left(\mathbf{L}, (\mathbf{x}, \mathbf{v})\right) = \left\|\mathbf{x}^T\mathbf{L} - \mathbf{v}^T\right\|_2^2$. Note that we can also write the empirical risk in matrix form:

$$\hat{L}_V(\mathbf{L}) = \frac{1}{n} \left\|\mathbf{X}\mathbf{L} - \mathbf{V}\right\|_\mathcal{F}^2 . \tag{5.2}$$

We also define the true risk of a matrix $\mathbf{L}$ as:

$$L_\mathcal{V}(\mathbf{L}) = \mathop{\mathbb{E}}_{(\mathbf{x},\mathbf{v}) \sim \mathcal{D}_\mathcal{V}} l\left(\mathbf{L}, (\mathbf{x}, \mathbf{v})\right). \tag{5.3}$$

The idea is to learn a new space of representation where each example is close to its associated virtual point. Note that $\mathbf{L}$ is a $d \times d'$ matrix and if $d' < d$ we also perform dimensionality reduction.

Theorem 5.1 (Optimal solution of Problem (5.1))**.** *The optimal solution of Problem* (5.1) *can be found in closed form. Furthermore, we can derive two equivalent solutions:*

$$\mathbf{L}_V = \left(\mathbf{X}^T\mathbf{X} + \lambda n \mathbf{I}\right)^{-1} \mathbf{X}^T\mathbf{V} \tag{5.4}$$

$$\Leftrightarrow \qquad \mathbf{L}_V = \mathbf{X}^T \left(\mathbf{X}\mathbf{X}^T + \lambda n \mathbf{I}\right)^{-1} \mathbf{V}. \tag{5.5}$$

Proof. The proof of this theorem can be found in Appendix D.1. □

From Equation (5.4) we deduce the matrix $\mathbf{M}_V$:

$$\mathbf{M}_V = \mathbf{L}_V \mathbf{L}_V^T = \left(\mathbf{X}^T\mathbf{X} + \lambda n \mathbf{I}\right)^{-1} \mathbf{X}^T \mathbf{V}\mathbf{V}^T \mathbf{X} \left(\mathbf{X}^T\mathbf{X} + \lambda n \mathbf{I}\right)^{-1}. \tag{5.6}$$

Note that $\mathbf{M}_V$ is PSD by construction:

$$\mathbf{x}^T \mathbf{M}_V \mathbf{x} = \mathbf{x}^T \mathbf{L}_V \mathbf{L}_V^T \mathbf{x} = \left\|\mathbf{L}_V^T \mathbf{x}\right\|_2^2 \geq 0.$$

So far, we have focused on the linear setting. We now present a kernelized version, showing that it is possible to learn a metric in a very high dimensional space without an explicit projection.

Let $\phi(\mathbf{x})$ be a projection function and $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x}')$ be its associated kernel (Section 1.4). For the sake of readability, let $\mathbf{K}_\mathbf{X} = \mathbf{\Phi}_\mathbf{X} \mathbf{\Phi}_\mathbf{X}^T$ where $\mathbf{\Phi}_\mathbf{X} = \left(\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_n)\right)^T$. We also define V_k the kernelized version of V. Given the solution matrix $\mathbf{L}_V$ presented in Equation (5.5), we have:

$$\mathbf{M}_V = \mathbf{X}^T \left(\mathbf{X}\mathbf{X}^T + \lambda n \mathbf{I}\right)^{-1} \mathbf{V}\mathbf{V}^T \left(\mathbf{X}\mathbf{X}^T + \lambda n \mathbf{I}\right)^{-1} \mathbf{X}.$$

Then, $\mathbf{M}_{V_k}$ the kernelized version of the matrix $\mathbf{M}_V$ is defined such that:

$$\mathbf{M}_{V_k} = \mathbf{\Phi}_\mathbf{X}^T \left(\mathbf{K}_\mathbf{X} + \lambda n \mathbf{I}\right)^{-1} \mathbf{V}\mathbf{V}^T \left(\mathbf{K}_\mathbf{X} + \lambda n \mathbf{I}\right)^{-1} \mathbf{\Phi}_\mathbf{X}.$$

The squared Mahalanobis distance can be written as:

$$d^2_\mathbf{M}\left(\mathbf{x}, \mathbf{x}'\right) = \mathbf{x}^T \mathbf{M} \mathbf{x} + \mathbf{x}'^T \mathbf{M} \mathbf{x}' - 2\mathbf{x}^T \mathbf{M} \mathbf{x}'. \tag{5.7}$$

Thus we can obtain:

$$d^2_{\mathbf{M}_{V_k}}\left(\phi(\mathbf{x}), \phi\left(\mathbf{x}'\right)\right) = \phi(\mathbf{x})^T \mathbf{M}_{V_k} \phi(\mathbf{x}) + \phi\left(\mathbf{x}'\right)^T \mathbf{M}_{V_k} \phi\left(\mathbf{x}'\right) - 2\phi(\mathbf{x})^T \mathbf{M}_{V_k} \phi\left(\mathbf{x}'\right) \tag{5.8}$$

the kernelized version by considering that:

$$\begin{aligned} \phi(\mathbf{x})^T \mathbf{M}_{V_k} \phi(\mathbf{x}) &= \phi(\mathbf{x})^T \mathbf{\Phi}_\mathbf{X}^T \left(\mathbf{K}_\mathbf{X} + \lambda n \mathbf{I}\right)^{-1} \mathbf{V}\mathbf{V}^T \left(\mathbf{K}_\mathbf{X} + \lambda n \mathbf{I}\right)^{-1} \mathbf{\Phi}_\mathbf{X} \phi(\mathbf{x}) && (5.9) \\ &= \mathbf{k}_{\mathbf{X},\mathbf{x}}^T \left(\mathbf{K}_\mathbf{X} + \lambda n \mathbf{I}\right)^{-1} \mathbf{V}\mathbf{V}^T \left(\mathbf{K}_\mathbf{X} + \lambda n \mathbf{I}\right)^{-1} \mathbf{k}_{\mathbf{X},\mathbf{x}} && (5.10) \end{aligned}$$

where $\mathbf{k}_{\mathbf{X},\mathbf{x}} = \left(k(\mathbf{x}, \mathbf{x}_1), \ldots, k(\mathbf{x}, \mathbf{x}_n)\right)^T$ is the similarity vector to the examples with respect to k.

Note that it is also possible to obtain a kernelized version of $\mathbf{L}_V$:

$$\mathbf{L}_{V_k} = \mathbf{\Phi}_\mathbf{X}^T \left(\mathbf{K}_\mathbf{X} + \lambda n \mathbf{I}\right)^{-1} \mathbf{V}.$$

This result is close to a previous one already derived in Cortes et al. (2005) in a structured output setting. The main difference is the fact that we do not use a kernel on the output

(the virtual points here). Hence, it is possible to compute the projection of an example $\mathbf{x}$ of dimension d in a new space of dimension d':

$$\begin{aligned}\phi(\mathbf{x})\,\mathbf{L}_{V_k} &= \phi(\mathbf{x})^T\,\mathbf{\Phi}_{\mathbf{X}}^T\,(\mathbf{K}_{\mathbf{X}} + \lambda n\mathbf{I})^{-1}\,\mathbf{V} \\ &= \mathbf{k}_{\mathbf{X},\mathbf{x}}^T\,(\mathbf{K}_{\mathbf{X}} + \lambda n\mathbf{I})^{-1}\,\mathbf{V}.\end{aligned}$$

where $\mathbf{k}_{\mathbf{X},\mathbf{x}} = (k(\mathbf{x},\mathbf{x}_1),\ldots,k(\mathbf{x},\mathbf{x}_n))^T$ is the similarity vector to the examples with respect to k. Recall that we are interested in learning a distance between examples and not in the prediction of the virtual points which only serve as a way to bring closer similar examples and push far away dissimilar examples.

In this section we presented our linear and kernelized metric learning approaches when the virtual points are given. If these points could be chosen by hand, we believe that an automatic solution would be preferable. We propose to address this problem in the next section.

5.3 Choosing the Virtual Points

Previously, we assumed to have access to the function $f_{\mathcal{V}} : \mathcal{X} \times \mathcal{Y} \to \mathcal{V}$. In this subsection, we present two methods for generating automatically the set of virtual points and the mapping $f_{\mathcal{V}}$.

5.3.1 Using Optimal Transport on the Learning Set

In this first approach we propose to generate the virtual points associated to the examples of a training set T as follows. First we use a variation of the landmark selection method proposed in Kar and Jain (2011) to choose in T a set T' of n' landmarks. Then we use a recent variation of the optimal transport problem proposed by Courty et al. (2014b) to associate each example $\mathbf{z}_i \in T$ to the landmarks $\mathbf{z}'_1,\ldots,\mathbf{z}'_{n'} \in T'$ with weights $\mathbf{W}(i,1),\ldots,\mathbf{W}(i,n')$ such that:

$$\sum_j \mathbf{W}(i,j) = 1. \tag{5.11}$$

The virtual points are then defined as a weighted combination of the landmarks. Let $\mathbf{X}'$ be the matrix form of T', for an example $\mathbf{z}_i \in T$ we define $f_{\mathcal{V}}$ as:

$$f_{\mathcal{V}}(\mathbf{x}_i, y_i) = \mathbf{W}(i,)\mathbf{X}'. \tag{5.12}$$

In the following we present our landmark selection method based on the work of Kar and Jain (2011) and the variation of the optimal transport problem (Courty et al., 2014b) that we consider.

Landmark Selection

To select the set T' we propose an adaptation of the selection method of Kar and Jain (2011) allowing us to take into account some diversity among the landmarks. Our approach is a

fully automatic procedure and is summarized in Algorithm 2 and works as follows. First we assume without loss of generality that the examples are centred in $\mathbf{0}$ and we select as the first landmark the example $\mathbf{z} \in T$ furthest from $\mathbf{0}$. Each new landmark is selected as the example $\mathbf{z} \in T$ with the largest minimum distance with the landmarks in T'. To avoid explicitly choosing the number of landmarks we propose to stop the selection process under two conditions:

- the number of landmarks is greater than the number of classes,
- the maximum distance between an example and a landmark is lower than the mean of pairwise distances between all the examples of T.

input : $T = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ a set of examples; $\mathcal{Y}$ the label set.
output: T' a subset of T
begin
  μ = mean of distances between all the examples of T
  $\mathbf{x}_{\max} = \arg\max_{\mathbf{x} \in T} \|\mathbf{x} - \mathbf{0}\|_2$
  $T' = \{\mathbf{x}_{\max}\}$
  $T = T \setminus T'$
  $\varepsilon = \max_{\mathbf{x} \in T} \min_{\mathbf{x}' \in T'} \left\|\mathbf{x} - \mathbf{x}'\right\|_2$
  while $|T'| < |\mathcal{Y}|$ **or** $\varepsilon > \mu$ **do**
    $\mathbf{x}_{\max} = \arg\max_{\mathbf{x} \in T} \sum_{\mathbf{x}' \in T'} \left\|\mathbf{x} - \mathbf{x}'\right\|_2$
    $T' = T' \cup \{\mathbf{x}_{\max}\}$
    $T = T \setminus T'$
    $\varepsilon = \max_{\mathbf{x} \in T} \min_{\mathbf{x}' \in T'} \left\|\mathbf{x} - \mathbf{x}'\right\|_2$
  end
end

Algorithm 2: Selecting T' from a set of examples T.

Optimal Transport

Assume that you have an input distribution and an output distribution, the goal of optimal transport (Villani, 2008) is to align the two distributions at a minimal cost. We come back to this general problem in Section 6.2 and instead we consider a particular case of discrete optimal transport where the idea is to transport the examples of an input set T toward the examples of an output set T' at a minimal cost. Here we start by presenting the solution to this problem proposed by Courty et al. (2014b) before explaining how we use it to associate the training examples to the landmarks.

In the particular case of discrete optimal transport we assume that each example in T has a mass of $\frac{1}{n}$ where n is the number of examples in T. Similarly each example in T' has

a mass of $\frac{1}{n'}$. We also consider that we have a cost matrix $\mathbf{C} \in \mathbb{R}_+^{n \times n'}$ where each entry $\mathbf{C}(i,j)$ represents the cost of moving example $\mathbf{x}_i \in T$ toward example $\mathbf{x}'_j \in T'$. For example we can set $\mathbf{C}(i,j) = \left\|\mathbf{x}_i - \mathbf{x}'_j\right\|_2$. The goal is then to learn a weight matrix $\mathbf{\Gamma} \in \mathbb{R}_+^{n \times n'}$ which minimizes the transport cost $\langle \mathbf{\Gamma}, \mathbf{C} \rangle_{\mathcal{F}}$ between the two sets. Note that this matrix should take into account the mass associated with each example, i.e. it has to respect the constraints:

$$\begin{aligned} \forall \mathbf{x}_i \in T, \mathbf{\Gamma}(i,)\mathbf{1}_{n'} = \frac{1}{n}, \\ \forall \mathbf{x}'_j \in T', \mathbf{\Gamma}(,j)\mathbf{1}_n = \frac{1}{n'} \end{aligned} \tag{5.13}$$

where $\mathbf{1}_{n'}$ and $\mathbf{1}_n$ are vectors of 1 of size n' and n respectively. To learn the matrix $\mathbf{\Gamma}$ Courty et al. (2014b) propose to use the following regularized optimization problem:

$$\begin{aligned} \underset{\mathbf{\Gamma} \in \mathbb{R}_+^{n \times n'}}{\arg\min} \ \langle \mathbf{\Gamma}, \mathbf{C} \rangle_{\mathcal{F}} - \frac{1}{\lambda} h(\mathbf{\Gamma}) + \eta \sum_j \sum_c \|\mathbf{\Gamma}(y_i = c, j)\|_q^p \\ \text{s.t. } \forall \mathbf{x}_i \in T, \mathbf{\Gamma}(i,)\mathbf{1}_{n'} = \frac{1}{n}, \\ \forall \mathbf{x}'_j \in T', \mathbf{\Gamma}(,j)\mathbf{1}_n = \frac{1}{n'}. \end{aligned}$$

The two regularization terms have different objectives.

- $-\frac{1}{\lambda} h(\mathbf{\Gamma})$ where $h(\mathbf{\Gamma}) = -\sum_{i,j} \mathbf{\Gamma}(i,j) \log(\mathbf{\Gamma}(i,j))$ is the entropy of gamma: this regularization term has been proposed by Cuturi (2013b). It allows one to solve the transportation problem more efficiently by using the Sinkhorn-Knopp algorithm (Knight, 2008). Furthermore by setting the value of the parameter λ it is possible to control the sparsity of the matrix $\mathbf{\Gamma}$. On the one hand if the matrix is sparse it implies that each example from T will be associated to a small number of examples in T'. On the other hand if the matrix is full it implies that each example from T will be associated to each example in T'.

- $\eta \sum_j \sum_c \|\mathbf{\Gamma}(y_i = c, j)\|_q^p$ where $\mathbf{\Gamma}(y_i = c, j)$ corresponds to the lines of the j^{th} column of $\mathbf{\Gamma}$ where the class of the input is c: this term has been proposed by Courty et al. (2014b). Its goal is to prevent input examples of different classes to move toward the same output examples by promoting group sparsity in the matrix $\mathbf{\Gamma}$. This is done thanks to the functions $\|\cdot\|_q^p$ corresponding to a ℓ_q-norm to the power of p used here with $q = 1$ and $p = \frac{1}{2}$ (See the $\ell_{p,q}$-norm in Section 1.4).

Once the matrix $\mathbf{\Gamma}$ has been learned it is possible to compute the image $\hat{\mathbf{x}}_i$ of an input example $\mathbf{x}_i \in T$ as follows:

$$\hat{\mathbf{x}}_i = n' \mathbf{\Gamma}(i,) \mathbf{X}' \tag{5.14}$$

where $\mathbf{X}'$ is the matrix form of T' with one example per line. In this case multiplying $\mathbf{\Gamma}$ by n' ensures that $\sum_j \mathbf{\Gamma}(i,j) = 1$ and thus the image $\hat{\mathbf{x}}_i$ can be seen as a linear combination of the

output examples. Note that the transport might imply non linear transformations of the input space. Indeed there is no guarantee that there exists a matrix $\mathbf{T}$ such that $\forall \mathbf{x}_i \in T, \hat{\mathbf{x}}_i = \mathbf{T}\mathbf{x}_i$.

Here we propose to use this optimal transport approach to learn the matrix $\mathbf{\Gamma}$ between the learning examples T and the landmarks T'. We then obtain the weight matrix used to compute the virtual points as $\mathbf{W} = n'\mathbf{\Gamma}$. Note that in this case our metric learning approach can be seen as a an approximation of the result given by the optimal transport[1].

5.3.2 Using a Class-based Representation Space

For this second approach, we propose to define virtual points as the unit vectors of a space of dimension $|\mathcal{Y}|$, i.e. the number of classes in the problem. Let $\mathbf{e}_j \in \mathbb{R}^{|\mathcal{Y}|}$ be such a unit vector ($1 \leq j \leq |\mathcal{Y}|$), i.e. a vector where all the attributes are 0 except for one attribute j which is set to 1, to which we associate a class label from $\mathcal{Y}$. In this case the $\mathbf{e}_j$ vectors can also be seen as the vertices of a standard $(|\mathcal{Y}|-1)$-simplex. For any learning example $(\mathbf{x}_i, y_i)$, we define $f_{\mathcal{V}}(\mathbf{x}_i, y_i) = \mathbf{e}_{\#y_i}$ where $\#y_i = j$ if $\mathbf{e}_j$ is mapped with the class y_i. Thus, we have exactly $|\mathcal{Y}|$ virtual points, each one corresponding to a unit vector and a class label.

We call this approach the class-based representation space method. If the number of classes is smaller than the number of dimensions used to represent the learning examples, then our method will also perform dimensionality reduction. Furthermore, our approach will try to project all the examples of one class on the same axis while examples of other classes will tend to be projected on different axes. The underlying intuition behind the new space defined by $\mathbf{L}_V$ is to make each attribute discriminant for one class. The interest of this approach is illustrated in Figure 5.2.

In this section we proposed two methods to define the virtual points but other approaches could be considered. For example Kusner et al. (2014) proposed a solution to compress a dataset by considering only a small number of examples in order to speed up nearest neighbours classification. Using this compressed dataset could be a way to define virtual points which summarize well the behaviour of the examples in each class. In the next section we show that our approach is theoretically founded.

5.4 Theoretical Analysis

In this section, we propose to theoretically show the interest of our approach by proving that the learned metric generalizes well, Section 5.4.1, and that it is possible to link it to a more classic metric learning formulation, Section 5.4.2.

5.4.1 Generalization Bound

In this section we show that a metric learned with Problem (5.1) generalizes well. To this extent we use the uniform stability framework presented in Section 1.3. In the following we

[1] In Chapter 6 we elaborate upon this idea by jointly learning the metric and the optimal transport.

assume that $\|\mathbf{x}\|_2 \leq B_{\mathbf{x}}$ and $\|\mathbf{v}\|_2 \leq B_{\mathbf{v}}$. Before proving that our approach is uniformly stable we start by presenting two lemmas showing that our loss is bounded and k-lipschitz continuous (Definition A.1).

Lemma 5.1 (Bounded loss function)**.** *Let $\mathbf{L}_V$ be the metric learned with Problem* (5.1) *with training set V, we have that for any example $(\mathbf{x}, \mathbf{v}) \sim \mathcal{D}_{\mathcal{V}}$:*

$$l\left(\mathbf{L}_V, (\mathbf{x}, \mathbf{v})\right) \leq B$$

with $B = B_{\mathbf{v}}^2 \left(1 + \frac{B_{\mathbf{x}}}{\sqrt{\lambda}}\right)^2$.

Proof. The proof of this lemma can be found in Appendix D.2. □

Lemma 5.2 (k-lipschitz continuity)**.** *Our loss is k-lipschitz with $k = 2B_{\mathbf{v}}B_{\mathbf{x}}\left(1 + \frac{B_{\mathbf{x}}}{\sqrt{\lambda}}\right)$.*

Proof. The proof of this lemma can be found in Appendix D.3. □

We can now show that our algorithm is uniformly stable (Definition 1.3).

Lemma 5.3 (Uniform stability)**.** *Our algorithm has a uniform stability in $\beta = \frac{8B_{\mathbf{v}}^2 B_{\mathbf{x}}^2}{\lambda n}\left(1 + \frac{B_{\mathbf{x}}}{\sqrt{\lambda}}\right)^2$.*

Proof. The proof of this lemma can be found in Appendix D.4. □

We can now prove our generalization bound.

Theorem 5.2 (Generalization bound)**.** *Let $\|\mathbf{v}\|_2 \leq B_{\mathbf{v}}$ for any $\mathbf{v} \in \mathcal{V}$ and $\|\mathbf{x}\|_2 \leq B_{\mathbf{x}}$ for any $\mathbf{x} \in \mathcal{X}$. Let $\mathbf{L}_V$ be the optimal solution of Problem* (5.1)*. With probability $1 - \delta$ we have:*

$$L_{\mathcal{V}}(\mathbf{L}_V) \leq \hat{L}_V(\mathbf{L}_V) + \frac{8B_{\mathbf{v}}^2 B_{\mathbf{x}}^2}{\lambda n}\left(1 + \frac{B_{\mathbf{x}}}{\sqrt{\lambda}}\right)^2 + \left(\left(\frac{16B_{\mathbf{x}}^2}{\lambda} + 1\right) B_{\mathbf{v}}^2 \left(1 + \frac{B_{\mathbf{x}}}{\sqrt{\lambda}}\right)^2\right)\sqrt{\frac{\ln \frac{1}{\delta}}{2n}}.$$

Proof. This theorem is a direct application of Theorem 1.1 (Bousquet and Elisseeff, 2002b) using the bound on the loss presented in Lemma 5.1 and the uniform stability of our algorithm proven in Lemma 5.3. □

We obtain a rate of convergence in $\mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$ which is standard with this kind of bounds.

Kernelized case Recall that in the linear case we assumed that $\|\mathbf{x}\|_2 \leq B_{\mathbf{x}}$. In the kernelized case, we only have to bound $\|\phi(\mathbf{x})\|_2$ where ϕ is the projection function associated to the used kernel. A common assumption (Audiffren and Kadri, 2013) when studying kernels is that $\exists\kappa$ such that $0 < \kappa < \infty$ and $k(\mathbf{x}, \mathbf{x}) \leq \kappa^2$. Hence, we have $\|\phi(\mathbf{x})\|_2^2 \leq \kappa^2$. Thus setting $B_{\mathbf{x}} = \kappa$ allows us to use the same proof than in the linear case leading us to the same generalization bound (the only difference being the value of $B_{\mathbf{x}}$).

5.4.2 Link with a Classic Metric Learning Formulation

In this section we show that it is possible to bound the true risk of a classic metric learning approach with the empirical risk of our formulation. Most of the classic metric learning approaches make use of a notion of margin between similar and dissimilar examples. Hence, similar examples have to be close to each other, i.e. at a distance smaller than a margin γ_1, and dissimilar examples have to be far from each other, i.e. at a distance greater than a margin γ_{-1}. Let $(\mathbf{x}, y)$ and $(\mathbf{x}', y')$ be two examples from $\mathcal{X} \times \mathcal{Y}$, using this notion of margin, we consider the following loss (Jin et al., 2009):

$$l\left(\mathbf{L}, (\mathbf{x}, y), (\mathbf{x}', y')\right) = \left[\delta_{yy'}(d^2\big(\mathbf{L}^T\mathbf{x}, \mathbf{L}^T\mathbf{x}'\big) - \gamma_{yy'})\right]_+ \tag{5.15}$$

where $\delta_{yy'} = 1$ if $y = y'$ and -1 otherwise, $\gamma_{yy'} = \gamma_1$ if $y = y'$ and γ_{-1} otherwise and d is the standard euclidean distance. The latter is the desired margin between examples. As introduced before, we consider that $\gamma_{yy'}$ takes a big value when the examples are dissimilar, i.e. when $\delta_{yy'} = -1$, and a small value when the examples are similar, i.e. when $\delta_{yy'} = 1$.

In the following we set, up to some constants, the margin between similar examples as the maximum distance between two virtual points associated to the same class and the margin between dissimilar examples as the minimum distance between two virtual points associated to different classes. Then we show that it is possible to bound, up to a constant factor, the true risk associated with the previous loss by the empirical risk of our approach.

Theorem 5.3 (Link with a classic metric learning approach)**.** *Let $\mathcal{D}_\mathcal{T}$ be a distribution over $\mathcal{X} \times \mathcal{Y}$. Let $\mathcal{V} \subset \mathbb{R}^{d'}$ be a finite set of virtual points and $f_\mathcal{V}$ is defined as $f_\mathcal{V}(\mathbf{x}, y) = \mathbf{v}$, $\mathbf{v} \in \mathcal{V}$. Let $\|\mathbf{v}\|_2 \leq B_\mathbf{v}$ for any $\mathbf{v} \in \mathcal{V}$ and $\|\mathbf{x}\|_2 \leq B_\mathbf{x}$ for any $\mathbf{x} \in \mathcal{X}$. Let $\gamma_1 = 2\max_{\mathbf{x},\mathbf{x}',y=y'} \|\mathbf{v} - \mathbf{v}'\|_2^2$ and $\gamma_{-1} = \frac{1}{2}\min_{\mathbf{x},\mathbf{x}',y\neq y'} \|\mathbf{v} - \mathbf{v}'\|_2^2$. Let $\mathbf{L}_V$ be the optimal solution of Problem (5.1), we have with probability $1 - \delta$:*

$$\begin{aligned}
&\mathop{\mathbb{E}}_{(\mathbf{x},y),(\mathbf{x}',y')\sim\mathcal{D}_\mathcal{T}} \left[\delta_{yy'}(d^2\big(\mathbf{L}_V^T\mathbf{x}, \mathbf{L}_V^T\mathbf{x}'\big) - \gamma_{yy'})\right]_+ \\
&\qquad \leq 8\left(\hat{L}_V(\mathbf{L}_V) + \frac{8B_\mathbf{v}^2B_\mathbf{x}^2}{\lambda n}\left(1 + \frac{B_\mathbf{x}}{\sqrt{\lambda}}\right)^2 + \left(\left(\frac{16B_\mathbf{x}^2}{\lambda} + 1\right)B_\mathbf{v}^2\left(1 + \frac{B_\mathbf{x}}{\sqrt{\lambda}}\right)^2\right)\sqrt{\frac{\ln\frac{1}{\delta}}{2n}})\right).
\end{aligned}$$

Proof. First of all, let us consider two examples $\mathbf{x}$ and $\mathbf{x}'$ and their associated virtual points $\mathbf{v}$ and $\mathbf{v}'$.

Using the fact that distances respect the triangle inequality, one can obtain:

$$d\big(\mathbf{L}_V^T\mathbf{x}, \mathbf{L}_V^T\mathbf{x}'\big) \leq d\big(\mathbf{L}_V^T\mathbf{x}, \mathbf{v}\big) + d\big(\mathbf{v}, \mathbf{v}'\big) + d\big(\mathbf{v}', \mathbf{L}_V^T\mathbf{x}'\big)\,.$$

Then squaring both sides of the inequality gives:

$$\begin{aligned}
d^2\big(\mathbf{L}_V^T\mathbf{x}, \mathbf{L}_V^T\mathbf{x}'\big) \leq\ & d^2\big(\mathbf{L}_V^T\mathbf{x}, \mathbf{v}\big) + d^2\big(\mathbf{v}, \mathbf{v}'\big) + d^2\big(\mathbf{v}', \mathbf{L}_V^T\mathbf{x}'\big) \\
&+ 2(d\big(\mathbf{L}_V^T\mathbf{x}, \mathbf{v}\big) + d\big(\mathbf{v}', \mathbf{L}_V^T\mathbf{x}'\big))d\big(\mathbf{v}, \mathbf{v}'\big) + 2d\big(\mathbf{L}_V^T\mathbf{x}, \mathbf{v}\big)\,d\big(\mathbf{v}', \mathbf{L}_V^T\mathbf{x}'\big)\,.
\end{aligned}$$

Finally, using Legendre identity[2] twice, we obtain:

$$d^2\big(\mathbf{L}_V^T\mathbf{x}, \mathbf{L}_V^T\mathbf{x}'\big) \leq 4d^2\big(\mathbf{L}_V^T\mathbf{x}, \mathbf{v}\big) + 2d^2\big(\mathbf{v}, \mathbf{v}'\big) + 4d^2\big(\mathbf{v}', \mathbf{L}_V^T\mathbf{x}'\big)\,.$$

[2]Legendre identity is $(a+b)^2 - (a-b)^2 = 4ab$ from which we deduce $a^2 + b^2 \geq 2ab$.

Similarly, switching the role of $d\big(\mathbf{L}_V^T\mathbf{x}, \mathbf{L}_V^T\mathbf{x}'\big)$ and $d(\mathbf{v}, \mathbf{v}')$ we have:

$$\begin{aligned} & d^2\big(\mathbf{v}, \mathbf{v}'\big) \leq 4d^2\big(\mathbf{L}_V^T\mathbf{x}, \mathbf{v}\big) + 2d^2\big(\mathbf{L}_V^T\mathbf{x}, \mathbf{L}_V^T\mathbf{x}'\big) + 4d^2\big(\mathbf{v}', \mathbf{L}_V^T\mathbf{x}'\big) \\ \Leftrightarrow \quad & -d^2\big(\mathbf{L}_V^T\mathbf{x}, \mathbf{L}_V^T\mathbf{x}'\big) \leq 2d^2\big(\mathbf{L}_V^T\mathbf{x}, \mathbf{v}\big) + 2d^2\big(\mathbf{v}', \mathbf{L}_V^T\mathbf{x}'\big) - \frac{1}{2}d^2\big(\mathbf{v}, \mathbf{v}'\big) \\ \Leftrightarrow \quad & -d^2\big(\mathbf{L}_V^T\mathbf{x}, \mathbf{L}_V^T\mathbf{x}'\big) \leq 4d^2\big(\mathbf{L}_V^T\mathbf{x}, \mathbf{v}\big) + 4d^2\big(\mathbf{v}', \mathbf{L}_V^T\mathbf{x}'\big) - \frac{1}{2}d^2\big(\mathbf{v}, \mathbf{v}'\big) \end{aligned}$$

Now, let us consider $\mathbf{x}$ and $\mathbf{x}'$ two examples of the same class, i.e. $\delta_{yy'} = 1$, we have:

$$\begin{aligned} \big[\delta_{yy'}(d^2\big(\mathbf{L}_V^T\mathbf{x}, \mathbf{L}_V^T\mathbf{x}'\big) - \gamma_{yy'})\big]_+ &= \big[d^2\big(\mathbf{L}_V^T\mathbf{x}, \mathbf{L}_V^T\mathbf{x}'\big) - \gamma_1\big]_+ \\ &\leq \big[4d^2\big(\mathbf{L}_V^T\mathbf{x}, \mathbf{v}\big) + 4d^2\big(\mathbf{v}', \mathbf{L}_V^T\mathbf{x}'\big) + 2d^2\big(\mathbf{v}, \mathbf{v}'\big) - \gamma_1\big]_+ \\ &\qquad (\gamma_1 \geq 2d^2(\mathbf{v}, \mathbf{v}').) \\ &\leq 4d^2\big(\mathbf{L}_V^T\mathbf{x}, \mathbf{v}\big) + 4d^2\big(\mathbf{v}', \mathbf{L}_V^T\mathbf{x}'\big)\,. \end{aligned} \tag{5.16}$$

Similarly, we consider $\mathbf{x}$ and $\mathbf{x}'$ two examples of different classes, i.e. $\delta_{yy'} = -1$, and we obtain:

$$\begin{aligned} \big[\delta_{yy'}(d^2\big(\mathbf{L}_V^T\mathbf{x}, \mathbf{L}_V^T\mathbf{x}'\big) - \gamma_{yy'})\big]_+ &= \big[-d^2\big(\mathbf{L}_V^T\mathbf{x}, \mathbf{L}_V^T\mathbf{x}'\big) + \gamma_{-1}\big]_+ \\ &\leq \left[4d^2\big(\mathbf{L}_V^T\mathbf{x}, \mathbf{v}\big) + 4d^2\big(\mathbf{v}', \mathbf{L}_V^T\mathbf{x}'\big) - \frac{1}{2}d^2\big(\mathbf{v}, \mathbf{v}'\big) + \gamma_{-1}\right]_+ \\ &\qquad (\gamma_{-1} \leq \tfrac{1}{2}d^2(\mathbf{v}, \mathbf{v}').) \\ &\leq 4d^2\big(\mathbf{L}_V^T\mathbf{x}, \mathbf{v}\big) + 4d^2\big(\mathbf{v}', \mathbf{L}_V^T\mathbf{x}'\big)\,. \end{aligned} \tag{5.17}$$

Noting that we obtain the same inequality for similar and dissimilar examples and taking the expectation on both sides gives:

$$\begin{aligned} & \mathop{\mathbb{E}}_{(\mathbf{x},y),(\mathbf{x}',y')\sim\mathcal{D}_\mathcal{T}} \Big[\delta_{yy'}(d^2\big(\mathbf{L}_V^T\mathbf{x}, \mathbf{L}_V^T\mathbf{x}'\big) - \gamma_{\delta_{yy'}})\Big]_+ \\ &\leq \mathop{\mathbb{E}}_{(\mathbf{x},y),(\mathbf{x}',y')\sim\mathcal{D}_\mathcal{T}} 4d^2\big(\mathbf{L}_V^T\mathbf{x}, \mathbf{v}\big) + 4d^2\big(\mathbf{v}', \mathbf{L}_V^T\mathbf{x}'\big) \\ &= \mathop{\mathbb{E}}_{(\mathbf{x},y),(\mathbf{x}',y')\sim\mathcal{D}_\mathcal{T}} 4d^2\big(\mathbf{L}_V^T\mathbf{x}, \mathbf{v}\big) + \mathop{\mathbb{E}}_{(\mathbf{x},y),(\mathbf{x}',y')\sim\mathcal{D}_\mathcal{T}} 4d^2\big(\mathbf{v}', \mathbf{L}_V^T\mathbf{x}'\big) \\ &= 8 \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{D}_\mathcal{T}} d^2\big(\mathbf{L}_V^T\mathbf{x}, \mathbf{v}\big) \\ &= 8L_\mathcal{V}(\mathbf{L}_V). \end{aligned} \tag{5.18}$$

Applying Theorem 5.2 to the last inequality gives the theorem. □

In Theorem 5.3, we can notice that the margins are related to the distances between virtual points and correspond to the ideal margins, i.e. the margins that we would like to achieve after the learning step. In practice, we can also define $\hat{\gamma}_1$ and $\hat{\gamma}_{-1}$ the observed margins obtained after the learning step. All the similar examples are in a sphere centred in their

corresponding virtual point and of diameter $\hat{\gamma}_1 = 2 \max_{(\mathbf{x},\mathbf{v})\in V} \left\|\mathbf{x}^T\mathbf{L}_V - \mathbf{v}^T\right\|_2$. Similarly, the distance between spheres of dissimilar examples is $\hat{\gamma}_{-1} = \min_{\mathbf{v},\mathbf{v}',\mathbf{v}\neq\mathbf{v}'} \|\mathbf{v} - \mathbf{v}'\|_2 - \hat{\gamma}_1$. As a consequence, even if we do not use cannot-link constraints, our algorithm is able to push reasonably far away dissimilar examples by minimizing the diameter of the sphere around similar examples.

In this section we have shown that the metrics learned by our algorithm generalize well and that our method can be theoretically linked to a classic metric learning approach. In the next section we empirically show the interest of our approach for several classification problems.

5.5 Experiments

In this section we propose an empirical evaluation of our method. On the one hand in Subsection 5.5.1 we compare it to several methods on a task of metric learning for classification and we provide some graphics showing 2D projections of the space learned by RVML-Lin-Class and RVML-RBF-Class on one dataset illustrating the capability of these approaches to learn discriminative attributes. On the other hand in Subsection 5.5.2 we further study the interest of explicitly choosing the virtual points using the methods presented in Section 5.3.

In all these experiments we consider 13 different datasets coming from either the UCI Lichman (2013) repository or used in recent works in metric learning Kedem et al. (2012); Shi et al. (2014); Bellet et al. (2012). For isolet, splice and svmguide1 we have access to a standard training/test partition, for the other datasets we use a 70% training/30% test partition, we perform the experiments on 10 different splits and we average the result. We normalize the examples with respect to the training set by subtracting for each attribute its mean and dividing by 3 times its standard deviation. We set our regularization parameter λ with a 5-fold cross validation on the training set. After the metric learning step, we use a 1-nearest neighbour classifier to assess the performance of the metric and we report the accuracy obtained. Note that we also report the mean accuracy over the 13 tasks. Even if we are conscious that the different datasets consider different classification problems, it gives a rough idea of the global performance of the compared approaches.

5.5.1 Metric Learning for Classification

Here we consider the problem of learning a metric for a classification task. We consider two sets of experiments. In the first set we consider our linear formulation used with the two virtual points selection methods presented in this chapter, namely RVML-Lin-OT based on optimal transport (Section 5.3.1) and RVML-Lin-Class using the class-based representation space method (Section 5.3.2). We compare our two approaches to three baselines:

- 1-NN: A 1-nearest neighbour classifier without metric learning,

- LMNN (Weinberger et al., 2005),
- SCML (Shi et al., 2014).

In a second set we consider the kernelized versions of RVML, namely RVML-RBF-OT and RVML-RBF-Class, based respectively on optimal transport and class-based representation space methods with a RBF kernel with the parameter σ fixed as the mean of all pairwise training set euclidean distances (Kar and Jain, 2011). We compare them to non linear methods using a KPCA with a RBF kernel[3] as a pre-process. The number of dimensions is fixed as the one of the original space for high dimensional datasets (more than 100 attributes), to 3 times the original dimension when the dimension is smaller (between 5 and 100 attributes) and to 4 times the original dimension for the lowest dimensional datasets (less than 5 attributes). We also consider some local metric learning methods. Hence we compare our approach with 4 non-linear baselines:

- 1-NN-KPCA: A 1-nearest neighbour classifier in the KPCA space without metric learning,
- LMNN-KPCA: LMNN in the KPCA-space,
- GB-LMNN: A non linear version of LMNN(Kedem et al., 2012),
- SCMLLOCAL: The local version of SCML(Shi et al., 2014).

For all the baselines (linear and non linear), we use the implementations available online letting them handle hyper-parameters tuning.

The results for linear methods are presented in Table 5.1 while Table 5.2 gives the results obtained with the non linear approaches. In each table, the best result on each line is highlighted with a bold font while the second to best result is underlined. A star indicates either that the best baseline is significantly better than our best result or that our best result is significantly better than the best baseline according to classic significance tests (the p-value being fixed at 0.05).

We can make the following remarks. In the linear setting, our approaches are very competitive with state of the art approaches and RVML-Lin-OT tends to be the best on average even though SCML also performs very well on some datasets (the difference is not significant). RVML-Lin-Class performs slightly less on average. Considering now the non linear methods, our approaches improve their performance and are significantly better than the others on average, RVML-RBF-Class has the best average behaviour in this setting. These experiments show that our regressive formulation is very competitive and is even able to improve state of the art performances in a non linear setting.

Considering the virtual points selection, we can observe that the OT formulation performs better than the class-based representation space one in the linear case, while it is the opposite

[3]With the σ parameter fixed as previously to the mean of all pairwise training set euclidean distances.

Table 5.1: Comparison of our approach with several baselines in the linear setting. The best result is highlighted with a bold font while the second to best result is underlined. A star indicates that the best result for the baseline is significantly better than our best result or that our best result is significantly better than the best baseline result.

	Baselines			Our approach	
Base	1-NN	LMNN	SCML	RVML-Lin-OT	RVML-Lin-Class
Amazon	41.51 ± 3.24	65.50 ± 2.28	71.68 ± 1.86	71.62 ± 1.34	**73.09 ± 2.49**
Breast	95.49 ± 0.79	95.49 ± 0.89	**96.50 ± 0.64***	95.24 ± 1.21	95.34 ± 0.95
Caltech	18.04 ± 2.20	49.68 ± 2.76	52.84 ± 1.61	52.51 ± 2.41	**55.41 ± 2.55***
DSLR	29.61 ± 4.38	**76.08 ± 4.79**	65.10 ± 9.00	74.71 ± 5.27	75.29 ± 5.08
Ionosphere	86.23 ± 1.95	88.02 ± 3.02	**90.38 ± 2.55***	87.36 ± 3.12	82.74 ± 2.81
Isolet	88.97	**95.83**	89.61	91.40	94.61
Letters	94.74 ± 0.27	**96.43 ± 0.28***	96.13 ± 0.20	90.25 ± 0.60	95.51 ± 0.26
Pima	69.91 ± 1.69	70.04 ± 2.20	69.22 ± 2.60	**70.48** ± 3.19	69.57 ± 2.85
Scale	78.68 ± 2.66	78.20 ± 1.91	**93.39 ± 1.70***	90.05 ± 2.13	87.94 ± 1.99
Splice	71.17	82.02	**85.43**	84.64	78.44
Svmguide1	**95.12**	95.03	87.38	94.83	85.25
Wine	96.18 ± 1.59	98.36 ± 1.03	96.91 ± 1.93	**98.55 ± 1.67**	98.18 ± 1.48
Webcam	42.90 ± 4.19	85.81 ± 3.75	**90.43 ± 2.70**	88.60 ± 3.63	88.60 ± 2.69
mean	69.89	82.81	83.46	**83.86**	83.07

Table 5.2: Comparison of our approach with several baselines in the non linear case. The best result is highlighted with a bold font while the second to best result is underlined. A star indicates that the best result for the baseline is significantly better than our best result or that our best result is significantly better than the best baseline result.

	Baselines				Our approach	
Base	1NN-KPCA	LMNN-KPCA	GBLMNN	SCMLLOCAL	RVML-RBF-OT	RVML-RBF-Class
Amazon	20.27 ± 2.42	53.16 ± 3.73	65.53 ± 2.32	69.14 ± 1.74	73.51 ± 0.83	**76.22 ± 2.09***
Breast	92.43 ± 2.19	95.39 ± 1.32	95.58 ± 0.87	**96.31 ± 0.66**	95.73 ± 0.97	95.78 ± 0.92
Caltech	20.82 ± 8.29	29.88 ± 10.89	49.91 ± 2.80	50.56 ± 1.62	54.39 ± 1.89	**57.98 ± 2.22***
DSLR	64.90 ± 5.81	73.92 ± 7.57	76.08 ± 4.79	62.55 ± 6.94	70.39 ± 4.48	**76.67 ± 4.57**
Ionosphere	75.57 ± 2.79	85.66 ± 2.55	87.36 ± 3.02	90.94 ± 3.02	90.66 ± 3.10	**93.11 ± 3.30***
Isolet	68.70	96.28	96.02	91.40	95.96	**96.73**
Letter	95.39 ± 0.27	**97.17*** ± 0.18**	96.51 ± 0.25	96.63 ± 0.26	91.26 ± 0.50	96.09 ± 0.21
Pima	69.57 ± 2.64	69.48 ± 2.04	69.52 ± 2.27	68.40 ± 2.75	69.35 ± 2.95	**70.74 ± 2.36**
Scale	78.36 ± 0.88	88.10 ± 2.26	77.88 ± 2.43	93.86 ± 1.78	**95.19 ± 1.46***	94.07 ± 2.02
Splice	66.99	**88.97**	82.21	87.13	88.51	88.32
Svmguide1	**95.72**	95.60	95.00	87.40	95.67	95.05
Wine	92.18 ± 1.23	95.82 ± 2.98	98.00 ± 1.34	96.55 ± 2.00	**98.91 ± 1.53**	98.00 ± 1.81
Webcam	73.55 ± 4.57	84.52 ± 3.83	85.81 ± 3.75	88.71 ± 2.83	88.71 ± 4.28	**88.92 ± 2.91**
mean	70.34	81.07	82.72	83.04	85.25	**86.74**

in the non linear case. We think that this can be explained by the fact that the OT approach generates more virtual points in a potentially non linear way which brings more expressiveness for the linear case. On the other hand, in the non linear one, the relative small number of virtual points used by the class-based method seems to induce a better regularization.

To illustrate the capability of RVML-Lin-Class and RVML-RBF-Class to learn discriminative attributes we propose to select two dimensions out of the 26 of the space learned by these approaches on the isolet dataset. We selected 3 pairs of axis and the images obtained are presented in Figure 5.2. On the same line, we plot two images corresponding to the same axis pair: on the left column for RVML-Lin-Class and on the right column for RVML-RBF-Class. Note that for each axis, there is only one class for which the value of the attribute tends to be 1, for all the other classes this feature tends to be 0. Furthermore, we can note that the kernelized version of our metric outputs a more discriminative space: the examples are brought closer to their corresponding virtual point than in the linear version.

5.5.2 Interest of Explicitly Choosing Virtual Points

In the previous subsection we have seen that our approach is very competitive. Here we demonstrate the interest of explicitly choosing the virtual points.

Class based virtual points In Globerson and Roweis (2005) the authors propose to collapse similar examples on a single point, an implicit virtual point, while pushing far away dissimilar examples. This behaviour can, in fact, be achieved by any margin based metric learning approach by setting the margin between similar examples to 0 and the margin between dissimilar examples to a high value. Thus to illustrate the interest of using explicit virtual points, we propose to compare our approach to Information Theoretic Metric Learning, ITML (Davis et al., 2007), when considering the aforementioned margins (ITML-Collapse). For the sake of completeness we also consider ITML with tuned margins (ITML). The results are presented in Table 5.3 and show that, on average, ITML-Collapse and ITML are less accurate than RVML-Lin-Class hinting that considering explicit virtual points is better than considering implicit ones but also that learning a metric where each axis is discriminative is indeed beneficial for classification.

Optimal transport based virtual points To further assess the interest of using our OT based formulation to select virtual points and associate them to examples, we propose to compare it with a random based approach (Random). In this latter setting, we randomly select a subset of examples for each class to act as virtual points and we randomly associate each example of this class to these virtual points. The results in the linear case are presented in Table 5.4 while the results in the non linear case are presented in Table 5.5. Overall, randomly selecting the virtual points is less accurate than using the OT based formulation. This is especially true in the linear case where the metric is less expressive than in the

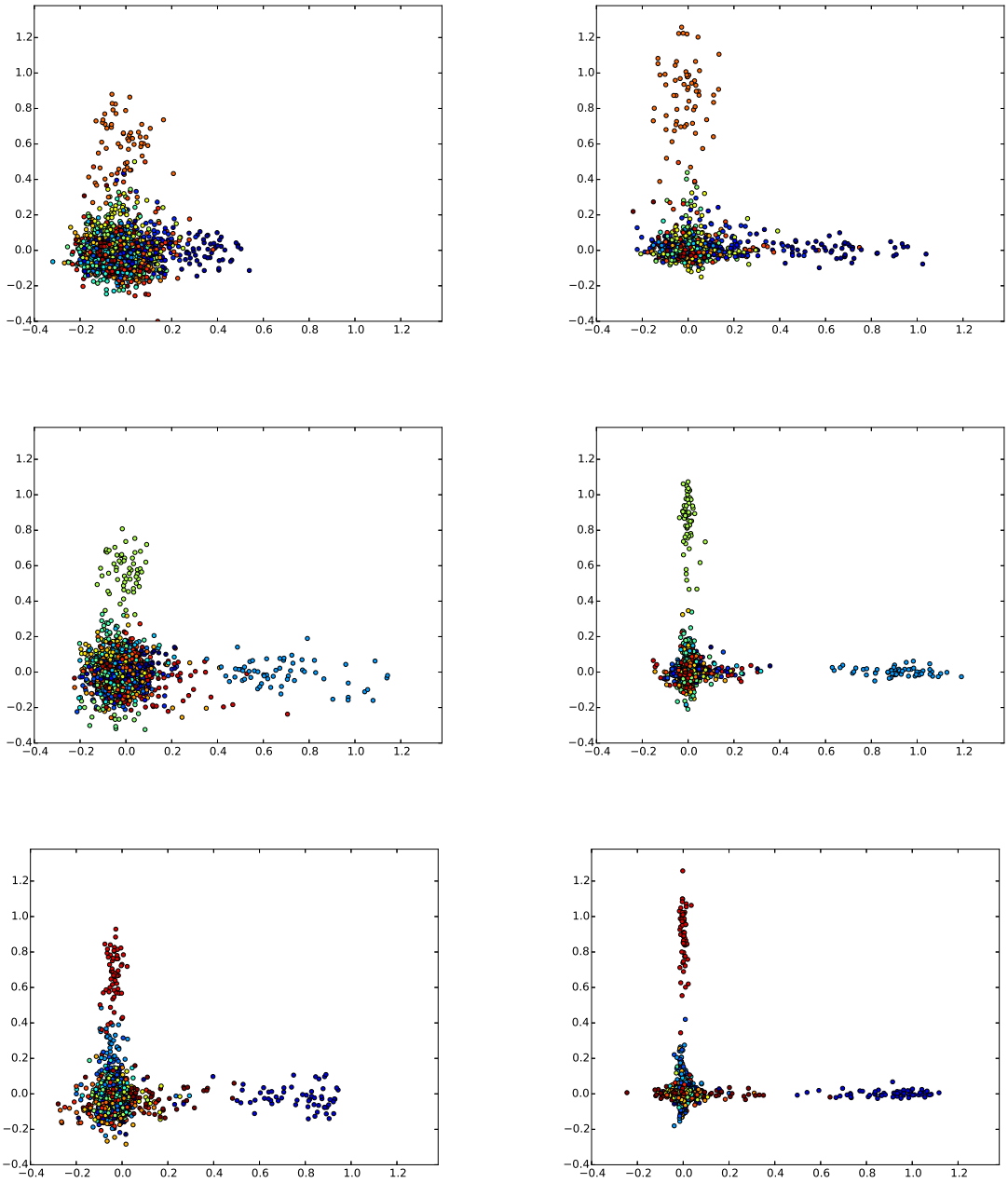


Figure 5.2: In the learned space from the isolet dataset, we randomly select 2 attributes three times and plot the 2D projection on each pair. The first line corresponds to features 1 and 20, the second line to features 7 and 14 and the third line to features 2 and 23. The left column corresponds to the space learned by RVML-Lin-Class (linear) and the right column to the one learned by RVML-RBF-Class (non linear).

Table 5.3: Comparison between a method with explicit virtual points (RVML-Lin-Class) and a method with implicit virtual points (ITML-Collapse). The best result is highlighted with a bold font.

Base	RVML-Lin-Class	ITML-Collapse	ITML
Amazon	**73.09 ± 2.49**	57.97 ± 3.36	65.91 ± 2.64
Breast	95.34 ± 0.95	94.56 ± 1.41	**95.49 ± 1.15**
Caltech	**55.41 ± 2.55**	37.34 ± 2.01	47.31 ± 2.75
DSLR	75.29 ± 5.08	**77.25 ± 4.15**	77.25 ± 4.91
Ionosphere	82.74 ± 2.81	85.75 ± 6.23	**88.11 ± 1.68**
Isolet	**94.61**	74.53	92.88
Letters	95.51 ± 0.26	**95.67 ± 0.30**	95.00 ± 0.64
Pima	69.57 ± 2.85	**71.08 ± 2.13**	70.26 ± 1.38
Scale	**87.94 ± 1.99**	87.51 ± 4.39	87.67 ± 2.71
Splice	**78.44**	66.80	71.49
Svmguide1	85.25	94.62	**95.00**
Wine	**98.18 ± 1.48**	85.91 ± 3.74	96.91 ± 1.93
Webcam	88.60 ± 2.69	**97.64 ± 2.43**	86.56 ± 2.88
mean	**83.07**	78.97	82.30

Table 5.4: Comparison of our OT based formulation to a random selection approach when learning a linear metric. The best result is highlighted with a bold font.

	OT based approach	Random		
Base	RVML-Lin-OT	1 VP per class	2 VP per class	5 VP per class
Amazon	71.62 ± 1.34	**74.23 ± 2.15**	72.92 ± 2.31	70.31 ± 2.82
Breast	95.24 ± 1.21	**95.34 ± 0.95**	95.29 ± 1.32	94.90 ± 1.92
Caltech	52.51 ± 2.41	**55.09 ± 2.38**	53.63 ± 2.12	49.59 ± 1.69
DSLR	**74.71 ± 5.27**	70.59 ± 6.06	63.53 ± 5.08	52.16 ± 8.68
Ionosphere	87.36 ± 3.12	82.74 ± 2.81	88.40 ± 4.05	**90.28 ± 3.33**
Isolet	91.40	92.75	**94.16**	92.43
Letters	90.25 ± 0.60	89.90 ± 1.02	90.54 ± 1.24	**91.13 ± 0.74**
Pima	**70.48 ± 3.19**	69.57 ± 2.85	69.35 ± 2.44	69.26 ± 2.60
Scale	**90.05 ± 2.13**	88.10 ± 2.57	89.47 ± 2.99	89.21 ± 2.68
Splice	**84.64**	78.44	78.94	80.87
Svmguide1	**94.83**	85.25	86.90	94.70
Wine	**98.55 ± 1.67**	98.55 ± 1.43	97.64 ± 2.43	98.00 ± 1.34
Webcam	88.60 ± 3.63	**88.92 ± 3.21**	86.24 ± 2.95	81.18 ± 3.56
mean	**83.86**	82.27	82.08	81.08

kernelized case and thus requires more meaningful virtual points. Hence, selecting virtual points and correctly associating them to the examples is key to obtain a good performance.

5.6 Conclusion

In this chapter we presented a new metric learning approach based on a regression and aiming at bringing closer the learning examples to some a priori defined virtual points. The number of constraints has the advantage of growing linearly with the size of the learning set in opposition to the quadratic grow of standard must-link cannot-link approaches. Moreover, our method can be solved in closed form and can be easily kernelized allowing us to deal with non linear problems. Additionally, we proposed two methods to define the virtual points: one making use of recent advances in the field of optimal transport and one based on unit vectors of a class-based representation space allowing one to perform directly some dimensionality reduction. Theoretically, we have shown that the metrics learned with our approach generalize well and that we are able to link our empirical risk to the true risk of a classic metric learning formulation. Finally, we empirically show that explicitly choosing the virtual points is important and that our approach is competitive with the state of the art in the linear case and outperforms some classic approaches in the non linear one.

Table 5.5: Comparison of our OT based formulation to a random selection approach when learning a non linear metric. The best result is highlighted with a bold font.

	OT based approach	Random		
Base	RVML-RBF-OT	1 VP per class	2 VP per class	5 VP per class
Amazon	73.51 ± 0.83	**75.74 ± 2.35**	72.68 ± 2.02	70.07 ± 2.86
Breast	95.73 ± 0.97	95.73 ± 1.07	**95.83 ± 0.80**	95.58 ± 1.38
Caltech	54.39 ± 1.89	**58.33 ± 2.05**	53.98 ± 3.18	50.35 ± 1.89
DSLR	**70.39 ± 4.48**	65.29 ± 7.51	58.24 ± 7.79	48.82 ± 8.03
Ionosphere	**90.66 ± 3.10**	90.57 ± 3.05	89.25 ± 3.73	90.38 ± 3.26
Isolet	95.96	**96.99**	96.54	95.25
Letters	91.26 ± 0.50	91.77 ± 0.43	91.87 ± 0.52	**92.04 ± 0.62**
Pima	69.35 ± 2.95	70.82 ± 4.60	**71.26 ± 2.84**	70.00 ± 2.56
Scale	**95.19 ± 1.46**	93.39 ± 2.19	91.96 ± 1.69	91.32 ± 1.95
Splice	**88.51**	88.37	88.46	87.22
Svmguide1	**95.67**	95.03	95.55	95.88
Wine	**98.91 ± 1.53**	97.82 ± 1.88	97.27 ± 1.96	97.82 ± 1.67
Webcam	**88.71 ± 4.28**	87.31 ± 2.99	83.01 ± 3.28	76.67 ± 4.78
mean	**85.25**	85.17	83.53	81.65

We think that this work opens the door to design new metric learning formulations, in particular the definition of the virtual points can bring a way to control some particular properties of the metric (rank, locality, discriminative power, ...). As a consequence, this aspect opens new issues which are in part related to landmark selection problems but also to the ability to embed expressive semantic constraints to satisfy by means of the virtual points. Other perspectives include the development of a specific solver, of online versions, the use of low rank-inducing norms or the conception of new local metric learning methods. Another direction would be to study similarity learning extensions to perform linear classification with generalization guarantees on the classifier such as in Bellet et al. (2012); Balcan et al. (2008).

In this chapter we have addressed the problem of explicitly controlling the behaviour of a metric by introducing the notion of virtual points. It allows us to design metrics with a behaviour well tailored to the task at hand, for example classification. However choosing these virtual points can be difficult and the methods proposed in Section 5.3 might not always be satisfactory. In the next chapter we propose to build upon RVML and the optimal transport based virtual points to design a new algorithm able to learn a transformation which brings closer two distributions in a principled way.

Chapter 6

Mapping Estimation for Discrete Optimal Transport

This chapter is based on the following publication

Michaël Perrot, Nicolas Courty, Rémi Flamary, and Amaury Habrard. Mapping estimation of discrete optimal transport. In *Advances in Neural Information Processing Systems (NIPS-16)*, 2016

Abstract

In this chapter we propose to address the problem of learning a transformation from a Mahalanobis distance which follows some particular geometric transformations. Such a metric could be very beneficial for domain adaptation problems where the goal is to align the source and the target domains. Here we propose to consider geometric transformations which come from the result of an optimal transport problem. Indeed it is a reasonable procedure to align distributions and it has been shown to perform well in domain adaptation. Most of the computational approaches of optimal transport use the Kantorovich relaxation of the problem to learn a probabilistic coupling $\mathbf{\Gamma}$ between the training examples but do not address the problem of learning the transport map $f_{\mathcal{S}\to\mathcal{T}}$ linked to the original Monge problem. Consequently, the fact that the coupling can only be used to transform training examples and not for out of samples ones lowers the potential usage of such methods. In this chapter we propose to combine the most interesting features of each method and we propose a new framework to estimate the transport map, also called the mapping, of a coupling. This estimation takes the form of a matrix $\mathbf{L}$ which corresponds to a new metric in the source domain. In this case we show that our approach is similar to RVML, presented in Chapter 5, where we define the transformation of the examples induced by the coupling as the virtual points for each example. However instead of considering that this coupling is defined a priori, we jointly learn it along the metric. It results in a jointly convex formulation which can be efficiently optimized and has the beneficial effect of smoothing the result of optimal transport. Empirically, we show the interest and the relevance of our method in two tasks, namely unsupervised domain adaptation and image editing.

6.1 Introduction

Many metric learning approaches have focused on learning a linear transformation in the form of a Mahalanobis distance or a bilinear similarity (See Section 2.2). One can notice that these methods do not try to control this transformation with respect to some particular geometric transformations but rather try to bring closer similar examples and push far away dissimilar ones. However considering other kind of transformations might be relevant for some problems. This is for example the case in domain adaptation (See Section 1.5) where one has to estimate and overcome the shift between a source and a target distribution. In this context, a few works in metric learning have proposed to learn a metric in order to move closer source and target instances (Saenko et al., 2010; Kulis et al., 2011). However these methods often require some sort of supervision to associate the examples with each other and, as mentioned before, remain limited by the kind of constraints considered.

Among approaches able to align distributions, an interesting solution is to consider optimal transport based methods which have recently shown their interest in domain adaptation. The idea is to learn a transformation of the source examples such that the source and the target are aligned. This transformation takes the form of a coupling of minimal cost between source and target where the cost function is for example the euclidean distance between the examples. One of the main drawbacks of using this coupling in optimal transport is that it can only be used to map source examples which have been seen during the training process and it is not applicable to out-of-sample examples. Hence, despite showing good performances in practice (Courty et al., 2014b) this approach cannot be used when new examples have to be mapped from source to target domains.

In this chapter we propose to consider the best of both worlds by learning a transformation whose behaviour is controlled by the transport map implied by the coupling $\mathbf{\Gamma}$ of a discrete optimal transport problem. Our formulation is based on a jointly convex optimization problem which admits two appealing interpretations. One the one hand it can be seen as learning a linear mapping regularized by an optimal transport map[1]. On the other hand we can also see the approach as the computation of the optimal transport map regularized with respect to the definition of a mapping. Furthermore under some mild conditions on the set of transformations considered we will show some ties between this approach and RVML developed in Chapter 5. This formulation can be efficiently solved thanks to an alternating block-coordinate descent and actually benefits the two models. On the one hand we obtain smoother optimal transport maps which are compliant with a linear mapping usable as an out-of-sample transformation. This learned transformation is able to take into account some geometrical information captured by optimal transport. Another important aspect of our contribution is that it is in fact not limited to linear mappings as it can be kernelized. In this case it conveniently expresses non linear and out-of-sample transformations, thus enhancing the faithfulness to the true optimal transport map. See Figure 6.1 for an illustration of our

[1]This optimal transport map is implied by the coupling $\mathbf{\Gamma}$ but cannot be computed.

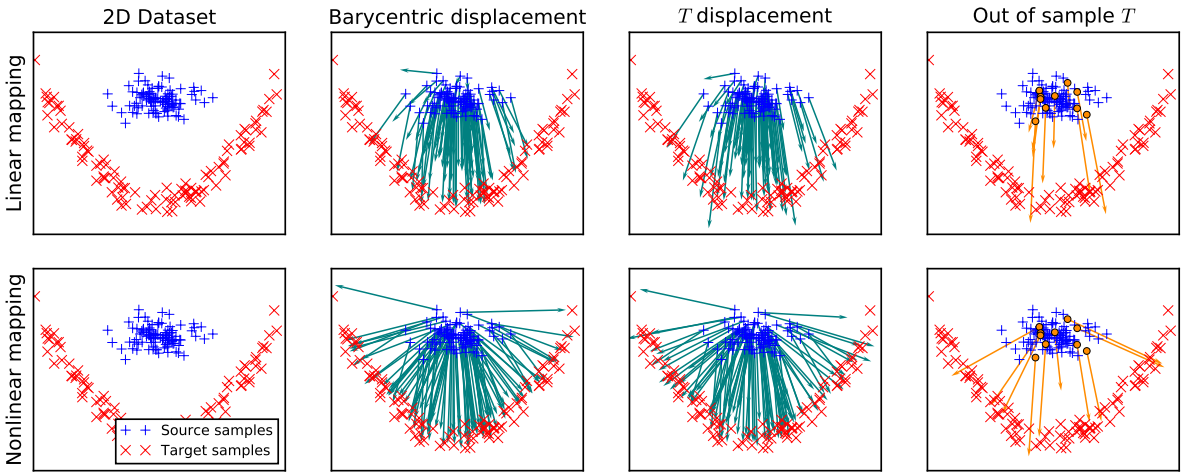


Figure 6.1: Illustration of our approach on the clown dataset when learning a linear transformation (top) and a non linear transformation obtained by kernelization (bottom). In both cases we considered the same original data, depicted in the first column, where the blue crosses correspond to the source examples and the red crosses to the target examples. The second and third column respectively show the couplings and the transformations jointly learned by our approach. The fourth column demonstrates the generalization ability of the transformations on new examples. Note that the couplings cannot be used on these examples that have not been seen during the learning process.

method. We also provide a brief discussion on the theoretical challenges behind our approach and we provide some empirical evidence of its interest in domain adaptation and in image editing.

The remaining of this chapter is organized as follows. Section 6.2 is dedicated to a presentation of the problem of optimal transport. In Section 6.3 we present our approach to jointly learn the coupling and the corresponding general transformation. Section 6.4 presents the optimisation scheme used to solve our formulation. Here we consider several possible transformations showing that our approach is close to RVML. In Section 6.5 we discuss some theoretical aspects of this work. In Section 6.6 we show the good behaviour of our approach in several experiments before concluding in Section 6.7.

6.2 Optimal Transport

In this section we present the problem of optimal transport. We start by recalling several recent approaches which successfully make use of it before formalising the problem.

In recent years optimal transport (Villani, 2009) has received a lot of attention in the machine learning community (e.g. (Canas and Rosasco, 2012; Cuturi, 2013a; Solomon et al.,

2014; Frogner et al., 2015)). This gain of interest comes from several nice properties of optimal transport when used as a divergence to compare discrete distributions. On the one hand it provides a sound and theoretically grounded way of comparing multivariate probability distributions without the need of estimating parametric versions. On the other hand by considering the geometry of the underlying space through a cost metric, it can encode useful informations about the nature of the problem. Optimal transport is usually expressed as an optimal cost functional but it also enjoys a dual variational formulation (Villani, 2009, Chapter 5).

Optimal transport has been proven to be useful in several settings. As a first example it corresponds to the Wasserstein distance in the space of probability distributions. Using this distance it is possible to compute means and barycentres (Cuturi and Doucet, 2014; Benamou et al., 2015) or to perform a PCA in the space of probability measures (Seguy and Cuturi, 2015). This distance has also been used in subspace identification problems for analysing the differences between distributions (Mueller and Jaakkola, 2015), in graph based semi-supervised learning to propagate histogram labels across nodes (Solomon et al., 2014) or as a way to define a loss function for multi-label learning (Frogner et al., 2015). As a second example optimal transport enjoys a variety of bounds for the convergence rate of empirical to population measures which can be used to derive new probabilistic bounds for the performance of unsupervised learning algorithms such as k-means (Canas and Rosasco, 2012). As a last example optimal transport is a mean of interpolation between distributions (McCann, 1997) that has been used in Bayesian inference (Reich, 2013), color transfer (Ferradans et al., 2014) or domain adaptation (Courty et al., 2014a).

On the computational side, one of the major gain for optimal transport is the recent development of regularized versions that lead to efficient algorithms Cuturi (2013a); Benamou et al. (2015); Cuturi and Peyré (2016). Most of optimal transport formulations are based on the computation of a (probabilistic) coupling matrix that can be seen as a bipartite graph between the bins of the distributions. This coupling, also denoted transportation matrix suffers from some drawbacks: it is always restricted to the data samples used to compute this map. In other words when a new dataset (or sample) is available, one has to recompute an optimal transport problem to deal with the new instances which can be prohibitive from some applications in particular when the task is similar or related. From a machine learning standpoint, this also means that we do not know how to have a good approximation of an optimal transport map computed from a small sample that can be generalized to unseen data. This is particularly critical when one considers large scale applications, or even medium-scales such as image editing problems. In this chapter, we bridge this gap by learning an explicit transformation that can be interpreted as a good approximation of the transport. As far as we know, this is the first approach that addresses directly this problem of out-of-sample mapping.

6.2.1 Formalisation

In this subsection we propose a more formal presentation of the problem of optimal transport. We present Monge's and Kantorovich's formulations which answer the problem of finding a map of minimal cost. We also present the notion of Barycentric mapping which corresponds to the transformation implied by the coupling of Kantorovich's formulation. First of all let $\mathcal{S}$ and $\mathcal{T}$ be the source and target domains respectively defined as the distribution $\mathcal{D}_\mathcal{S}$ on the space $\mathcal{X}^s$ and the distribution $\mathcal{D}_\mathcal{T}$ over the space $\mathcal{X}^t$. In this chapter we upper-script with s any element associated with the source domain and with t any element associated with the target domain.

Monge problem Let $\mathcal{X}^s \in \mathbb{R}^{d^s}$ and $\mathcal{X}^t \in \mathbb{R}^{d^t}$ be two separable metric spaces such that any probability measure on $\mathcal{X}^s$ (or $\mathcal{X}^t$) is a Radon measure. By considering a cost function $c : \mathcal{X}^s \times \mathcal{X}^t \to [0, \infty[$, Monge's formulation of the optimal transport problem is to find a transform map $f_{\mathcal{S}\to\mathcal{T}} : \mathcal{X}^s \to \mathcal{X}^t$ (also known as a push-forward operator) between two probability measures $\mathcal{D}_\mathcal{S}$ on $\mathcal{X}^s$ and $\mathcal{D}_\mathcal{T}$ on $\mathcal{X}^t$ realizing the infimum of the following function

$$\inf \left\{ \int_{\mathcal{X}^s} c(\mathbf{x}^s, f_{\mathcal{S}\to\mathcal{T}}(\mathbf{x}^s))\, d\mathcal{D}_\mathcal{S}(\mathbf{x}^s), \quad f_{\mathcal{S}\to\mathcal{T}} \# \mathcal{D}_\mathcal{S} = \mathcal{D}_\mathcal{T} \right\}. \tag{6.1}$$

When reaching this infimum, the corresponding map $f_{\mathcal{S}\to\mathcal{T}}$ is an optimal transport map. It associates one point from $\mathcal{X}^s$ to a single point in $\mathcal{X}^t$. Therefore, the existence of this map is not always guaranteed, as when for example $\mathcal{D}_\mathcal{S}$ is a Dirac and $\mathcal{D}_\mathcal{T}$ is not. As such, the existence of solutions for this problem can in general not be established when $\mathcal{D}_\mathcal{S}$ and $\mathcal{D}_\mathcal{T}$ are supported on a different number of Diracs. Yet, in a machine learning context, data samples usually form discrete distributions, but can be seen as observations of a regular, continuous (with respect to the Lebesgue measure) underlying distribution, thus fulfilling existence conditions (see (Villani, 2009, Chapter 9)). As such, assuming for the existence of $f_{\mathcal{S}\to\mathcal{T}}$ calls for a relaxation of the previous problem.

Kantorovich relaxation The Kantorovitch formulation of the optimal transportation Kantorovich (1942) is a convex relaxation of the Monge problem. Let us define Π as the set of all probabilistic couplings $\in \mathcal{P}(\mathcal{X}^s \times \mathcal{X}^t)$ the space of all joint distributions with marginals $\mathcal{D}_\mathcal{S}$ and $\mathcal{D}_\mathcal{T}$. The Kantorovitch problem seeks for a general coupling $f_{\mathcal{X}^s\times\mathcal{X}^t} \in \Pi$ between $\mathcal{X}^s$ and $\mathcal{X}^t$ solving the following problem:

$$\underset{f_{\mathcal{X}^s\times\mathcal{X}^t}\in\Pi}{\arg\min} \int_{\mathcal{X}^s\times\mathcal{X}^t} c\big(\mathbf{x}^s, \mathbf{x}^t\big)\, df_{\mathcal{X}^s\times\mathcal{X}^t}\big(\mathbf{x}^s, \mathbf{x}^t\big) \tag{6.2}$$

The optimal coupling always exists (Villani, 2009, Theorem 4.1). This leads to a simple writing of the optimal transport problem in the discrete case, i.e. whenever $\mathcal{D}_\mathcal{S}$ and $\mathcal{D}_\mathcal{T}$ are only accessible through discrete samples $S = \{\mathbf{x}_i^s\}_{i=1}^{n^s}$ and $T = \{\mathbf{x}_i^t\}_{i=1}^{n^t}$ (designed by $\mathbf{X}^s$ and $\mathbf{X}^t$ in matrix form with one example on each line). The corresponding empirical distributions can be written as $\widehat{\mathcal{D}_\mathcal{S}} = \sum_{i=1}^{n^s} p_i^s \delta_{\mathbf{x}_i^s}$ and $\widehat{\mathcal{D}_\mathcal{T}} = \sum_{i=1}^{n_t} p_i^t \delta_{\mathbf{x}_i^t}$ where $\delta_\mathbf{x}$ is the Dirac function at

location $\mathbf{x} \in \mathcal{X}$. p_i^s and p_i^t are probability masses associated to the i-th sample and belong to the probability simplex, i.e. $\sum_{i=1}^{n^s} p_i^s = \sum_{i=1}^{n^t} p_i^t = 1$[2]. Let $\hat{\Pi}$ be the set of probabilistic couplings between the two empirical distributions defined as:

$$\hat{\Pi} = \left\{ \boldsymbol{\Gamma} \in (\mathbb{R}^+)^{n^s \times n^t} | \, \boldsymbol{\Gamma} \mathbf{1}_{n^t} = \widehat{\mathcal{D}_{\mathcal{S}}}, \boldsymbol{\Gamma}^T \mathbf{1}_{n^s} = \widehat{\mathcal{D}_{\mathcal{T}}} \right\} \tag{6.3}$$

where $\mathbf{1}_n$ is a n-dimensional vector of ones. Problem (6.2) becomes:

$$\underset{\boldsymbol{\Gamma} \in \hat{\Pi}}{\arg\min} \quad \langle \boldsymbol{\Gamma}, \mathbf{C} \rangle_{\mathcal{F}} \tag{6.4}$$

where $\langle \cdot, \cdot \rangle_{\mathcal{F}}$ is the Frobenius dot product[3] and $\mathbf{C} \geq 0$ is the $n^s \times n^t$ cost matrix related to the function c.

Barycentric mapping Once the probabilistic coupling $\boldsymbol{\Gamma}$ has been computed one needs to perform the transformation of examples from $\mathcal{X}^s$ to $\mathcal{X}^t$. This transformation can be conveniently expressed with respect to the set of examples $\mathbf{X}^t$ as the following barycentric mapping Reich (2013); Courty et al. (2014a); Ferradans et al. (2014):

$$\widehat{\mathbf{x}_i^s} = \underset{\mathbf{x}^s \in \mathcal{X}^s}{\arg\min} \quad \sum_{j=1}^{n^t} \boldsymbol{\Gamma}(i,j) c\big(\mathbf{x}^s, \mathbf{x}_j^t\big) \, . \tag{6.5}$$

where $\widehat{\mathbf{x}_i^s}$ is the image of example $\mathbf{x}_i^s$ with coupling $\boldsymbol{\Gamma}$. When the cost function is the squared euclidean distance[4], this barycentre corresponds to a weighted average and the sample is mapped into the convex hull of the target samples. For all source samples, this barycentric mapping can therefore be expressed as:

$$\widehat{\mathbf{X}^s} = f_{\boldsymbol{\Gamma}}(\mathbf{X}^s) \doteq \mathrm{diag}(\boldsymbol{\Gamma} \mathbf{1}_{n^t})^{-1} \boldsymbol{\Gamma} \mathbf{X}^t. \tag{6.6}$$

In the rest of the chapter we will focus on an uniform sampling[5] hence $\widehat{\mathbf{X}^s} = n^s \boldsymbol{\Gamma} \mathbf{X}^t$. The main drawback of the mapping ((6.6)) is that it does not allow the projection of out-of-sample examples which do not have been seen during the learning process of $\boldsymbol{\Gamma}$. It means that to transport a new example $\mathbf{x}^s \sim \mathcal{D}_{\mathcal{S}}$ one has to compute the coupling matrix $\boldsymbol{\Gamma}$ again using this new example. Also, while some authors consider specific regularization of $\boldsymbol{\Gamma}$ Cuturi (2013a); Courty et al. (2014a) to control the nature of the coupling, inducing specific properties of the transformation $f_{\mathcal{S} \to \mathcal{T}}$ (i.e. regularity, divergence free, etc.) is hard to achieve.

In the next section we present a relaxation of the optimal transport problem, which consists in jointly learning $\boldsymbol{\Gamma}$ and $f_{\mathcal{S} \to \mathcal{T}}$. We derive the corresponding optimization problem, and show its usefulness in specific scenarios.

[2] If we consider that the examples are drawn i.i.d. we have $p_i = \frac{1}{n}$ for every example.
[3] $\langle \mathbf{A}, \mathbf{B} \rangle_{\mathcal{F}} = \mathrm{Tr}(\mathbf{A}^T \mathbf{B})$
[4] $c(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\|_2^2$
[5] In other words the examples are drawn i.i.d. from $\mathcal{D}_{\mathcal{S}}$ and $\mathcal{D}_{\mathcal{T}}$.

6.3 General Framework

In this chapter we propose to solve the problem of optimal transport by jointly learning the matrix $\mathbf{\Gamma}$ and the transformation function $f_{\mathcal{S}\to\mathcal{T}}$. First of all we denote $\mathcal{H}$ the space of transformations. Let $\mathbf{X}^s$ and $\mathbf{X}^t$ be matrices where each line is an example drawn from $\mathcal{D}_\mathcal{S}$ and $\mathcal{D}_\mathcal{T}$. We propose the following optimisation problem:

$$\underset{\substack{f_{\mathcal{S}\to\mathcal{T}}\in\mathcal{H}\\ \mathbf{\Gamma}\in\hat{\Pi}}}{\arg\min} f(\mathbf{\Gamma}, f_{\mathcal{S}\to\mathcal{T}}) = \frac{1}{n^s d^t}\left\|f_{\mathcal{S}\to\mathcal{T}}(\mathbf{X}^s) - n^s\mathbf{\Gamma}\mathbf{X}^t\right\|_\mathcal{F}^2 + \frac{\lambda_\mathbf{\Gamma}}{\max(\mathbf{C})}\langle\mathbf{\Gamma},\mathbf{C}\rangle_\mathcal{F} + \frac{\lambda_{f_{\mathcal{S}\to\mathcal{T}}}}{d^s d^t}R(f_{\mathcal{S}\to\mathcal{T}}) \tag{6.7}$$

where $f_{\mathcal{S}\to\mathcal{T}}(\mathbf{X}^s)$ is a short-hand for the application of $f_{\mathcal{S}\to\mathcal{T}}$ on each example in $\mathbf{X}^s$, R is a regularization term on $f_{\mathcal{S}\to\mathcal{T}}$ and $\lambda_\mathbf{\Gamma}, \lambda_{f_{\mathcal{S}\to\mathcal{T}}}$ are hyper-parameters controlling the trade-off between the different terms in the optimization problem. The constants in front of each term normalize their values to be of the same order of magnitude. The first term in Problem (6.7) depends on both $f_{\mathcal{S}\to\mathcal{T}}$ and $\mathbf{\Gamma}$ and controls the closeness between the transformation induced by $f_{\mathcal{S}\to\mathcal{T}}$ and the barycentric interpolation obtained from $\mathbf{\Gamma}$. The second term only depends on $\mathbf{\Gamma}$ and corresponds to the standard optimal transport loss. The third term regularizes $f_{\mathcal{S}\to\mathcal{T}}$ to ensure a better generalization of the learned transformation.

In the next section we propose an efficient solution to optimize Problem (6.7) and we discuss several possible choices for the set of transformations $\mathcal{H}$, in particular we show a link with RVML (Chapter 5).

6.4 Optimisation

A standard approach to solve Problem (6.7) is to use a block-coordinate descent (Tseng, 2001) where the idea is to alternatively optimize for $f_{\mathcal{S}\to\mathcal{T}}$ and $\mathbf{\Gamma}$. In the next theorem we show that under some mild assumptions on the regularization term R and the function space $\mathcal{H}$ this problem is jointly convex. In this case we are guaranteed to converge to the optimal solution if the formulation is strictly convex with respect to $f_{\mathcal{S}\to\mathcal{T}}$ and $\mathbf{\Gamma}$ respectively. While this is not the case for $\mathbf{\Gamma}$ in our formulation, our algorithm works well in practice and a small regularization term can be added if theoretical convergence is required[6].

Theorem 6.1. *Let $\mathcal{H}$ be a convex space and R be a convex regularization. Problem* (6.7) *is jointly convex in $f_{\mathcal{S}\to\mathcal{T}}$ and $\mathbf{\Gamma}$.*

Proof. First of all recall that a sum of jointly convex functions is jointly convex. Hence it is sufficient to show that the three terms of Problem (6.7) are jointly convex. We note:

$$f_1(\mathbf{\Gamma}, f_{\mathcal{S}\to\mathcal{T}}) = \frac{1}{n^s d^t}\left\|f_{\mathcal{S}\to\mathcal{T}}(\mathbf{X}^s) - n^s\mathbf{\Gamma}\mathbf{X}^t\right\|_\mathcal{F}^2,$$

[6]For example this regularization term could be the Frobenius norm.

$$f_2(\mathbf{\Gamma}) = \frac{\lambda_{\mathbf{\Gamma}}}{\max(\mathbf{C})} \langle \mathbf{\Gamma}, \mathbf{C} \rangle_{\mathcal{F}},$$
$$f_3(f_{\mathcal{S}\to\mathcal{T}}) = \frac{\lambda_{f_{\mathcal{S}\to\mathcal{T}}}}{d^s d^t} R(f_{\mathcal{S}\to\mathcal{T}}).$$

Note that by construction f_2 and f_3 are jointly convex in $\mathbf{\Gamma}$ and $f_{\mathcal{S}\to\mathcal{T}}$. We will show that f_1 is also jointly convex. Let $g(\mathbf{\Gamma}, f_{\mathcal{S}\to\mathcal{T}}) = \left\| f_{\mathcal{S}\to\mathcal{T}}(\mathbf{X}^s) - n^s \mathbf{\Gamma} \mathbf{X}^t \right\|_{\mathcal{F}}$, we want to show that:

$$g\big(t\mathbf{\Gamma} + (t-1)\mathbf{\Gamma}', t f_{\mathcal{S}\to\mathcal{T}} + (1-t) f'_{\mathcal{S}\to\mathcal{T}}\big) \leq t g(\mathbf{\Gamma}, f_{\mathcal{S}\to\mathcal{T}}) + (1-t) g\big(\mathbf{\Gamma}', f'_{\mathcal{S}\to\mathcal{T}}\big).$$

We have:

$$\begin{aligned}
&\left\| (t f_{\mathcal{S}\to\mathcal{T}} + (1-t) f'_{\mathcal{S}\to\mathcal{T}})(\mathbf{X}^s) - n^s (t\mathbf{\Gamma} + (t-1)\mathbf{\Gamma}') \mathbf{X}^t \right\|_{\mathcal{F}} \\
&\qquad \text{(Triangle inequality and definition of } \mathcal{H}.) \\
&\leq \left\| t f_{\mathcal{S}\to\mathcal{T}}(\mathbf{X}^s) - t n^s \mathbf{\Gamma} \mathbf{X}^t \right\|_{\mathcal{F}} + \left\| (1-t) f'_{\mathcal{S}\to\mathcal{T}}(\mathbf{X}^s) - (1-t) n^s \mathbf{\Gamma}' \mathbf{X}^t \right\|_{\mathcal{F}} \\
&\qquad (t \in [0,1].) \\
&\leq t \left\| f_{\mathcal{S}\to\mathcal{T}}(\mathbf{X}^s) - n^s \mathbf{\Gamma} \mathbf{X}^t \right\|_{\mathcal{F}} + (1-t) \left\| f'_{\mathcal{S}\to\mathcal{T}}(\mathbf{X}^s) - n^s \mathbf{\Gamma}' \mathbf{X}^t \right\|_{\mathcal{F}}
\end{aligned}$$

Furthermore noting that g is convex and positive we have:

$$\begin{aligned}
&\left[g\big(t\mathbf{\Gamma} + (t-1)\mathbf{\Gamma}', t f_{\mathcal{S}\to\mathcal{T}} + (1-t) f'_{\mathcal{S}\to\mathcal{T}}\big) \right]^2 \\
&\qquad (\forall x \in \mathbb{R}^+, x \to x^2 \text{ is non decreasing.}) \\
&\leq \left[t g(\mathbf{\Gamma}, f_{\mathcal{S}\to\mathcal{T}}) + (1-t) g\big(\mathbf{\Gamma}', f'_{\mathcal{S}\to\mathcal{T}}\big) \right]^2 \\
&\qquad (\forall x \in \mathbb{R}, x \to x^2 \text{ is convex.}) \\
&\leq t \left[g(\mathbf{\Gamma}, f_{\mathcal{S}\to\mathcal{T}}) \right]^2 + (1-t) \left[g\big(\mathbf{\Gamma}', f'_{\mathcal{S}\to\mathcal{T}}\big) \right]^2.
\end{aligned}$$

Noting that $f_1(\mathbf{\Gamma}, f_{\mathcal{S}\to\mathcal{T}}) = \frac{1}{n^s d^t} g(\mathbf{\Gamma}, f_{\mathcal{S}\to\mathcal{T}})^2$ concludes the proof. □

As discussed above we propose to solve Problem (6.7) using a block-coordinates approach. As such we derive an efficient way to solve the problem for $\mathbf{\Gamma}$ when $f_{\mathcal{S}\to\mathcal{T}}$ is fixed and for $f_{\mathcal{S}\to\mathcal{T}}$ when $\mathbf{\Gamma}$ is fixed.

Solving for $\mathbf{\Gamma}$ with $f_{\mathcal{S}\to\mathcal{T}}$ fixed In this case Problem (6.7) becomes:

$$\arg\min_{\mathbf{\Gamma}\in\hat{\Pi}} f(\mathbf{\Gamma}, f_{\mathcal{S}\to\mathcal{T}}) = \frac{1}{n^s d^t} \left\| f_{\mathcal{S}\to\mathcal{T}}(\mathbf{X}^s) - n^s \mathbf{\Gamma} \mathbf{X}^t \right\|_{\mathcal{F}}^2 + \frac{\lambda_{\mathbf{\Gamma}}}{\max(\mathbf{C})} \langle \mathbf{\Gamma}, \mathbf{C} \rangle_{\mathcal{F}} + \frac{\lambda_{f_{\mathcal{S}\to\mathcal{T}}}}{d^s d^t} R(f_{\mathcal{S}\to\mathcal{T}}) \tag{6.8}$$

where $f_{\mathcal{S}\to\mathcal{T}}$ is the current transformation. To solve such an optimization problem a common approach is to use the Frank-Wolfe algorithm Ferradans et al. (2014); Frank and Wolfe (1956). It is a procedure for solving any convex constrained optimization problems with a convex and continuously differentiable objective function over a compact convex subset of any vector space. This algorithm can find an ϵ approximation of the optimal solution in $O(1/\epsilon)$ iterations Jaggi (2013). The approach is detailed in Algorithm 3.

input : The current values of $\mathbf{\Gamma}$ and $f_{\mathcal{S}\to\mathcal{T}}$.
output: The new value of $\mathbf{\Gamma}$.
begin
  Initialize $k = 0$ and $\mathbf{\Gamma}_0 = \mathbf{\Gamma}$
  repeat
    Solve $\mathbf{\Gamma}_{k+\frac{1}{2}} = \underset{\mathbf{\Gamma}_{k+\frac{1}{2}} \in \hat{\Pi}}{\arg\min} \left\langle \mathbf{\Gamma}_{k+\frac{1}{2}}, \nabla f(\mathbf{\Gamma}_k, f_{\mathcal{S}\to\mathcal{T}}) \right\rangle_{\mathcal{F}}$ with

$$\nabla f(\mathbf{\Gamma}, f_{\mathcal{S}\to\mathcal{T}}) = \frac{\lambda_{\mathbf{\Gamma}}}{\max(\mathbf{C})}\mathbf{C} - \frac{2}{d^t} f_{\mathcal{S}\to\mathcal{T}}(\mathbf{X}^s)\,\mathbf{X}^{tT} + \frac{2}{d^t} n^s \mathbf{\Gamma}\mathbf{X}^t\mathbf{X}^{tT}.$$

    Find the optimal step α_k satisfying the Armijo rule that minimizes $f\Big((1-\alpha)\mathbf{\Gamma}_k + \alpha\mathbf{\Gamma}_{k+\frac{1}{2}}, f_{\mathcal{S}\to\mathcal{T}}\Big)$.
    Update $\mathbf{\Gamma}_{k+1} = (1-\alpha_k)\mathbf{\Gamma}_k + \alpha_k\mathbf{\Gamma}_{k+\frac{1}{2}}$ and $k = k+1$.
  until *convergence*
end

Algorithm 3: Updating $\mathbf{\Gamma}$ with the Frank-Wolfe algorithm.

Solving for $f_{\mathcal{S}\to\mathcal{T}}$ with $\mathbf{\Gamma}$ fixed In this case Problem (6.7) becomes:

$$\underset{f_{\mathcal{S}\to\mathcal{T}}\in\mathcal{H}}{\arg\min}\, f(\mathbf{\Gamma}, f_{\mathcal{S}\to\mathcal{T}}) = \frac{1}{n^s d^t}\left\|f_{\mathcal{S}\to\mathcal{T}}(\mathbf{X}^s) - n^s\mathbf{\Gamma}\mathbf{X}^t\right\|_{\mathcal{F}}^2 + \frac{\lambda_{\mathbf{\Gamma}}}{\max(\mathbf{C})}\langle\mathbf{\Gamma}, \mathbf{C}\rangle_{\mathcal{F}} + \frac{\lambda_{f_{\mathcal{S}\to\mathcal{T}}}}{d^s d^t} R(f_{\mathcal{S}\to\mathcal{T}}) \tag{6.9}$$

where $\mathbf{\Gamma}$ is the current mapping between the examples. The solution to this optimization problem depends on $\mathcal{H}$ and R. This is discussed in detail in the next subsection.

6.4.1 Choosing $\mathcal{H}$

In the previous subsection we presented our method when considering a general set of functions $\mathcal{H}$. We now turn our attention toward several possibilities for the choice of $\mathcal{H}$. On the one hand we propose to define $\mathcal{H}$ as a set of linear transformations from $\mathcal{X}^s$ to $\mathcal{X}^t$. On the other hand using the kernel trick, we propose to consider non-linear transformations. Furthermore in both cases we consider the biased and non biased settings. In this case our approach boils down to learn a transformation matrix $\mathbf{L}$. It can then be seen as using RVML (Chapter 5) where the virtual points are defined thanks to the barycentric mapping associated to the current coupling.

Linear transformations A first way to define $\mathcal{H}$ is to consider linear transformations induced by a $d_s \times d^t$ real matrix $\mathbf{L}$:

$$\mathcal{H} = \left\{ f_{\mathcal{S}\to\mathcal{T}} : \exists \mathbf{L} \in \mathbb{R}^{d^s \times d^t} \text{ s.t. } \forall \mathbf{x}^s \in \mathcal{X}^s, f_{\mathcal{S}\to\mathcal{T}}(\mathbf{x}^s) = \mathbf{x}^{sT}\mathbf{L} \right\}. \tag{6.10}$$

Furthermore, we define $R(f_{\mathcal{S}\to\mathcal{T}}) = \|\mathbf{L} - \mathbf{I}\|_{\mathcal{F}}^2$ where $\mathbf{I}$ is the identity matrix. We choose to bias $\mathbf{L}$ toward $\mathbf{I}$ in order to ensure that the examples are not moved too far away from their

initial position. In this case we can rewrite optimization problem (6.7) as:

$$\underset{\mathbf{L}\in\mathbb{R}^{d^s\times d^t},\mathbf{\Gamma}\in\hat{\Pi}}{\arg\min} \frac{1}{n^s d^t}\left\|\mathbf{X}^s\mathbf{L}-n^s\mathbf{\Gamma}\mathbf{X}^t\right\|_{\mathcal{F}}^2+\frac{\lambda_{\mathbf{\Gamma}}}{\max(\mathbf{C})}\langle\mathbf{\Gamma},\mathbf{C}\rangle_{\mathcal{F}}+\frac{\lambda_{\mathbf{L}}}{d^s d^t}\left\|\mathbf{L}-\mathbf{I}\right\|_{\mathcal{F}}^2. \tag{6.11}$$

According to Algorithm 4 a part of our procedure requires to solve optimization problem (6.11) when $\mathbf{\Gamma}$ is fixed. One solution is to use the following closed form for $\mathbf{L}$:

$$\mathbf{L}=\left(\frac{1}{n^s d^t}\mathbf{X}^{sT}\mathbf{X}^s+\frac{\lambda_{\mathbf{L}}}{d^s d^t}\mathbf{I}\right)^{-1}\left(\frac{1}{n^s d^s}\mathbf{X}^{sT}n^s\mathbf{\Gamma}\mathbf{X}^t+\frac{\lambda_{\mathbf{L}}}{d^s d^t}\mathbf{I}\right) \tag{6.12}$$

where $(\cdot)^{-1}$ is the matrix inverse (Moore-Penrose pseudo-inverse when the matrix is singular). In the previous definition of $\mathcal{H}$, we considered non biased linear transformations. However it is sometimes desirable to add a bias to the transformation.

Biased linear transformations In the biased linear case $\mathcal{H}$ becomes:

$$\mathcal{H}=\left\{f_{\mathcal{S}\to\mathcal{T}}:\exists\mathbf{L}\in\mathbb{R}^{d^s\times d^t},\exists\mathbf{b}\in\mathbb{R}^{d^t}\text{ s.t. }\forall\mathbf{x}^s\in\mathcal{X}^s, f_{\mathcal{S}\to\mathcal{T}}(\mathbf{x}^s)=\mathbf{x}^{sT}\mathbf{L}+\mathbf{b}^T=\begin{pmatrix}\mathbf{x}^{sT} & 1\end{pmatrix}\begin{pmatrix}\mathbf{L}\\ \mathbf{b}^T\end{pmatrix}\right\}. \tag{6.13}$$

In this case, Problem (6.7) becomes:

$$\underset{\begin{pmatrix}\mathbf{L}\\ \mathbf{b}^T\end{pmatrix}\in\mathbb{R}^{d^s+1\times d^t},\mathbf{\Gamma}\in\hat{\Pi}}{\arg\min} \frac{1}{n^s d^t}\left\|\begin{pmatrix}\mathbf{X}^s & \mathbf{1}\end{pmatrix}\begin{pmatrix}\mathbf{L}\\ \mathbf{b}^T\end{pmatrix}-n^s\mathbf{\Gamma}\mathbf{X}^t\right\|_{\mathcal{F}}^2+\frac{\lambda_{\mathbf{\Gamma}}}{\max(\mathbf{C})}\langle\mathbf{\Gamma},\mathbf{C}\rangle_{\mathcal{F}}+\frac{\lambda_{\mathbf{L}}}{d^s d^t}\left\|\mathbf{L}-\mathbf{I}\right\|_{\mathcal{F}}^2. \tag{6.14}$$

As in the non biased case, it is possible to find a closed form solution for $\begin{pmatrix}\mathbf{L}\\ \mathbf{b}^T\end{pmatrix}$ when $\mathbf{\Gamma}$ is fixed:

$$\begin{pmatrix}\mathbf{L}\\ \mathbf{b}^T\end{pmatrix}=\left(\frac{1}{n^s d^t}\begin{pmatrix}\mathbf{X}^s\\ \mathbf{1}^T\end{pmatrix}\begin{pmatrix}\mathbf{X}^s & \mathbf{1}\end{pmatrix}+\frac{\lambda_{\mathbf{L}}}{d^s d^t}\begin{pmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{0}^T & 0\end{pmatrix}\right)^{-1}\left(\frac{1}{n^s d^t}\begin{pmatrix}\mathbf{X}^s\\ \mathbf{1}^T\end{pmatrix}n^s\mathbf{\Gamma}\mathbf{X}^t+\frac{\lambda_{\mathbf{L}}}{d^s d^t}\begin{pmatrix}\mathbf{I}\\ \mathbf{0}^T\end{pmatrix}\right) \tag{6.15}$$

Non-linear transformations In some cases a linear transformation is not sufficient to approximate the optimal transport. Hence, we propose to consider non-linear transformations. To do this, let ϕ be a non-linear function associated to a kernel function $k:\mathcal{X}^s\times\mathcal{X}^s\to\mathbb{R}$ such that $k(\mathbf{x}^s,\mathbf{x}^{s\prime})=\langle\phi(\mathbf{x}^s),\phi(\mathbf{x}^{s\prime})\rangle_{\mathcal{H}}$. We can then define $\mathcal{H}$ for a given set of examples $\mathbf{X}^s$ as:

$$\mathcal{H}=\left\{f_{\mathcal{S}\to\mathcal{T}}:\exists\mathbf{L}\in\mathbb{R}^{n^s\times d^t}\text{ s.t. }\forall\mathbf{x}^s\in\mathcal{X}^s, f_{\mathcal{S}\to\mathcal{T}}(\mathbf{x}^s)=k_{\mathbf{X}^s}\left(\mathbf{x}^{sT}\right)\mathbf{L}\right\} \tag{6.16}$$

where $k_{\mathbf{X}^s}\left(\mathbf{x}^{sT}\right)$ is a short-hand for the vector $\begin{pmatrix} k(\mathbf{x}^s, \mathbf{x}_1^s) & k(\mathbf{x}, \mathbf{x}_2^s) & \cdots & k(\mathbf{x}^s, \mathbf{x}_{n^s}^s) \end{pmatrix}$ where $\mathbf{x}_1^s, \cdots, \mathbf{x}_{n^s}^s$ are the rows of $\mathbf{X}^s$. In this case optimization problem (6.7) becomes:

$$\underset{\mathbf{L}\in\mathbb{R}^{n^s\times d^t},\boldsymbol{\Gamma}\in\hat{\Pi}}{\arg\min} \frac{1}{n^s d^t}\left\|k_{\mathbf{X}^s}(\mathbf{X}^s)\,\mathbf{L} - n^s\boldsymbol{\Gamma}\mathbf{X}^t\right\|_{\mathcal{F}}^2 + \frac{\lambda_{\boldsymbol{\Gamma}}}{\max(\mathbf{C})}\left\langle\boldsymbol{\Gamma},\mathbf{C}\right\rangle_{\mathcal{F}} + \frac{\lambda_{\mathbf{L}}}{n^s d^t}\left\|k_{\mathbf{X}^s}(\cdot)\,\mathbf{L}\right\|_{\mathcal{F}}^2. \quad (6.17)$$

where $k_{\mathbf{X}^s}(\cdot)$ is a short-hand for the vector $\begin{pmatrix} k(\cdot, \mathbf{x}_1^s) & \cdots & k(\cdot, \mathbf{x}_{n^s}^s) \end{pmatrix} = \begin{pmatrix} \phi(\mathbf{x}_1^s) & \cdots & \phi(\mathbf{x}_{n^s}^s) \end{pmatrix}$. As in the linear case there is a closed form solution for $\mathbf{L}$ when $\boldsymbol{\Gamma}$ is fixed:

$$\mathbf{L} = \left(\frac{1}{n^s d^t}k_{\mathbf{X}^s}(\mathbf{X}^s) + \frac{\lambda_{\mathbf{L}}}{n^s d^t}\mathbf{I}\right)^{-1}\frac{1}{n^s d^t}n^s\boldsymbol{\Gamma}\mathbf{X}^t. \quad (6.18)$$

As in the linear case it might be interesting to use a bias.

Biased non-linear transformations In the biased non-linear case $\mathcal{H}$ becomes:

$$\mathcal{H} = \left\{ f_{\mathcal{S}\to\mathcal{T}} : \exists\mathbf{L}\in\mathbb{R}^{n^s\times d^t}, \exists\mathbf{b}\in\mathbb{R}^{d^t} \text{ s.t. } \forall\mathbf{x}^s\in\mathcal{X}^s, f_{\mathcal{S}\to\mathcal{T}}(\mathbf{x}^s) = \begin{pmatrix} k_{\mathbf{X}^s}\left(\mathbf{x}^{sT}\right) & \mathbf{1} \end{pmatrix}\begin{pmatrix} \mathbf{L} \\ \mathbf{b}^T \end{pmatrix}\right\} \quad (6.19)$$

Optimization problem (6.7) can be rewritten as:

$$\underset{\begin{pmatrix}\mathbf{L}\\ \mathbf{b}^T\end{pmatrix}\in\mathbb{R}^{n^s+1\times d^t},\boldsymbol{\Gamma}\in\hat{\Pi}}{\arg\min} \frac{1}{n^s d^t}\left\|\begin{pmatrix} k_{\mathbf{X}^s}(\mathbf{X}^s) & \mathbf{1}\end{pmatrix}\begin{pmatrix}\mathbf{L}\\ \mathbf{b}^T\end{pmatrix} - n^s\boldsymbol{\Gamma}\mathbf{X}^t\right\|_{\mathcal{F}}^2 + \frac{\lambda_{\boldsymbol{\Gamma}}}{\max(\mathbf{C})}\left\langle\boldsymbol{\Gamma},\mathbf{C}\right\rangle_{\mathcal{F}} + \frac{\lambda_{\mathbf{L}}}{n^s d^t}\left\|k_{\mathbf{X}^s}(\cdot)\,\mathbf{L}\right\|_{\mathcal{F}}^2. \quad (6.20)$$

As in the non biased case, it is possible to find a closed form solution for $\begin{pmatrix}\mathbf{L}\\ \mathbf{b}^T\end{pmatrix}$ when $\boldsymbol{\Gamma}$ is fixed:

$$\begin{pmatrix}\mathbf{L}\\ \mathbf{b}^T\end{pmatrix} = \left(\frac{1}{n^s d^t}\begin{pmatrix}\mathbf{K}_{\mathbf{X}^s\mathbf{X}^s}\\ \mathbf{1}^T\end{pmatrix}\begin{pmatrix}\mathbf{K}_{\mathbf{X}^s\mathbf{X}^s} & \mathbf{1}\end{pmatrix} + \frac{\lambda_{\mathbf{L}}}{d^s d^t}\begin{pmatrix}\mathbf{K}_{\mathbf{X}^s\mathbf{X}^s} & \mathbf{0}\\ \mathbf{0}^T & 0\end{pmatrix}\right)^{-1}\frac{1}{n^s d^t}\begin{pmatrix}\mathbf{K}_{\mathbf{X}^s\mathbf{X}^s}\\ \mathbf{1}^T\end{pmatrix}n^s\boldsymbol{\Gamma}\mathbf{X}^t \quad (6.21)$$

where $\mathbf{K}_{\mathbf{X}^s\mathbf{X}^s} = k_{\mathbf{X}^s}(\mathbf{X}^s)$.

A summary of our approach can be found in Algorithm 4.

6.5 Discussion on Theoretical Aspects

In this section we propose to discuss some theoretical considerations about our framework and more precisely on the quality of the learned transformation denoted by $f_{\mathbf{L}}$ to show its dependence on the matrix $\mathbf{L}$. To assess this quality we consider the Frobenius norm between $f_{\mathbf{L}}$ and the true transport map, denoted $f_{\mathcal{X}^s\to\mathcal{X}^t}$, that we would obtain if we could solve the

input : $\mathbf{X}^s, \mathbf{X}^t$ source and target examples and $\lambda_{\mathbf{\Gamma}}, \lambda_{\mathbf{L}}$ hyper parameters.
output: $\mathbf{L}, \mathbf{\Gamma}$.
begin
  Initialize $k = 0$, $\mathbf{\Gamma}_0 \in \hat{\Pi}$ and $\mathbf{L}_0 = \mathbf{I}$
  repeat
    Learn $\mathbf{\Gamma}_{k+1}$ with fixed $\mathbf{L}_k$ using a Frank-Wolfe approach (Algorithm 3).
    Learn $\mathbf{L}_{k+1}$ using Equations (6.12), (6.15), (6.18) or (6.21) with fixed $\mathbf{\Gamma}_{k+1}$.
    Set $k = k + 1$.
  until *convergence*
end

Algorithm 4: Joint Learning of $\mathbf{L}$ and $\mathbf{\Gamma}$.

Monge problem. Let $f_{\mathbf{\Gamma}}$ be the empirical barycentric mapping using the probabilistic coupling $\mathbf{\Gamma}$ learned between $\mathbf{X}^s$ and $\mathbf{X}^t$. Similarly let $f_{\mathcal{X}^s \times \mathcal{X}^t}$ be the theoretical barycentric mapping associated with the probabilistic coupling learned on $\mathcal{D}_{\mathcal{S}}, \mathcal{D}_{\mathcal{T}}$ the whole distributions and which corresponds to the solution of Kantorovich's problem. Using a slight abuse of notations we denote by $f_{\mathbf{\Gamma}}(\mathbf{x}^s)$ and $f_{\mathcal{X}^s \times \mathcal{X}^t}(\mathbf{x}^s)$ the projection of $\mathbf{x}^s \in \mathbf{X}^s$ by these barycentric mappings. We have the following simple theorem on the quality of the learned transformation.

Theorem 6.2 (Bound on the quality of the learned transformation)**.** *With high probability we have:*

$$\begin{aligned}\mathop{\mathbb{E}}_{\mathbf{x}^s \sim \mathcal{D}_{\mathcal{S}}} \|f_{\mathbf{L}}(\mathbf{x}^s) - f_{\mathcal{X}^s \to \mathcal{X}^t}(\mathbf{x}^s)\|_{\mathcal{F}}^2 \leq 4 \sum_{\mathbf{x}^s \in \mathbf{X}^s} \|f_{\mathbf{L}}(\mathbf{x}^s) - f_{\mathbf{\Gamma}}(\mathbf{x}^s)\|_{\mathcal{F}}^2 + \mathcal{O}\left(\frac{1}{\sqrt{n_s}}\right) \\ + 4 \sum_{\mathbf{x}^s \in \mathbf{X}^s} \|f_{\mathbf{\Gamma}}(\mathbf{x}^s) - f_{\mathcal{X}^s \times \mathcal{X}^t}(\mathbf{x}^s)\|_{\mathcal{F}}^2 \\ + 2 \mathop{\mathbb{E}}_{\mathbf{x}^s \sim \mathcal{D}_{\mathcal{S}}} \|f_{\mathcal{X}^s \times \mathcal{X}^t}(\mathbf{x}^s) - f_{\mathcal{X}^s \to \mathcal{X}^t}(\mathbf{x}^s)\|_{\mathcal{F}}^2 \qquad (6.22)\end{aligned}$$

Proof.

$$\begin{aligned}&\mathop{\mathbb{E}}_{\mathbf{x}^s \sim \mathcal{D}_{\mathcal{S}}} \|f_{\mathbf{L}}(\mathbf{x}^s) - f_{\mathcal{X}^s \to \mathcal{X}^t}(\mathbf{x}^s)\|_{\mathcal{F}}^2 \\ &\qquad \text{(Triangle inequality.)} \\ &\leq \mathop{\mathbb{E}}_{\mathbf{x}^s \sim \mathcal{D}_{\mathcal{S}}} \left(\|f_{\mathbf{L}}(\mathbf{x}^s) - f_{\mathcal{X}^s \times \mathcal{X}^t}(\mathbf{x}^s)\|_{\mathcal{F}} + \|f_{\mathcal{X}^s \times \mathcal{X}^t}(\mathbf{x}^s) - f_{\mathcal{X}^s \to \mathcal{X}^t}(\mathbf{x}^s)\|_{\mathcal{F}}\right)^2 \\ &\qquad ((a+b)^2 \leq 2a^2 + 2b^2.) \\ &\leq 2 \mathop{\mathbb{E}}_{\mathbf{x}^s \sim \mathcal{D}_{\mathcal{S}}} \|f_{\mathbf{L}}(\mathbf{x}^s) - f_{\mathcal{X}^s \times \mathcal{X}^t}(\mathbf{x}^s)\|_{\mathcal{F}}^2 + 2 \mathop{\mathbb{E}}_{\mathbf{x}^s \sim \mathcal{D}_{\mathcal{S}}} \|f_{\mathcal{X}^s \times \mathcal{X}^t}(\mathbf{x}^s) - f_{\mathcal{X}^s \to \mathcal{X}^t}(\mathbf{x}^s)\|_{\mathcal{F}}^2\end{aligned}$$

Furthermore considering that $\mathcal{H}$ is as proposed in Section 6.4 and using Theorem 5.2 in Chapter 5 we have with high probability that:

$$\begin{aligned}\mathop{\mathbb{E}}_{\mathbf{x}^s \sim \mathcal{D}_{\mathcal{S}}} \|f_{\mathbf{L}}(\mathbf{x}^s) - f_{\mathcal{X}^s \to \mathcal{X}^t}(\mathbf{x}^s)\|_{\mathcal{F}}^2 \leq 2 \sum_{\mathbf{x}^s \in \mathbf{X}^s} \|f_{\mathbf{L}}(\mathbf{x}^s) - f_{\mathcal{X}^s \times \mathcal{X}^t}(\mathbf{x}^s)\|_{\mathcal{F}}^2 + \mathcal{O}\left(\frac{1}{\sqrt{n_s}}\right) \\ + 2 \mathop{\mathbb{E}}_{\mathbf{x}^s \sim \mathcal{D}_{\mathcal{S}}} \|f_{\mathcal{X}^s \times \mathcal{X}^t}(\mathbf{x}^s) - f_{\mathcal{X}^s \to \mathcal{X}^t}(\mathbf{x}^s)\|_{\mathcal{F}}^2 .\end{aligned}$$

$$
\begin{aligned}
&\text{(Triangle inequality.)}\\
&\leq 2 \sum_{\mathbf{x}^s \in \mathbf{X}^s} \left(\|f_{\mathbf{L}}(\mathbf{x}^s) - f_{\mathbf{\Gamma}}(\mathbf{x}^s)\|_{\mathcal{F}} + \|f_{\mathbf{\Gamma}}(\mathbf{x}^s) - f_{\mathcal{X}^s \times \mathcal{X}^t}(\mathbf{x}^s)\|_{\mathcal{F}} \right)^2 \\
&\quad + \mathcal{O}\left(\frac{1}{\sqrt{n_s}}\right) + 2 \mathop{\mathbb{E}}_{\mathbf{x}^s \sim \mathcal{D}_{\mathcal{S}}} \|f_{\mathcal{X}^s \times \mathcal{X}^t}(\mathbf{x}^s) - f_{\mathcal{X}^s \to \mathcal{X}^t}(\mathbf{x}^s)\|_{\mathcal{F}}^2 . \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad ((a+b)^2 \leq 2a^2 + 2b^2.)\\
&\leq 4 \sum_{\mathbf{x}^s \in \mathbf{X}^s} \|f_{\mathbf{L}}(\mathbf{x}^s) - f_{\mathbf{\Gamma}}(\mathbf{x}^s)\|_{\mathcal{F}}^2 + \mathcal{O}\left(\frac{1}{\sqrt{n_s}}\right) \\
&\quad + 4 \sum_{\mathbf{x}^s \in \mathbf{X}^s} \|f_{\mathbf{\Gamma}}(\mathbf{x}^s) - f_{\mathcal{X}^s \times \mathcal{X}^t}(\mathbf{x}^s)\|_{\mathcal{F}}^2 \\
&\quad + 2 \mathop{\mathbb{E}}_{\mathbf{x}^s \sim \mathcal{D}_{\mathcal{S}}} \|f_{\mathcal{X}^s \times \mathcal{X}^t}(\mathbf{x}^s) - f_{\mathcal{X}^s \to \mathcal{X}^t}(\mathbf{x}^s)\|_{\mathcal{F}}^2 \qquad (6.23)
\end{aligned}
$$

□

From Inequality (6.22) we deduce that there are three key quantities related to the quality of the learned transformation $f_{\mathbf{L}}$:

- $\sum_{\mathbf{x}^s \in \mathbf{X}^s} \|f_{\mathbf{L}}(\mathbf{x}^s) - f_{\mathbf{\Gamma}}(\mathbf{x}^s)\|_{\mathcal{F}}^2 + \mathcal{O}\left(\frac{1}{\sqrt{n_s}}\right)$: This first quantity is the difference between the transformation and the empirical barycentric mapping with respect to the Frobenius norm. It is the one that we minimize in Problem (6.7) and should be as small as possible to obtain a better approximation $f_{\mathbf{L}}$. Furthermore, by definition, the coupling used to compute the empirical barycentric mapping of this term also appears in Problem (6.7).

- $\sum_{\mathbf{x}^s \in \mathbf{X}^s} \|f_{\mathbf{\Gamma}}(\mathbf{x}^s) - f_{\mathcal{X}^s \times \mathcal{X}^t}(\mathbf{x}^s)\|_{\mathcal{F}}^2$: This second quantity is the difference between the learned barycentric mapping and the theoretical one which could be obtained by learning on the whole distribution. We expect this quantity to decrease uniformly with respect to the number of examples as it corresponds to a measure of how well a mapping learned on a limited set reflects the true mapping.

- $\mathbb{E}_{\mathbf{x}^s \sim \mathcal{D}_{\mathcal{S}}} \|f_{\mathcal{X}^s \times \mathcal{X}^t}(\mathbf{x}^s) - f_{\mathcal{X}^s \to \mathcal{X}^t}(\mathbf{x}^s)\|_{\mathcal{F}}^2$: This third quantity is the difference between the theoretical barycentric mapping and the true transformation. We expect this quantity to be small as it characterizes that a barycentric mapping using a coupling learned on the whole distributions is a good approximation of the true transport map.

Note that we only expect the second and third term to be small but we do not prove it. Indeed these quantities are difficult to bound because of a lack of theoretical results related to these in the literature. Nevertheless we think that this discussion opens the door for new theoretical perspectives to use OT in a Machine Learning setting but these are beyond the scope of this thesis.

6.6 Experiments

In this section we propose to experimentally validate our approach on two tasks. The first one is an unsupervised domain adaptation one while the second one deals with the problem

of seamless copy in images.

6.6.1 Application in Unsupervised Domain Adaptation

In this first experiment we show the interest of our approach in an unsupervised domain adaptation task.

Datasets We consider two domain adaptation datasets namely the Moons dataset Bruzzone and Marconcini (2010) and the Office-Caltech dataset Gong et al. (2012). The Moons dataset is a binary classification task which consists of 2 domains. The source domain corresponds to two intertwined moons, each one representing one class. The target domain is built by rotating the source domain. The rotation angle ranges from 10 to 90 degrees leading to 9 different adaptation tasks of increasing difficulty. In this dataset the examples are of dimension 2 and we consider 300 source examples and 300 target examples for training and 1000 target examples for testing. The Office-Caltech dataset is a 10 class image classification task which consists of 4 domains. These domains are amazom (A), dslr (D), webcam (W) and Caltech10 (C) and corresponds to images coming from different sources. Following this, there are 12 adaptation tasks where each domain is in turn considered as the source or the target (denoted source $\rightarrow$ target in the results). During the training process we consider all the examples from the source domain and half of the examples from the target domain, the other half being used as the test set. To represent the images we use deep learning features of size 4096 named decaf6 Donahue et al. (2014). Note that we also used this dataset in Chapter 4 but with the original SIFT features (Gong et al., 2012).

Methods We consider 6 different baselines. The first one is a simple 1-Nearest-Neighbour (1-NN) using the original source examples only. The second and third ones are two widely used domain adaptation approaches, namely Geodesic Flow Kernel (GFK) Gong et al. (2012) and Subspace Alignment (SA) Fernando et al. (2013). The fourth to sixth baselines are OT based approaches: The classic OT method (OT), OT with an entropy based regularization (OTE) Cuturi (2013a) and OT with a $\ell_{1,2}$ regularization (L1L2) Courty et al. (2014a). We present the results of our approach with the linear (OTLin) and kernel (OTKer) versions of the transformation. We also consider their biased counterparts (*B). For all the baselines the idea is to apply the learned transformation on the source and then to use a 1-NN classifier on the labelled source examples to classify the target examples.

Experimental setup We consider the following experimental setup for all the methods and datasets. All the results presented in this section are averaged over 10 trials. For each trial we consider three sets of examples, a labelled source training set denoted $\mathbf{X}^s, \mathbf{y}^s$, an unlabelled target training set denoted $\mathbf{X}^{t^{\text{train}}}$ and a labelled target testing set $\mathbf{X}^{t^{\text{test}}}, \mathbf{y}^{t^{\text{test}}}$. The model is learned on $\mathbf{X}^s, \mathbf{y}^s$ and $\mathbf{X}^{t^{\text{train}}}$ and evaluated on $\mathbf{X}^{t^{\text{test}}}, \mathbf{y}^{t^{\text{test}}}$ with a 1-NN learned on $\mathbf{X}^s, \mathbf{y}^s$. All the hyper-parameters are tuned according to a grid search from the source and

target training instances from a reverse validation procedure close to Bruzzone and Marconcini (2010); Zhong et al. (2010) and presented in Algorithm 5. We use this approach to affect a score to all the possible instantiations of hyper parameters and we then select the best among these. As a score we compute an average accuracy of a two fold method. The idea is to split the source training set in two halves. From one half we learn a model f that is used to label the target training set. These new labels are then used to label the second half of the source training set to obtain a first accuracy. The role of the two halves are then reversed to obtain a second accuracy. The model f is learned with an algorithm $\mathcal{A}_{\boldsymbol{\lambda}}$ using some hyper parameters λ and is able to bring closer the source and the target examples. For example, with our linear mapping learned from our regularized OT formulation, we have $f\big(\mathbf{X}^t\big) = \mathbf{X}^t$ and $f(\mathbf{X}^s) = f_{\mathbf{L}}(\mathbf{X}^s) = \mathbf{X}^s\mathbf{L}$. For the hyper parameters of the compared methods we use the following ranges: For GFK and SA we choose the dimension of the subspace as $d \in \{3, 6, \dots, 30\}$, for L1L2 and OTE we set the parameter for entropy regularization in $\{10^{-6}, 10^{-5}, \dots, 10^{5}\}$, for L1L2 we choose the class related parameter $\eta \in \{10^{-5}, 10^{-4}, \cdots, 10^{2}\}$, for all our methods we choose $\lambda_{\mathbf{L}}, \lambda_{\mathbf{\Gamma}} \in \{10^{-3}, 10^{-2}, \dots, 10^{0}\}$.

In this algorithm f is any model able to bring closer the source and the target. For example, with our linear mapping learned from our regularized OT formulation, we have $f\big(\mathbf{X}^t\big) = \mathbf{X}^t$ and $f(\mathbf{X}^s) = f_{\mathbf{L}}(\mathbf{X}^s) = \mathbf{X}^s\mathbf{L}$.

input : $(\mathbf{X}^s, \mathbf{y}^s)$ source examples and their labels, $\mathbf{X}^t$ target examples, $\mathcal{A}_{\boldsymbol{\lambda}}$ a learning procedure using hyper-parameters $\boldsymbol{\lambda}$.
output: Average accuracy of $\mathcal{A}_{\boldsymbol{\lambda}}$.
begin
Split $(\mathbf{X}^s, \mathbf{y}^s)$ in two halves $(\mathbf{X}^{s1}, \mathbf{y}^{s1})$ and $(\mathbf{X}^{s2}, \mathbf{y}^{s2})$.
Learn $f^1 = \mathcal{A}_{\boldsymbol{\lambda}}(\mathbf{X}^{s1}, \mathbf{y}^{s1}, \mathbf{X}^t)$ and set $\mathbf{y}^{t1}$ the pseudo-labels of $f^1(\mathbf{X}^t)$ obtained from a 1NN learned on $\left(f^1\big(\mathbf{X}^{s1}\big), \mathbf{y}^{s1}\right)$.
Set s^1 the accuracy of a 1NN learned on $\left(f^1(\mathbf{X}^t), \mathbf{y}^{t1}\right)$ and evaluated on $\left(f^1\big(\mathbf{X}^{s2}\big), \mathbf{y}^{s2}\right)$.
Learn $f^2 = \mathcal{A}_{\boldsymbol{\lambda}}(\mathbf{X}^{s2}, \mathbf{y}^{s2}, \mathbf{X}^t)$ and set $\mathbf{y}^{t2}$ the pseudo-labels of $f^2(\mathbf{X}^t)$ obtained from a 1NN learned on $\left(f^2\big(\mathbf{X}^{s2}\big), \mathbf{y}^{s2}\right)$.
Set s^2 the accuracy of a 1NN learned on $\left(f^2(\mathbf{X}^t), \mathbf{y}^{t2}\right)$ and evaluated on $\left(f^2\big(\mathbf{X}^{s1}\big), \mathbf{y}^{s1}\right)$.
return $\frac{s^1+s^2}{2}$.
end

Algorithm 5: Circular validation.

The results on the Moons dataset are presented in Table 6.1 and those for Office-Caltech are given in Table 6.2. A first important remark is that for both datasets the results obtained by using the barycentric mapping with $f_{\mathbf{\Gamma}}$ and the results obtained by directly using the transformation $f_{\mathbf{L}}$ are almost the same. It shows that our method allows us to learn a function $f_{\mathbf{L}}$ that is a good approximation of $f_{\mathbf{\Gamma}}$ and that $f_{\mathbf{\Gamma}}$ is well adapted to the class of transformations $\mathcal{H}$. In terms of accuracy, our approach tends to give the best results in most of the cases which shows that we are effectively able to move closer the distributions in a relevant way. For the Moons datasets, the last four approaches (including ours) based on OT

Table 6.1: Accuracy on the Moons dataset. The best result for each angle is highlighted with a bold font.

Angle	1-NN	GFK	SA	OT	L1L2	OTE	OTLin		OTLinB		OTKer		OTKerB	
							$f_{\mathbf{L}}$	$f_{\mathbf{\Gamma}}$	$f_{\mathbf{L}}$	$f_{\mathbf{\Gamma}}$	$f_{\mathbf{L}}$	$f_{\mathbf{\Gamma}}$	$f_{\mathbf{L}}$	$f_{\mathbf{\Gamma}}$
10	99.99	99.86	99.99	97.88	99.56	99.95	**100.**	**100.**	**100.**	**100.**	**100.**	**100.**	**100.**	**100.**
20	93.08	95.75	93.08	94.96	98.74	**100.**	**100.**	99.96	**100.**	99.97	**100.**	**100.**	**100.**	**100.**
30	83.98	92.55	83.98	90.62	98.36	**100.**	99.82	99.9	99.78	99.86	99.99	**100.**	99.99	**100.**
40	77.07	90.85	74.41	83.73	95.8	**99.98**	98.32	98.65	98.1	98.46	99.65	99.73	99.63	99.74
50	61.73	90.22	73.13	77.75	87.69	87.29	97.8	97.56	97.48	97.5	99.12	**99.23**	99.11	99.14
60	41.21	79.37	72.35	71.0	88.3	86.35	96.42	**97.22**	95.84	97.04	96.59	96.8	96.62	96.81
70	23.08	61.05	72.27	64.48	89.03	77.46	88.04	**94.66**	88.21	94.32	80.77	81.54	82.45	83.06
80	20.72	36.16	72.31	57.34	73.6	58.79	76.91	**81.01**	76.58	80.74	73.96	74.13	73.94	74.24
90	19.4	43.08	34.16	50.97	58.1	51.31	67.88	67.96	67.13	**68.06**	56.32	55.77	57.57	55.42

Table 6.2: Accuracy on the Office-Caltech dataset. The best result for each task is highlighted with a bold font.

Task	1-NN	GFK	SA	OT	L1L2	OTE	OTLin		OTLinB		OTKer		OTKerB	
							$f_{\mathbf{L}}$	$f_{\mathbf{\Gamma}}$	$f_{\mathbf{L}}$	$f_{\mathbf{\Gamma}}$	$f_{\mathbf{L}}$	$f_{\mathbf{\Gamma}}$	$f_{\mathbf{L}}$	$f_{\mathbf{\Gamma}}$
$D \rightarrow W$	89.47	93.31	95.56	76.95	95.7	95.7	97.28	97.28	97.28	97.28	98.41	**98.48**	**98.48**	**98.48**
$D \rightarrow A$	62.52	77.23	88.5	70.83	74.9	74.85	85.73	85.73	85.75	85.75	**89.92**	89.9	89.54	89.54
$D \rightarrow C$	51.81	69.73	**78.99**	68.09	67.85	68.03	77.15	77.15	77.43	77.43	69.1	69.17	69.27	69.31
$W \rightarrow D$	99.25	**99.75**	99.63	74.13	94.38	94.38	99.38	99.38	**99.75**	**99.75**	97.25	97.25	96.88	96.88
$W \rightarrow A$	62.5	72.38	79.25	67.6	71.33	71.35	**81.46**	**81.46**	81.38	81.38	78.5	78.35	78.52	78.81
$W \rightarrow C$	59.5	63.74	55.02	63.1	67.78	67.78	**75.87**	**75.87**	75.41	75.41	72.71	72.7	65.12	63.26
$A \rightarrow D$	65.25	75.88	**83.75**	64.63	70.13	70.5	80.63	80.63	80.38	80.5	65.63	65.5	71.88	71.5
$A \rightarrow W$	56.75	68.01	74.57	66.82	67.15	67.28	**74.64**	**74.64**	74.37	74.37	66.36	64.77	70.	68.87
$A \rightarrow C$	70.09	75.71	79.2	70.43	74.06	74.31	81.81	81.81	81.6	81.63	84.38	84.43	**84.49**	84.47
$C \rightarrow D$	75.88	79.5	85.	66.	69.75	70.25	87.13	87.13	**87.25**	**87.25**	70.13	70.	78.63	78.63
$C \rightarrow W$	65.17	70.66	74.44	59.21	63.77	63.77	78.28	78.28	78.48	78.48	80.	**80.4**	73.51	73.38
$C \rightarrow A$	85.79	87.13	89.33	75.25	76.63	76.67	**89.94**	**89.94**	89.71	89.71	82.38	82.15	83.56	83.48
Mean	70.33	77.75	81.94	68.59	74.45	74.57	**84.11**	**84.11**	84.07	84.08	79.56	79.43	79.99	79.72

obtain similar results until 40 degrees while other DA methods fail to obtain good results at 20 degrees. Beyond 50 degrees, our approach tends to obtain significantly better results (more than 10 points of accuracy) and is more stable when the difficulty of the problem increases. For Office-Caltech, our results are significantly better than other approaches which clearly illustrates the potential of our method for difficult DA tasks. As a conclusion, forcing the OT to learn a smoother map $f_{\mathbf{L}}$ allows the approach to get a better robustness.

6.6.2 Seamless Copy in Images with Gradient Adaptation

We propose here a direct application of our mapping estimation in the context of image editing. While several papers using optimal transport are focusing on color adaptation Ferradans et al. (2014); Solomon et al. (2015), we explore here a new variant in the domain of image editing: the seamless editing or cloning in images. In this context, one may desire to import a region from a given source image to a target image. As a direct copy of the region leads to inaccurate results in the final image nearby the boundaries of the copied selection, a very popular method, proposed by Pérez and co-workers Pérez et al. (2003), allows to seamlessly

blend the target image and the selection. This technique, coined as *Poisson Image Editing*, operates in the gradient domain of the image. Hence, the gradients of the selection operate as a guidance field for an image reconstruction based on membrane interpolation with appropriate boundary conditions extracted from the target image.

Let f be an unknown scalar function (usually a component of the color space of the image) defined on a given region of the image Ω. Let f^t be the target image defined everywhere apart from the interior of Ω. The Poisson editing method operates by solving for f as the following variational optimization problem with Dirichlet boundary conditions:

$$\min_f \int\int_\Omega |\nabla f - \mathbf{v}|^2 \quad \text{with} \quad f|_{\partial\Omega} = f^t|_{\partial\Omega}. \tag{6.24}$$

Here, $\mathbf{v}$ is the guidance field, which is usually given as the gradient from the source image f^s over the domain Ω, i.e. $\mathbf{v} = \nabla f^s|_\Omega$. One can show that the unique solution to this problem is the solution of the following Poisson equation Pérez et al. (2003):

$$\Delta f = \text{div } \mathbf{v} \quad \text{over } \Omega, \quad \text{with} \quad f|_{\partial\Omega} = f^t|_{\partial\Omega}. \tag{6.25}$$

Using appropriate first order discretization of the Laplacian operator, solving for this problem amounts to solve a big sparse linear system, which can be performed efficiently with multigrid solvers.

Though appealing, this technique is prone to errors due to local contrast change or false colors resulting from the integration. While some solutions combining both gradient and color domains exist Deng et al. (2012), this editing technique usually requires the source and target images to have similar colors and contrast. Here, we propose to enhance the generality of this technique by forcing the gradient distribution from the source image to follow the gradient distribution in the target image. As a result, the seamless cloning not only blends smoothly the copied region in the target domain, but also constraints the color dynamics to that of the target image. Hence, a part of the style of the target image is preserved. We start by learning a transfer function $f_{\mathbf{L}} : \mathbb{R}^6 \to \mathbb{R}^6$ with our method, where 6 denotes the vertical and horizontal components of gradient per color. Following our method which aligns the distribution of gradients in the source image to the target one, we then solve for the following system:

$$\Delta f = \text{div } f_{\mathbf{L}}(\mathbf{v}) \quad \text{over } \Omega, \quad \text{with} \quad f|_{\partial\Omega} = f^t|_{\partial\Omega}. \tag{6.26}$$

When dealing with images, the number of source and target gradients are largely exceeding tens of thousands and it is mandatory to consider methods that scale appropriately. As such, our technique can readily learn the transfer function $f_{\mathbf{L}}$ over a limited set of gradients and generalizes appropriately to unseen gradients. Several illustrations of this method are proposed in a context of face swapping in Figure 6.2. As one can observe, the original method of Poisson image editing Pérez et al. (2003) (3rd column) tends to preserve the color dynamic of the original image and fails in copying the style of the target image. Our method was tested with a linear and kernel version of $f_{\mathbf{L}}$, that was learned with only 500 gradients

sampled randomly from both sources ($\lambda_{\mathbf{L}} = 10^{-2}$, $\lambda_{\mathbf{L}} = 10^{3}$ for respectively the linear and kernel versions, and $\lambda_{\mathbf{\Gamma}} = 10^{-7}$ for both cases). As a general qualitative comment, one can observe that the kernel version of $f_{\mathbf{L}}$ is better at preserving the dynamics of the gradient, while the linear version tends to flatten the colors. In this low-dimensional space, this illustrates the need of a non-linear transform. We also illustrate one case of failure of our approach in Figure 6.3 where it is not possible to produce the same vast swaths of colors as in the target image since our method does not modify the spatial arrangement of the gradient. Regarding the computational time, the gradient adaptation is of the same order of magnitude as the Poisson equation solving, and each example are computed in less than 30s on a standard personal laptop.

6.7 Conclusion

In this chapter we proposed a solution to learn a transformation from a Mahalanobis distance whose behaviour is controlled by the geometric transformation induced by a transport map. We considered a jointly convex approach to learn both the coupling $\mathbf{\Gamma}$ and the transformation $f_{\mathbf{L}}$. From an optimal transport point of view this transformation can be seen as an approximation of the transport map given by $\mathbf{\Gamma}$ and allows us to project out-of-samples examples not seen during the learning process. Furthermore, jointly learning the coupling and the transformation allows us to regularize the transport by enforcing a certain smoothness on the transport map. We presented some theoretical considerations on the generalization ability of the learned transformation $f_{\mathbf{L}}$. Hence we discussed that under the assumption that the barycentric mapping generalizes well and is a good estimate of the true transformation, then $f_{\mathbf{L}}$ learned with our method should be a good approximation of the true transformation. We have shown that our approach is efficient in practice on two different tasks, namely domain adaptation and image editing. On the one hand, in the domain adaptation task, we obtained better results than standard optimal transport based approaches. Furthermore the results obtained by the coupling $\mathbf{\Gamma}$ and the transformation $f_{\mathbf{L}}$ are almost identical validating the approach. On the other hand, in a Computer Vision task, we have shown that the transformation $f_{\mathbf{L}}$ can be efficiently used on out-of-samples examples leading to visually smoother and better results than the standard approaches.

The framework presented in this chapter opens the door to several perspectives. First, from a theoretical standpoint the bound proposed raises some questions on the generalization ability of the barycentric mapping and on the estimation of the quality of the true barycentric mapping with respect to the target transformation. On a more practical side, note that in recent years regularized optimal transport has encountered a growing interest and several methods have been proposed to control the behaviour of the transport. As long as these regularization terms are convex, one could imagine use them in our framework. Another perspective could be to use our framework in a mini-batch setting where instead of learning from the whole dataset we can estimate a single function $f_{\mathcal{S}\to\mathcal{T}}$ from several couplings $\mathbf{\Gamma}$

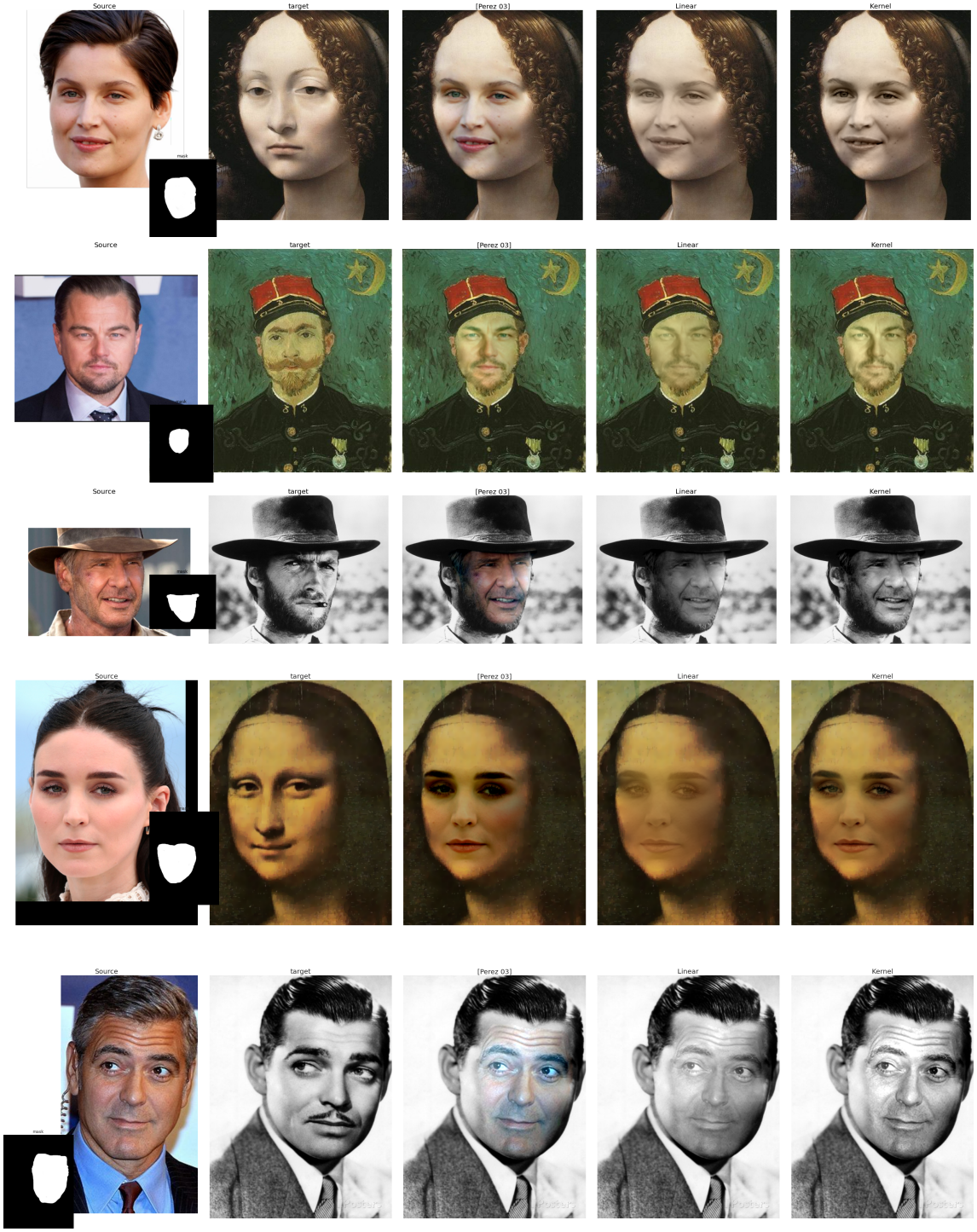


Figure 6.2: Illustrations of seamless copies with gradient adaptation. Each row is composed of the source image, the corresponding selection zone Ω described as a binary mask, and the target image. We compare here the linear (4th column) and kernel (5th column) versions of the map $f_{\mathbf{L}}$ with the original method of Pérez et al. (2003) (3rd column).

Figure 6.3: Illustration of failure of style adaptation.

optimized on different splits of the examples. We also believe that our framework could allow the use of the notion of optimal transport in deep architectures as, contrary to the coupling $\mathbf{\Gamma}$, the function $f_{\mathbf{L}}$ can be used on out-of-samples examples. As a last perspective we think that our framework could be used to learn some metrics in some other contexts such as unsupervised learning.

Conclusion and Perspectives

In this thesis we addressed the problem of learning a metric with a controlled behaviour. We considered two kinds of control on the learned metric. On the one hand we addressed the problem of learning with respect to a reference metric available either under the form of a given distance between a limited number of pairs or directly through its model. On the other hand we considered the problem of learning the underlying transformation of a Mahalanobis distance either able to precisely move the examples toward a set destination or controlled by a geometric transformation. Our contributions have taken the form of algorithmic and theoretical solutions.

Summary of the Contributions

Most of metric learning algorithm are interested in learning metrics able to bring closer similar examples and to push far away dissimilar ones. However, in some cases, one might be interested in predicting an exact value between two examples. This is for example the case when one has access to a limited number pairs for which the value of a reference metric is known. In our first contribution we addressed the problem of learning an approximation of this reference metric. We proposed a local metric learning algorithm and we theoretically analysed it showing that with a sufficient number of examples the learned model generalizes well. Furthermore we evaluated our approach on the computer vision problem of perceptual color differences. To this end we created a new dataset specifically designed for the problem at hand. Our empirical results showed the good behaviour of our approach and its ability to correctly approximate a reference metric. The dataset and the perceptually uniform distance that we learned are freely distributed (Perrot et al., 2014a).

Several approaches in metric learning empirically demonstrated the interest of using side information in the form of a source metric without theoretically proving that it was indeed beneficial. In the second contribution we proposed to bridge this gap. Hence we formalised the metric hypothesis transfer learning framework where the idea is to take into account a source metric in a biased regularization term. We proposed a theoretical analysis of this framework and, considering several theoretical approaches, we derived three different measures of goodness for the source metric. These measures are ways to evaluate the interest of a source metric for the problem at hand. Two of these measures are theoretical and thus are hard to use in practice. The third one, however, is empirical which means that it can

be computed and used to select the best source metric among a set of candidates. As an illustration it allowed us to propose an algorithm to weight the importance of the source metric. We demonstrated the wide range of applicability of the metric hypothesis transfer learning framework by proving that several loss functions and regularization terms fall into our theoretical analysis. Furthermore we empirically evaluated it on metric learning and semi-supervised domain adaptation tasks.

Most of metric learning approaches use similarity and dissimilarity constraints to learn a metric but do not explicitly control the behaviour of the underlying transformation. In our third contribution we addressed this problem by proposing a new approach where the desired destination of the examples is explicitly chosen through so-called virtual points. It allowed us to carefully control the learned metric and thus to design more problem specific models. For example for classification we proposed class based virtual points where the metric is learned such that each axis is discriminative for a particular class. We showed that our approach can easily be kernelized making it able to learn very expressive metrics. We also proposed a theoretical study demonstrating that our approach can be tied to a classic metric learning method. Lastly we empirically demonstrated its good performance on several well known datasets.

In our fourth contribution we addressed a problem similar to the third one. However instead of explicitly controlling the behaviour of each example individually we proposed to force a metric to follow a particular geometric transformation. Hence we considered transformations implied by the coupling learned by a discrete optimal transport problem which is particularly relevant for domain adaptation tasks. We proposed a solution to jointly learn this coupling and an associated metric through its underlying transformation. We derived an efficient optimization scheme and we showed that this approach could be further interpreted as a modification of our third contribution where the transformation and the virtual points are jointly learned. We empirically demonstrated the good behaviour of our approach on unsupervised domain adaptation and seamless copy tasks.

Perspectives

We have already presented specific perspectives for each of our contributions. In this part, we rather try to discuss more general future works that can represent some new research directions from the work presented in this thesis.

From an algorithmic standpoint our contributions are mainly based on batch optimization problems. A first perspective would be to extend the concepts presented here to the online learning setting. Following this idea it could be interesting to develop some mechanisms able to detect a potential drift in the distribution of the examples and to automatically change the behaviour of the metric accordingly. Such an approach could for example be used when learning a metric to solve a problem of tracking of objects in videos where the variations in the scene might call for different behaviours. Another perspective would be to consider active learning to improve the control over the metric. For example when learning a transformation

it could be interesting to obtain some feedback from the user to verify that the examples are moving in the correct direction. A motivating example could be the problem of domain adaptation, where active learning has already proven to be useful (Berlind and Urner, 2015), where obtaining some carefully selected feedbacks could ensure that the metric is correctly estimating the shift between the distributions.

On a more theoretical standpoint we can notice that in this thesis we were mainly interested in the generalization ability of the learned metric and not in its impact on the subsequent application. Following the latter idea Balcan et al. (2008) have demonstrated that the error of a linear classifier is tied to a measure of goodness of the similarity used to learn it. This goodness is related to the capacity of a metric at bringing closer similar examples and pushing far away dissimilar ones. However when learning a metric with controlled behaviour this measure might not be adapted. For example when learning a metric with respect to a reference metric (Chapters 3 and 4) one would probably be more interested in considering a measure telling if the learned metric is better than the reference one. Similarly when learning a transformation for a domain adaptation task (Chapter 4 and 6) one would probably put its focus on the ability of the metric at aligning the source and target distributions. It implies that a measure of the goodness of the metric is task dependent. An interesting perspective would be to consider some theoretical frameworks able to take into account a measure of goodness related to the task at hand and to prove that a good metric is indeed beneficial.

Another theoretical perspective would be to derive generalization bounds with a fast rate of convergence in the presence of additional informations. In Chapter 4 we proposed a first solution to this problem using the Rademacher complexity framework along with the additional information that is the goodness of a source metric. However this solution is not satisfying in the sense that the constraint on the source metric was somehow stronger than the result obtained on the learned metric. Nevertheless this is still encouraging in the sense that it shows that under strong assumptions it is possible to obtain a fast rate of convergence. Thus, if one manages to obtain weaker assumptions (See e.g. Srebro et al. (2010c)) it might be possible to obtain more meaningful results.

List of Publications

Publications in International Conferences

Michaël Perrot, Nicolas Courty, Rémi Flamary, and Amaury Habrard. Mapping estimation of discrete optimal transport. In *Advances in Neural Information Processing Systems (NIPS-16)*, 2016

Michaël Perrot and Amaury Habrard. Regressive virtual metric learning. In *Advances in Neural Information Processing Systems (NIPS-15)*, pages 1810–1818, 2015c

Michaël Perrot and Amaury Habrard. A theoretical analysis of metric hypothesis transfer learning. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 1708–1717, 2015d

Michaël Perrot, Amaury Habrard, Damien Muselet, and Marc Sebban. Modeling perceptual color differences by local metric learning. In *European Conference on Computer Vision (ECCV-15)*, pages 96–111. Springer International Publishing, 2014b

Communications in National Conferences

Michaël Perrot and Amaury Habrard. Bornes en généralisation à convergence rapide pour le transfert d'hypothèses en apprentissage de métriques. In *Conférence francophone sur l'Apprentissage Automatique (CAp-16)*, 2016

Michaël Perrot and Amaury Habrard. Apprentissage de mtriques par rgression. In *Conférence francophone sur l'Apprentissage Automatique (CAp-15)*, 2015a

Michaël Perrot and Amaury Habrard. Transfert d'informations en apprentissage de mtriques : une analyse thorique. In *Conférence francophone sur l'Apprentissage Automatique (CAp-15)*, 2015b

Michaël Perrot, Amaury Habrard, Damien Muselet, and Marc Sebban. Modélisation de distances couleur uniformes par apprentissage de métriques locales. In *Conférence francophone sur l'Apprentissage Automatique (CAp-14)*, 2014c

Appendix A

Theorems, Lemmas and Definitions

We present here several Theorems, Lemmas and Definitions used throughout the thesis but not presented in the main text for the sake of readability.

A.1 Properties of Loss Functions

Definition A.1 (k-lipschitz continuity)**.** *A loss function $l(h, \mathbf{z})$ is k-lipschitz with respect to its first argument if, for any hypotheses $h, g \in \mathcal{H}$ and any example $\mathbf{z}$, there exists $k \geq 0$ such that:*

$$|l(h, \mathbf{z}) - l(g, \mathbf{z})| \leq k \|h - g\|. \tag{A.1}$$

The k-lipschitz property ensures that the loss deviation does not exceed the deviation between two hypotheses h and g with respect to a positive constant k.

Definition A.2 ((σ, m)-admissibility)**.** *A loss function for metric learning $l(\mathbf{M}, \mathbf{z}, \mathbf{z}')$ is (σ, m)-admissible, with respect to $\mathbf{M}$, if it is convex with respect to its first argument and if for any two pairs of examples $\mathbf{z}, \mathbf{z}'$ and $\mathbf{z}'', \mathbf{z}'''$, we have:*

$$\left|l\left(\mathbf{M}, \mathbf{z}, \mathbf{z}'\right) - l\left(\mathbf{M}, \mathbf{z}'', \mathbf{z}'''\right)\right| \leq \sigma \left|\delta_{yy'} - \delta_{y''y'''}\right| + m \tag{A.2}$$

where $\delta_{yy'} = 1$ if $y = y'$ and -1 otherwise, i.e. $\left|\delta_{yy'} - \delta_{y''y'''}\right| \in \{0, 2\}$.

This property bounds the difference between the losses of two pairs of examples by a value only related to the labels plus a constant independent from the matrix $\mathbf{M}$.

Definition A.3 (H-smooth loss (Srebro et al., 2010c))**.** *A function $f : \mathbb{R} \to \mathbb{R}$ is H-smooth if it is twice differentiable and its first derivative is H-lipschitz continuous (Definition A.1).*

Lemma A.1 (Srebro et al. (2010b, Lemma B.1))**.** *For any H-smooth non-negative function $f : \mathbb{R} \to \mathbb{R}$ and $t, r \in \mathbb{R}$ we have that:*

$$(f(t) - f(r))^2 \leq 6H(f(t) + f(r))(t - r)^2. \tag{A.3}$$

A.2 Properties of Norms

Definition A.4 (Dual norm)**.** *Let $\|\cdot\|$ be a norm over a normed space $\mathcal{M}$. The dual norm $\|\cdot\|_*$ is defined as:*

$$\left\|\mathbf{M}'\right\|_* = \max_{\mathbf{M}} \left\langle \mathbf{M}', \mathbf{M} \right\rangle : \|\mathbf{M}\| \leq 1. \tag{A.4}$$

A.3 Properties of Algorithms

Definition A.5 (On-average-replace-one-stability (Shalev-Shwartz and Ben-David, 2014a))**.** *Let n be the number of examples considered during the learning step. Let $\epsilon : \mathbb{N} \to \mathbb{R}$ be monotonically decreasing with respect to n and let $U(1,n)$ be the uniform distribution over $1,\ldots,n$. An algorithm is on-average-replace-two-stable with rate $\epsilon(n)$ if for any distribution $\mathcal{D}_\mathcal{T}$:*

$$\mathop{\mathbb{E}}_{\substack{T\sim\mathcal{D}_\mathcal{T} \\ i\sim U(1,n) \\ \mathbf{z}\sim\mathcal{D}_\mathcal{T}}} \left[l\left(h_{T^i}, \mathbf{z}_i\right) - l\left(h_T, \mathbf{z}_i\right) \right] \leq \epsilon(n) \tag{A.5}$$

where h_T, respectively h_{T^i}, is the optimal solution when learning with the training set T, respectively T^i. T^i is obtained by replacing $\mathbf{z}_i$, the i^{th} example of T, by $\mathbf{z}$.

A.4 Concentration Inequalities

Theorem A.1 (McDiarmid's inequality (McDiarmid, 1989))**.** *Let $X_1, ..., X_n$ be n independent random variables taking values in $\mathcal{X}$ and let $Z = f(X_1, ..., X_n)$. If for each $1 \leq i \leq n$, there exists a constant c_i such that*

$$\sup_{\mathbf{x}_1,\ldots,\mathbf{x}_n,\mathbf{x}_i'\in\mathcal{X}} \left| f(\mathbf{x}_1,\ldots,\mathbf{x}_i,\ldots,\mathbf{x}_n) - f\big(\mathbf{x}_1,\ldots,\mathbf{x}_i',\ldots,\mathbf{x}_n\big) \right| \leq c_i, \forall 1 \leq i \leq n, \tag{A.6}$$

then for any $\epsilon > 0$,

$$\Pr\left(|Z - \mathbb{E}\left[Z\right]| \geq \epsilon\right) \leq 2\exp\left(\frac{-2\epsilon^2}{\sum_{i=1}^n c_i^2}\right). \tag{A.7}$$

Proposition A.1 (Van Der Vaart and Wellner (1996))**.** *Let $(n_0, n_1, \ldots, n_K)$ an i.i.d. multinomial random variable with parameters $n = \sum_{j=0}^K n_j$ and $(P(C_0), P(C_1), \ldots, P(C_K))$. By the Breteganolle-Huber-Carol inequality we have:*

$$Pr\left(\sum_{j=0}^K \left|\frac{n_j}{n} - \Pr(C_j)\right| \geq \epsilon\right) \leq 2^K \exp\left(\frac{-n\epsilon^2}{2}\right), \tag{A.8}$$

hence with probability at least $1-\delta$,

$$\sum_{j=0}^K \left|\frac{n_j}{n} - \Pr(C_j)\right| \leq \sqrt{\frac{2K\ln(2) + 2\ln(\frac{1}{\delta})}{n}}. \tag{A.9}$$

A.5 Other Theorems

Theorem A.2 (Union bound)**.** *Given a countable set of events $E_1, E_2, E_3, \ldots$:*

$$\Pr\left(\bigcup_i E_i\right) \leq \sum_i \Pr\left(E_i\right). \tag{A.10}$$

Theorem A.3 (Cauchy-Schwarz inequality)**.** *Let $\mathcal{X}$ be a vector space equipped with an inner product $\langle\cdot,\cdot\rangle$ defining a norm $\|\cdot\|$. Let $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$, then:*

$$\left\langle \mathbf{x}, \mathbf{x}' \right\rangle^2 \leq \langle \mathbf{x}, \mathbf{x} \rangle^2 \left\langle \mathbf{x}', \mathbf{x}' \right\rangle^2 \tag{A.11}$$

$$\Leftrightarrow \qquad \left|\left\langle \mathbf{x}, \mathbf{x}' \right\rangle\right| \leq \|\mathbf{x}\| \left\|\mathbf{x}'\right\|. \tag{A.12}$$

Definition A.6 (Convexity)**.** *A function f is convex if for all $\mathbf{w}$, $\mathbf{u}$, and $\alpha \in [0,1]$ we have:*

$$f(\alpha\mathbf{w} + (1-\alpha)\mathbf{u}) \leq \alpha f(\mathbf{w}) + (1-\alpha) f(\mathbf{u}). \tag{A.13}$$

Definition A.7 (c-strong convexity)**.** *A function f is c-strongly convex if for all $\mathbf{w}$, $\mathbf{u}$, and $\alpha \in [0,1]$ we have:*

$$f(\alpha\mathbf{w} + (1-\alpha)\mathbf{u}) \leq \alpha f(\mathbf{w}) + (1-\alpha) f(\mathbf{u}) - \frac{c}{2}\alpha(1-\alpha)\|\mathbf{w}-\mathbf{u}\|_2^2. \tag{A.14}$$

Theorem A.4 (Jensen's inequality)**.** *For any convex function f of a random variable X we have:*

$$f\left(\mathop{\mathbb{E}}_X[X]\right) \leq \mathop{\mathbb{E}}_X[f(X)]. \tag{A.15}$$

Appendix B

Proofs of Chapter 3

B.1 Proof of Lemma 3.1

Lemma (Bounded loss function)**.** *For any $0 \leq j \leq K$, let $\mathbf{M}_{T_j}$ be the metric learned for region C_j with the training set T_j, we have that for any example $(\mathbf{x}, \mathbf{x}', \Delta) \sim \mathcal{D}_{\mathcal{T}j}$:*

$$0 \leq l\left(\mathbf{M}_{T_j}, (\mathbf{x}, \mathbf{x}', \Delta)\right) \leq B_j, \tag{B.1}$$

with $B_j = \max\left(\frac{\Delta_{\max}}{\sqrt{\lambda_j}}, \Delta_{\max}^2\right)$.

Proof. First of all note that the absolute value is always positive which gives the first inequality. Furthermore $\mathbf{M}_{T_j}$ is an optimal solution of Problem (3.5). Hence we have:

$$\hat{L}_{T_j}(\mathbf{M}_{T_j}) + \lambda_j \left\|\mathbf{M}_{T_j}\right\|_{\mathcal{F}}^2 \leq \hat{L}_{T_j}(\mathbf{0}) + \lambda_j \left\|\mathbf{0}\right\|_{\mathcal{F}}^2$$

$$\Leftrightarrow \quad \frac{1}{n_j} \sum_{(\mathbf{x},\mathbf{x}',\Delta)\in T_j} l\left(\mathbf{M}_{T_j}, (\mathbf{x}, \mathbf{x}', \Delta)\right) + \lambda_j \left\|\mathbf{M}_{T_j}\right\|_{\mathcal{F}}^2 \leq \frac{1}{n_j} \sum_{(\mathbf{x},\mathbf{x}',\Delta)\in T_j} l\left(\mathbf{0}, (\mathbf{x}, \mathbf{x}', \Delta)\right) + \lambda_j \left\|\mathbf{0}\right\|_{\mathcal{F}}^2$$

(Positive loss function and $\|\mathbf{0}\|_{\mathcal{F}} = 0$.)

$$\Rightarrow \quad \lambda_j \left\|\mathbf{M}_{T_j}\right\|_{\mathcal{F}}^2 \leq \frac{1}{n_j} \sum_{(\mathbf{x},\mathbf{x}',\Delta)\in T_j} l\left(\mathbf{0}, (\mathbf{x}, \mathbf{x}', \Delta)\right)$$

($l\left(\mathbf{0}, (\mathbf{x}, \mathbf{x}', \Delta)\right) \leq \Delta_{\max}^2$.)

$$\Rightarrow \quad \lambda_j \left\|\mathbf{M}_{T_j}\right\|_{\mathcal{F}}^2 \leq \Delta_{\max}^2$$

$$\Rightarrow \quad \left\|\mathbf{M}_{T_j}\right\|_{\mathcal{F}} \leq \frac{\Delta_{\max}}{\sqrt{\lambda_j}}. \tag{B.2}$$

We can now prove the second inequality of the lemma:

$$l\left(\mathbf{M}_{T_j}, (\mathbf{x}, \mathbf{x}', \Delta)\right) = \left|(\mathbf{x} - \mathbf{x}')^T \mathbf{M}_{T_j} (\mathbf{x} - \mathbf{x}') - \Delta^2\right|$$

(Difference between two positive values.)

$$\leq \max\left((\mathbf{x} - \mathbf{x}')^T \mathbf{M}_{T_j} (\mathbf{x} - \mathbf{x}'), \Delta^2\right)$$

(Cauchy-Shwartz inequality (Theorem A.3).)

$$\leq \max\left(\left\|\mathbf{x}-\mathbf{x}'\right\|_2^2 \left\|\mathbf{M}_{T_j}\right\|_{\mathcal{F}}, \Delta^2\right)$$

(Equation (B.2), $\left\|\mathbf{x}-\mathbf{x}'\right\|_2 \leq 1$ and $\Delta \leq \Delta_{\max}$.)

$$\leq \max\left(\frac{\Delta_{\max}}{\sqrt{\lambda_j}}, \Delta_{\max}^2\right) \tag{B.3}$$

Setting $B_j = \max\left(\frac{\Delta_{\max}}{\sqrt{\lambda_j}}, \Delta_{\max}^2\right)$ gives the lemma. □

B.2 Proof of Lemma 3.2

Lemma (k-lipschitz continuity)**.** *Let $\mathbf{M}_{T_j}$ and $\mathbf{M}'_{T_j}$ be two matrices for a region C_j and $(\mathbf{x}, \mathbf{x}', \Delta)$ be an example. Our loss $l\left(\mathbf{M}_{T_j}, (\mathbf{x}, \mathbf{x}', \Delta)\right)$ is k-lipschitz continuous with $k = D_j^2$.*

Proof.

$$\begin{aligned}
&\left|l\left(\mathbf{M}_{T_j}, (\mathbf{x}, \mathbf{x}', \Delta)\right) - l\left(\mathbf{M}'_{T_j}, (\mathbf{x}, \mathbf{x}', \Delta)\right)\right| \\
&\qquad = \left|\left|(\mathbf{x}-\mathbf{x}')^T \mathbf{M}_{T_j}(\mathbf{x}-\mathbf{x}') - \Delta^2\right| - \left|(\mathbf{x}-\mathbf{x}')^T \mathbf{M}'_{T_j}(\mathbf{x}-\mathbf{x}') - \Delta^2\right|\right| \\
&\qquad\qquad \text{(Triangle inequality.)} \\
&\qquad \leq \left|(\mathbf{x}-\mathbf{x}')^T \mathbf{M}_{T_j}(\mathbf{x}-\mathbf{x}') - (\mathbf{x}-\mathbf{x}')^T \mathbf{M}'_{T_j}(\mathbf{x}-\mathbf{x}')\right| \\
&\qquad = \left|(\mathbf{x}-\mathbf{x}')^T (\mathbf{M}_{T_j} - \mathbf{M}'_{T_j})(\mathbf{x}-\mathbf{x}')\right| \\
&\qquad\qquad \text{(Cauchy-Shwartz inequality (Theorem A.3).)} \\
&\qquad \leq \left\|\mathbf{x}-\mathbf{x}'\right\|_2^2 \left\|\mathbf{M}_{T_j} - \mathbf{M}'_{T_j}\right\|_{\mathcal{F}} \\
&\qquad\qquad (D_j = \max_{(\mathbf{x},\mathbf{x}',\Delta)\sim\mathcal{D}_{\mathcal{T}j}} \left\|\mathbf{x}-\mathbf{x}'\right\|_2.) \\
&\qquad \leq D_j^2 \left\|\mathbf{M}_{T_j} - \mathbf{M}'_{T_j}\right\|_{\mathcal{F}}
\end{aligned}$$

Setting $k = D_j^2$ gives the lemma. □

B.3 Proof of Lemma 3.3

To prove Lemma 3.3 we need the following technical lemma.

Lemma B.1. *Let $F_{T_j}(\mathbf{M}) = \hat{L}_{T_j}(\mathbf{M}) + \lambda_j \left\|\mathbf{M}\right\|_{\mathcal{F}}^2$ and $F_{T_j^i}(\mathbf{M}) = \hat{L}_{T_j^i}(\mathbf{M}) + \lambda_j \left\|\mathbf{M}\right\|_{\mathcal{F}}^2$ be the functions minimized in Problem (3.5) where T_j and T_j^i are two training samples of n_j examples. T_j^i is obtained by replacing example i from T_j by another example drawn independently from $\mathcal{D}_{\mathcal{T}j}$. Let $\mathbf{M}_{T_j}$ and $\mathbf{M}_{T_j^i}$ be their respective minimizers, and λ_j be the regularization parameter used in our algorithm. Let $\Delta_{\mathbf{M}_{T_j}} = \mathbf{M}_{T_j} - \mathbf{M}_{T_j^i}$, then, we have, for any $t \in [0,1]$,*

$$\left\|\mathbf{M}_{T_j}\right\|_{\mathcal{F}}^2 - \left\|\mathbf{M}_{T_j} - t\Delta_{\mathbf{M}_{T_j}}\right\|_{\mathcal{F}}^2 + \left\|\mathbf{M}_{T_j^i}\right\|_{\mathcal{F}}^2 - \left\|\mathbf{M}_{T_j^i} + t\Delta_{\mathbf{M}_{T_j}}\right\|_{\mathcal{F}}^2 \leq \frac{2kt}{\lambda_j n_j}\left\|\Delta_{\mathbf{M}_{T_j}}\right\|_{\mathcal{F}}. \tag{B.4}$$

Proof. $\hat{L}_{T_j^i}$ is a convex function, thus, for any $t \in [0,1]$, we can write:

$$\hat{L}_{T_j^i}(\mathbf{M}_{T_j} - t\Delta_{\mathbf{M}_{T_j}}) - \hat{L}_{T_j^i}(\mathbf{M}_{T_j}) \leq t\left(\hat{L}_{T_j^i}(\mathbf{M}_{T_j^i}) - \hat{L}_{T_j^i}(\mathbf{M}_{T_j})\right), \tag{B.5}$$

$$\hat{L}_{T_j^i}(\mathbf{M}_{T_j^i} + t\Delta_{\mathbf{M}_{T_j}}) - \hat{L}_{T_j^i}(\mathbf{M}_{T_j^i}) \leq t\left(\hat{L}_{T_j^i}(\mathbf{M}_{T_j}) - \hat{L}_{T_j^i}(\mathbf{M}_{T_j^i})\right). \tag{B.6}$$

By summing Inequalities (B.5) and (B.6) we obtain

$$\hat{L}_{T_j^i}(\mathbf{M}_{T_j} - t\Delta_{\mathbf{M}_{T_j}}) - \hat{L}_{T_j^i}(\mathbf{M}_{T_j}) + \hat{L}_{T_j^i}(\mathbf{M}_{T_j^i} + t\Delta_{\mathbf{M}_{T_j}}) - \hat{L}_{T_j^i}(\mathbf{M}_{T_j^i}) \leq 0. \tag{B.7}$$

Since $\mathbf{M}_{T_j}$ and $\mathbf{M}_{T_j^i}$ are minimizers of F_{T_j} and $F_{T_j^i}$, we can write:

$$F_{T_j}(\mathbf{M}_{T_j}) - F_{T_j}\left(\mathbf{M}_{T_j} - t\Delta_{\mathbf{M}_{T_j}}\right) \leq 0, \tag{B.8}$$

$$F_{T_j^i}\left(\mathbf{M}_{T_j^i}\right) - F_{T_j^i}\left(\mathbf{M}_{T_j^i} + t\Delta_{\mathbf{M}_{T_j}}\right) \leq 0. \tag{B.9}$$

By summing Inequalities (B.8) and (B.9), we obtain:

$$\begin{aligned}\hat{L}_{T_j}(\mathbf{M}_{T_j}) - \hat{L}_{T_j}(\mathbf{M}_{T_j} - t\Delta_{\mathbf{M}_{T_j}}) + \lambda_j \left\|\mathbf{M}_{T_j}\right\|_{\mathcal{F}}^2 - \lambda_j \left\|\mathbf{M}_{T_j} - t\Delta_{\mathbf{M}_{T_j}}\right\|_{\mathcal{F}}^2 \\ + \hat{L}_{T_j^i}(\mathbf{M}_{T_j^i}) - \hat{L}_{T_j^i}(\mathbf{M}_{T_j^i} + t\Delta_{\mathbf{M}_{T_j}}) + \lambda_j \left\|\mathbf{M}_{T_j^i}\right\|_{\mathcal{F}}^2 - \lambda_j \left\|\mathbf{M}_{T_j^i} + t\Delta_{\mathbf{M}_{T_j}}\right\|_{\mathcal{F}}^2 \leq 0.\end{aligned} \tag{B.10}$$

We can now sum Inequalities (B.7) and (B.10) to obtain:

$$\begin{aligned}\hat{L}_{T_j}(\mathbf{M}_{T_j}) - \hat{L}_{T_j^i}(\mathbf{M}_{T_j}) - \hat{L}_{T_j}(\mathbf{M}_{T_j} - t\Delta_{\mathbf{M}_{T_j}}) + \hat{L}_{T_j^i}(\mathbf{M}_{T_j} - t\Delta_{\mathbf{M}_{T_j}}) \\ + \lambda_j \left\|\mathbf{M}_{T_j}\right\|_{\mathcal{F}}^2 - \lambda_j \left\|\mathbf{M}_{T_j} - t\Delta_{\mathbf{M}_{T_j}}\right\|_{\mathcal{F}}^2 + \lambda_j \left\|\mathbf{M}_{T_j^i}\right\|_{\mathcal{F}}^2 - \lambda_j \left\|\mathbf{M}_{T_j^i} + t\Delta_{\mathbf{M}_{T_j}}\right\|_{\mathcal{F}}^2 \leq 0.\end{aligned} \tag{B.11}$$

From Inequality (B.11), we can write:

$$\lambda_j \left\|\mathbf{M}_{T_j}\right\|_{\mathcal{F}}^2 - \lambda_j \left\|\mathbf{M}_{T_j} - t\Delta_{\mathbf{M}_{T_j}}\right\|_{\mathcal{F}}^2 + \lambda_j \left\|\mathbf{M}_{T_j^i}\right\|_{\mathcal{F}}^2 - \lambda_j \left\|\mathbf{M}_{T_j^i} + t\Delta_{\mathbf{M}_{T_j}}\right\|_{\mathcal{F}}^2 \leq C \tag{B.12}$$

with

$$C = \hat{L}_{T_j^i}(\mathbf{M}_{T_j}) - \hat{L}_{T_j}(\mathbf{M}_{T_j}) + \hat{L}_{T_j}(\mathbf{M}_{T_j} - t\Delta_{\mathbf{M}_{T_j}}) - \hat{L}_{T_j^i}(\mathbf{M}_{T_j} - t\Delta_{\mathbf{M}_{T_j}}).$$

We are now looking for a bound on C:

$$\begin{aligned}C &\leq \left|\hat{L}_{T_j}(\mathbf{M}_{T_j} - t\Delta_{\mathbf{M}_{T_j}}) - \hat{L}_{T_j}(\mathbf{M}_{T_j}) + \hat{L}_{T_j^i}(\mathbf{M}_{T_j}) - \hat{L}_{T_j^i}(\mathbf{M}_{T_j} - t\Delta_{\mathbf{M}_{T_j}})\right| \\ &= \frac{1}{n_j}\left|\sum_{(\mathbf{x},\mathbf{x}',\Delta)\in T_j} l\left(\mathbf{M}_{T_j} - t\Delta_{\mathbf{M}_{T_j}}, (\mathbf{x},\mathbf{x}',\Delta)\right) - l\left(\mathbf{M}_{T_j}, (\mathbf{x},\mathbf{x}',\Delta)\right)\right. \\ &\quad \left. + \sum_{(\mathbf{x}^i,\mathbf{x}^{i\prime},\Delta)\in T_j^i} l\left(\mathbf{M}_{T_j}, (\mathbf{x}^i,\mathbf{x}^{i\prime},\Delta)\right) - l\left(\mathbf{M}_{T_j} - t\Delta_{\mathbf{M}_{T_j}}, (\mathbf{x}^i,\mathbf{x}^{i\prime},\Delta)\right)\right|\end{aligned}$$

(T_j and T_j^i only differ by one pair.)

$$
\begin{aligned}
&= \frac{1}{n_j}\left|l\left(\mathbf{M}_{T_j} - t\Delta_{\mathbf{M}_{T_j}}, (\mathbf{x}_i, \mathbf{x}_i', \Delta)\right) - l\left(\mathbf{M}_{T_j}, (\mathbf{x}_i, \mathbf{x}_i', \Delta)\right)\right. \\
&\quad \left. + l\left(\mathbf{M}_{T_j}, (\mathbf{x}_i^i, \mathbf{x}_i^{i\prime}, \Delta)\right) - l\left(\mathbf{M}_{T_j} - t\Delta_{\mathbf{M}_{T_j}}, (\mathbf{x}_i^i, \mathbf{x}_i^{i\prime}, \Delta)\right)\right| \\
&\qquad\qquad \text{(Triangle inequality.)} \\
&\leq \frac{1}{n_j}\left|l\left(\mathbf{M}_{T_j} - t\Delta_{\mathbf{M}_{T_j}}, (\mathbf{x}_i, \mathbf{x}_i', \Delta)\right) - l\left(\mathbf{M}_{T_j}, (\mathbf{x}_i, \mathbf{x}_i', \Delta)\right)\right| \\
&\quad + \frac{1}{n_j}\left|l\left(\mathbf{M}_{T_j}, (\mathbf{x}_i^i, \mathbf{x}_i^{i\prime}, \Delta)\right) - l\left(\mathbf{M}_{T_j} - t\Delta_{\mathbf{M}_{T_j}}, (\mathbf{x}_i^i, \mathbf{x}_i^{i\prime}, \Delta)\right)\right| \\
&\qquad\qquad \text{(k-lipschitz continuity (Lemma 3.2).)} \\
&\leq \frac{1}{n_j}k\left\|\mathbf{M}_{T_j} - t\Delta_{\mathbf{M}_{T_j}} - \mathbf{M}_{T_j}\right\|_{\mathcal{F}} + \frac{1}{n_j}k\left\|\mathbf{M}_{T_j} - \mathbf{M}_{T_j} - t\Delta_{\mathbf{M}_{T_j}}\right\|_{\mathcal{F}} \\
&= \frac{1}{n_j}k\left\|t\Delta_{\mathbf{M}_{T_j}}\right\|_{\mathcal{F}} + \frac{1}{n_j}k\left\|-t\Delta_{\mathbf{M}_{T_j}}\right\|_{\mathcal{F}} \\
&\qquad\qquad \text{(Definition of } \|\cdot\|_{\mathcal{F}}\text{.)} \\
&= \frac{2kt}{n_j}\left\|\Delta_{\mathbf{M}_{T_j}}\right\|_{\mathcal{F}}.
\end{aligned}
$$

Combining this bound on C with Equation (B.12) and dividing both sides by λ_j gives the lemma. □

We are now ready to prove Lemma 3.3.

Lemma (Uniform stability per region C_j)**.** *Given two training samples T_j and T_j^i of n_j examples where T_j^i is obtained by replacing example i from T_j by another example drawn independently from $\mathcal{D}_{\mathcal{T}j}$. Let $\mathbf{M}_{T_j}$ and $\mathbf{M}_{T_j^i}$ be the respective optimal solutions of Problem* (3.5) *when learning with T_j and T_j^i. In region C_j our problem is β_j uniformly stable with $\beta_j = \frac{2D_j^4}{\lambda_j}$.*

Proof. By setting $t = \frac{1}{2}$ in Lemma B.1, one can obtain for the left hand side:

$$
\left\|\mathbf{M}_{T_j}\right\|_{\mathcal{F}}^2 - \left\|\mathbf{M}_{T_j} - \frac{1}{2}\Delta_{\mathbf{M}_{T_j}}\right\|_{\mathcal{F}}^2 + \left\|\mathbf{M}_{T_j^i}\right\|_{\mathcal{F}}^2 - \left\|\mathbf{M}_{T_j^i} + \frac{1}{2}\Delta_{\mathbf{M}_{T_j}}\right\|_{\mathcal{F}}^2 = \frac{1}{2}\left\|\Delta_{\mathbf{M}_{T_j}}\right\|_{\mathcal{F}}^2
$$

and thus

$$
\frac{1}{2}\left\|\Delta_{\mathbf{M}_{T_j}}\right\|_{\mathcal{F}}^2 \leq \frac{2k\frac{1}{2}}{\lambda_j n_j}\left\|\Delta_{\mathbf{M}_{T_j}}\right\|_{\mathcal{F}}, \tag{B.13}
$$

which implies

$$
\left\|\Delta_{\mathbf{M}_{T_j}}\right\|_{\mathcal{F}} \leq \frac{2k}{\lambda_j n_j}. \tag{B.14}
$$

Since our loss is k-lipschitz (Lemma 3.2) we have:

$$
\left|l\left(\mathbf{M}_{T_j}, (\mathbf{x}, \mathbf{x}', \Delta)\right) - l\left(\mathbf{M}_{T_j^i}, (\mathbf{x}, \mathbf{x}', \Delta)\right)\right| \leq k\left\|\Delta_{\mathbf{M}_{T_j}}\right\|_{\mathcal{F}} \tag{B.15}
$$

$$\leq \frac{2k^2}{\lambda_j n_j}. \tag{B.16}$$

In particular, since $k = D_j^2$,

$$\sup_{(\mathbf{x},\mathbf{x}',\Delta)\sim\mathcal{D}_{\mathcal{T}_j}} \left| l\left(\mathbf{M}_{T_j}, (\mathbf{x},\mathbf{x}',\Delta)\right) - l\left(\mathbf{M}_{T_j^i}, (\mathbf{x},\mathbf{x}',\Delta)\right)\right| \leq \frac{2D_j^4}{\lambda_j n_j}. \tag{B.17}$$

The last inequality matches the definition of uniform stability, Definition 1.3. Thus setting $\beta_j = \frac{2D_j^4}{\lambda_j}$ gives the lemma. □

B.4 Proof of Lemma 3.4

Lemma (Bound on $\mathbb{E}_{T_j\sim\mathcal{D}_{\mathcal{T}_j}}\left[R_{T_j}\right]$)**.** *For any β_j uniformly stable learning method of estimation error $R_{T_j} = L_{\mathcal{T}_j}(\mathbf{M}_{T_j}) - \hat{L}_{T_j}(\mathbf{M}_{T_j})$ for a training set T_j, we have:*

$$\mathop{\mathbb{E}}_{T_j\sim\mathcal{D}_{\mathcal{T}_j}}\left[R_{T_j}\right] \leq \frac{\beta_j}{n_j}. \tag{B.18}$$

Proof. First of all note that:

$$\begin{aligned}
\mathop{\mathbb{E}}_{T_j\sim\mathcal{D}_{\mathcal{T}_j}}\left[\hat{L}_{T_j}(\mathbf{M}_{T_j})\right] &= \frac{1}{n_j}\sum_{(\mathbf{x}_i,\mathbf{x}_i',\Delta)\in T_j} \mathop{\mathbb{E}}_{T_j\sim\mathcal{D}_{\mathcal{T}_j}}\left[l\left(\mathbf{M}_{T_j}, (\mathbf{x}_i,\mathbf{x}_i',\Delta)\right)\right] \\
&= \frac{1}{n_j}\sum_{(\mathbf{x}_i,\mathbf{x}_i',\Delta)\in T_j} \mathop{\mathbb{E}}_{T_j,(\mathbf{x},\mathbf{x}',\Delta)\sim\mathcal{D}_{\mathcal{T}_j}}\left[l\left(\mathbf{M}_{T_j^i}, (\mathbf{x},\mathbf{x}',\Delta)\right)\right] \\
&= \mathop{\mathbb{E}}_{T_j,(\mathbf{x},\mathbf{x}',\Delta)\sim\mathcal{D}_{\mathcal{T}_j}}\left[l\left(\mathbf{M}_{T_j^i}, (\mathbf{x},\mathbf{x}',\Delta)\right)\right]
\end{aligned}$$

The second to last equality comes from the fact that the pairs are drawn independently from $\mathcal{D}_{\mathcal{T}_j}$ and thus changing one example with another does not change the expectation. From this equality we deduce:

$$\begin{aligned}
\mathop{\mathbb{E}}_{T_j\sim\mathcal{D}_{\mathcal{T}_j}}\left[R_{T_j}\right] &= \mathop{\mathbb{E}}_{T_j\sim\mathcal{D}_{\mathcal{T}_j}}\left[L_{\mathcal{T}_j}(\mathbf{M}_{T_j}) - \hat{L}_{T_j}(\mathbf{M}_{T_j})\right] \\
&= \mathop{\mathbb{E}}_{T_j,(\mathbf{x},\mathbf{x}',\Delta)\sim\mathcal{D}_{\mathcal{T}_j}}\left[l\left(\mathbf{M}_{T_j}, (\mathbf{x},\mathbf{x}',\Delta)\right) - l\left(\mathbf{M}_{T_j^i}, (\mathbf{x},\mathbf{x}',\Delta)\right)\right] \\
&\leq \mathop{\mathbb{E}}_{T_j,(\mathbf{x},\mathbf{x}',\Delta)\sim\mathcal{D}_{\mathcal{T}_j}}\left|l\left(\mathbf{M}_{T_j}, (\mathbf{x},\mathbf{x}',\Delta)\right) - l\left(\mathbf{M}_{T_j^i}, (\mathbf{x},\mathbf{x}',\Delta)\right)\right| \\
&\qquad\qquad \text{(Uniform stability (Lemma 3.3).)} \\
&\leq \frac{\beta_j}{n}
\end{aligned}$$

□

B.5 Proof of Lemma 3.5

Lemma (Bound on $\left|R_{T_j} - R_{T_j^i}\right|$)**.** *For any β_j uniformly stable learning method of estimation error $R_{T_j} = L_{\mathcal{T}_j}(\mathbf{M}_{T_j}) - \hat{L}_{T_j}(\mathbf{M}_{T_j})$ for a training set T_j and any B_j bounded loss function we have:*

$$\left|R_{T_j} - R_{T_j^i}\right| \leq \frac{2\beta_j + B_j}{n_j}. \tag{B.19}$$

Proof.

$$\begin{aligned}
\left|R_{T_j} - R_{T_j^i}\right| &= \left|L_{\mathcal{T}_j}(\mathbf{M}_{T_j}) - \hat{L}_{T_j}(\mathbf{M}_{T_j}) - \left(L_{\mathcal{T}_j}(\mathbf{M}_{T_j^i}) - \hat{L}_{T_j^i}(\mathbf{M}_{T_j^i})\right)\right| \\
&= \left|L_{\mathcal{T}_j}(\mathbf{M}_{T_j}) - \hat{L}_{T_j}(\mathbf{M}_{T_j}) - L_{\mathcal{T}_j}(\mathbf{M}_{T_j^i}) + \hat{L}_{T_j^i}(\mathbf{M}_{T_j^i}) - \hat{L}_{T_j}(\mathbf{M}_{T_j^i}) + \hat{L}_{T_j}(\mathbf{M}_{T_j^i})\right| \\
&\qquad \text{(Triangle inequality.)} \\
&\leq \left|L_{\mathcal{T}_j}(\mathbf{M}_{T_j}) - L_{\mathcal{T}_j}(\mathbf{M}_{T_j^i})\right| + \left|\hat{L}_{T_j}(\mathbf{M}_{T_j^i}) - \hat{L}_{T_j}(\mathbf{M}_{T_j})\right| + \left|\hat{L}_{T_j^i}(\mathbf{M}_{T_j^i}) - \hat{L}_{T_j}(\mathbf{M}_{T_j^i})\right| \\
&\qquad \text{(Definition of } L_{\mathcal{T}_j} \text{ and triangle inequality.)} \\
&\leq \mathop{\mathbb{E}}_{(\mathbf{x},\mathbf{x}',\Delta)\sim\mathcal{D}_{\mathcal{T}_j}} \left[\left|l\left(\mathbf{M}_{T_j}, (\mathbf{x},\mathbf{x}',\Delta)\right) - l\left(\mathbf{M}_{T_j^i}, (\mathbf{x},\mathbf{x}',\Delta)\right)\right|\right] \\
&\quad + \left|\hat{L}_{T_j}(\mathbf{M}_{T_j^i}) - \hat{L}_{T_j}(\mathbf{M}_{T_j})\right| + \left|\hat{L}_{T_j^i}(\mathbf{M}_{T_j^i}) - \hat{L}_{T_j}(\mathbf{M}_{T_j^i})\right| \\
&\qquad \text{(Uniform stability (Lemma 3.3).)} \\
&\leq \frac{\beta_j}{n_j} + \left|\hat{L}_{T_j}(\mathbf{M}_{T_j^i}) - \hat{L}_{T_j}(\mathbf{M}_{T_j})\right| + \left|\hat{L}_{T_j^i}(\mathbf{M}_{T_j^i}) - \hat{L}_{T_j}(\mathbf{M}_{T_j^i})\right| \\
&\qquad \text{(Definition of } \hat{L}_{T_j} \text{ and triangle inequality.)} \\
&\leq \frac{\beta_j}{n_j} + \frac{1}{n_j} \sum_{(\mathbf{x},\mathbf{x}',\Delta)\in T_j} \left|l\left(\mathbf{M}_{T_j^i}, (\mathbf{x},\mathbf{x}',\Delta)\right) - l\left(\mathbf{M}_{T_j}, (\mathbf{x},\mathbf{x}',\Delta)\right)\right| \\
&\quad + \left|\hat{L}_{T_j^i}(\mathbf{M}_{T_j^i}) - \hat{L}_{T_j}(\mathbf{M}_{T_j^i})\right| \\
&\qquad \text{(Uniform stability (Lemma 3.3).)} \\
&\leq \frac{2\beta_j}{n_j} + \left|\hat{L}_{T_j^i}(\mathbf{M}_{T_j^i}) - \hat{L}_{T_j}(\mathbf{M}_{T_j^i})\right| \\
&\qquad (T_j \text{ and } T_j^i \text{ only differ by one pair.}) \\
&= \frac{2\beta_j}{n_j} + \frac{1}{n_j}\left|l\left(\mathbf{M}_{T_j^i}, (\mathbf{x}_i^i, \mathbf{x}_i^{i\prime}, \Delta)\right) - l\left(\mathbf{M}_{T_j^i}, (\mathbf{x}_i, \mathbf{x}_i', \Delta)\right)\right| \\
&\qquad \text{(Bounded loss function (Lemma 3.1).)} \\
&\leq \frac{2\beta_j}{n_j} + \frac{B_j}{n_j}
\end{aligned}$$

□

Appendix C

Proofs of Chapter 4

C.1 Proof of Lemma 4.1

Before proving the lemma we show that the biased Frobenius norm is strongly convex (Definition A.7).

Lemma C.1 (Strong convexity of the biased Frobenius norm)**.** *The biased Frobenius norm is* 2*-strongly convex.*

Proof.

$$
\begin{aligned}
&\left\|\alpha(\mathbf{M}) + (1-\alpha)(\mathbf{M}') - \mathbf{M}_{\mathcal{S}}\right\|_{\mathcal{F}}^2 \\
&\qquad = \left\|\alpha(\mathbf{M} - \mathbf{M}_{\mathcal{S}}) + (1-\alpha)(\mathbf{M}' - \mathbf{M}_{\mathcal{S}})\right\|_{\mathcal{F}}^2 \\
&\qquad\qquad\qquad\qquad\qquad\qquad \text{(2-strong convexity of the non biased Frobenius norm.)} \\
&\qquad \leq \alpha \left\|\mathbf{M} - \mathbf{M}_{\mathcal{S}}\right\|_{\mathcal{F}}^2 + (1-\alpha)\left\|\mathbf{M}' - \mathbf{M}_{\mathcal{S}}\right\|_{\mathcal{F}}^2 - \frac{2}{2}\alpha(1-\alpha)\left\|\mathbf{M} - \mathbf{M}_{\mathcal{S}} - \mathbf{M}' + \mathbf{M}_{\mathcal{S}}\right\|_{\mathcal{F}}^2 \\
&\qquad = \alpha \left\|\mathbf{M} - \mathbf{M}_{\mathcal{S}}\right\|_{\mathcal{F}}^2 + (1-\alpha)\left\|\mathbf{M}' - \mathbf{M}_{\mathcal{S}}\right\|_{\mathcal{F}}^2 - \frac{2}{2}\alpha(1-\alpha)\left\|\mathbf{M} - \mathbf{M}'\right\|_{\mathcal{F}}^2
\end{aligned}
$$

□

We can now prove Lemma 4.1.

Lemma (On-average-replace-two-stability)**.** *Given n the number of training examples, drawn i.i.d. from $\mathcal{D}_{\mathcal{T}}$, considered and a k-lipschitz loss function, any algorithm solving Problem* (4.1) *is on-average-replace-two-stable with $\epsilon(n) = \frac{8k^2}{\lambda n}$.*

Proof. Let $\mathbf{M}_T$, respectively $\mathbf{M}_{T^{i^j}}$, be the optimal solution when learning with the training set T, respectively T^{i^j}. Let $\mathbf{z}_k, \mathbf{z}_k^i, \mathbf{z}_k^{i^j}$ respectively be the k^{th} examples of training sets T, T^i, T^{i^j}. We have:

$$
\begin{aligned}
&\hat{L}_T(\mathbf{M}_{T^{i^j}}) + \lambda\left\|\mathbf{M}_{T^{i^j}} - \mathbf{M}_{\mathcal{S}}\right\|_{\mathcal{F}}^2 - (\hat{L}_T(\mathbf{M}_T) + \lambda\left\|\mathbf{M}_T - \mathbf{M}_{\mathcal{S}}\right\|_{\mathcal{F}}^2) \\
&\qquad = \frac{1}{n(n-1)} \sum_{\mathbf{z}\in T} \sum_{\substack{\mathbf{z}'\in T \\ \mathbf{z}\neq\mathbf{z}'}} l\left(\mathbf{M}_{T^{i^j}} - \mathbf{M}_T, \mathbf{z}, \mathbf{z}'\right) - l\left(\mathbf{M}_T, \mathbf{z}, \mathbf{z}'\right)
\end{aligned}
$$

(Adding and removing the same quantity.)

$$
\begin{aligned}
&= \hat{L}_{T^i}(\mathbf{M}_{T^{ij}}) + \lambda \left\|\mathbf{M}_{T^{ij}} - \mathbf{M}_{\mathcal{S}}\right\|_{\mathcal{F}}^2 - (\hat{L}_{T^i}(\mathbf{M}_T) + \lambda \left\|\mathbf{M}_T - \mathbf{M}_{\mathcal{S}}\right\|_{\mathcal{F}}^2) \\
&\quad - \sum_{\substack{\mathbf{z}^{i'} \in T^i \\ \mathbf{z}_i^i \neq \mathbf{z}^{i'}}} \frac{l\left(\mathbf{M}_{T^{ij}}, \mathbf{z}_i^i, \mathbf{z}^{i'}\right) - l\left(\mathbf{M}_T, \mathbf{z}_i^i, \mathbf{z}^{i'}\right)}{n(n-1)} - \sum_{\substack{\mathbf{z}^i \in T^i \\ \mathbf{z}^i \neq \mathbf{z}_i^{i'}}} \frac{l\left(\mathbf{M}_{T^{ij}}, \mathbf{z}^i, \mathbf{z}_i^{i'}\right) - l\left(\mathbf{M}_T, \mathbf{z}^i, \mathbf{z}_i^{i'}\right)}{n(n-1)} \\
&\quad + \sum_{\substack{\mathbf{z}' \in T \\ \mathbf{z}_i \neq \mathbf{z}'}} \frac{l\left(\mathbf{M}_{T^{ij}}, \mathbf{z}_i, \mathbf{z}'\right) - l\left(\mathbf{M}_T, \mathbf{z}_i, \mathbf{z}'\right)}{n(n-1)} + \sum_{\substack{\mathbf{z} \in T \\ \mathbf{z} \neq \mathbf{z}_i'}} \frac{l\left(\mathbf{M}_{T^{ij}}, \mathbf{z}, \mathbf{z}_i'\right) - l\left(\mathbf{M}_T, \mathbf{z}, \mathbf{z}_i'\right)}{n(n-1)}
\end{aligned}
$$

(Adding and removing the same quantity.)

$$
\begin{aligned}
&= \hat{L}_{T^{ij}}(\mathbf{M}_{T^{ij}}) + \lambda \left\|\mathbf{M}_{T^{ij}} - \mathbf{M}_{\mathcal{S}}\right\|_{\mathcal{F}}^2 - (\hat{L}_{T^{ij}}(\mathbf{M}_T) + \lambda \left\|\mathbf{M}_T - \mathbf{M}_{\mathcal{S}}\right\|_{\mathcal{F}}^2) \\
&\quad - \sum_{\substack{\mathbf{z}^{i'} \in T^i \\ \mathbf{z}_i^i \neq \mathbf{z}^{i'}}} \frac{l\left(\mathbf{M}_{T^{ij}}, \mathbf{z}_i^i, \mathbf{z}^{i'}\right) - l\left(\mathbf{M}_T, \mathbf{z}_i^i, \mathbf{z}^{i'}\right)}{n(n-1)} - \sum_{\substack{\mathbf{z}^i \in T^i \\ \mathbf{z}^i \neq \mathbf{z}_i^{i'}}} \frac{l\left(\mathbf{M}_{T^{ij}}, \mathbf{z}^i, \mathbf{z}_i^{i'}\right) - l\left(\mathbf{M}_T, \mathbf{z}^i, \mathbf{z}_i^{i'}\right)}{n(n-1)} \\
&\quad + \sum_{\substack{\mathbf{z}' \in T \\ \mathbf{z}_i \neq \mathbf{z}'}} \frac{l\left(\mathbf{M}_{T^{ij}}, \mathbf{z}_i, \mathbf{z}'\right) - l\left(\mathbf{M}_T, \mathbf{z}_i, \mathbf{z}'\right)}{n(n-1)} + \sum_{\substack{\mathbf{z} \in T \\ \mathbf{z} \neq \mathbf{z}_i'}} \frac{l\left(\mathbf{M}_{T^{ij}}, \mathbf{z}, \mathbf{z}_i'\right) - l\left(\mathbf{M}_T, \mathbf{z}, \mathbf{z}_i'\right)}{n(n-1)} \\
&\quad - \sum_{\substack{\mathbf{z}^{ij\,\prime} \in T^{ij} \\ \mathbf{z}_j^{ij} \neq \mathbf{z}^{ij\,\prime}}} \frac{l\left(\mathbf{M}_{T^{ij}}, \mathbf{z}_j^{ij}, \mathbf{z}^{ij\,\prime}\right) - l\left(\mathbf{M}_T, \mathbf{z}_j^{ij}, \mathbf{z}^{ij\,\prime}\right)}{n(n-1)} - \sum_{\substack{\mathbf{z}^{ij} \in T^{ij} \\ \mathbf{z}^{ij} \neq \mathbf{z}_j^{ij\,\prime}}} \frac{l\left(\mathbf{M}_{T^{ij}}, \mathbf{z}^{ij}, \mathbf{z}_j^{ij\,\prime}\right) - l\left(\mathbf{M}_T, \mathbf{z}^{ij}, \mathbf{z}_j^{ij\,\prime}\right)}{n(n-1)} \\
&\quad + \sum_{\substack{\mathbf{z}^{i'} \in T^i \\ \mathbf{z}_j^i \neq \mathbf{z}^{i'}}} \frac{l\left(\mathbf{M}_{T^{ij}}, \mathbf{z}_j^i, \mathbf{z}^{i'}\right) - l\left(\mathbf{M}_T, \mathbf{z}_j^i, \mathbf{z}^{i'}\right)}{n(n-1)} + \sum_{\substack{\mathbf{z}^i \in T^i \\ \mathbf{z}^i \neq \mathbf{z}_j^{i'}}} \frac{l\left(\mathbf{M}_{T^{ij}}, \mathbf{z}^i, \mathbf{z}_j^{i'}\right) - l\left(\mathbf{M}_T, \mathbf{z}^i, \mathbf{z}_j^{i'}\right)}{n(n-1)}
\end{aligned}
$$

(Triangle inequality and k-lipschitz continuity.)

$$
\begin{aligned}
&\leq \hat{L}_{T^{ij}}(\mathbf{M}_{T^{ij}}) + \lambda \left\|\mathbf{M}_{T^{ij}} - \mathbf{M}_{\mathcal{S}}\right\|_{\mathcal{F}}^2 - (\hat{L}_{T^{ij}}(\mathbf{M}_T) + \lambda \left\|\mathbf{M}_T - \mathbf{M}_{\mathcal{S}}\right\|_{\mathcal{F}}^2) \\
&\quad + \frac{8k \left\|\mathbf{M}_{T^{ij}} - \mathbf{M}\right\|_{\mathcal{F}}}{n}
\end{aligned}
$$

(Convex loss and optimality of $\mathbf{M}_{T^{ij}}$ when learning with T^{ij}.)

$$
\leq \frac{8k \left\|\mathbf{M}_{T^{ij}} - \mathbf{M}\right\|_{\mathcal{F}}}{n}
$$

Furthermore, from the 2-strong convexity of the biased Frobenius norm used as a regularization term (Lemma C.1) we deduce that Problem (4.1) is 2λ-strongly convex (Definition A.7). Given $\mathbf{M}_T$ the optimal solution of Problem (4.1) when learning with T, we have:

$$
\hat{L}_T(\mathbf{M}_{T^{ij}}) + \lambda \left\|\mathbf{M}_{T^{ij}} - \mathbf{M}_{\mathcal{S}}\right\|_{\mathcal{F}}^2 - (\hat{L}_T(\mathbf{M}_T) + \lambda \left\|\mathbf{M}_T - \mathbf{M}_{\mathcal{S}}\right\|_{\mathcal{F}}^2) \geq \frac{2\lambda}{2} \left\|\mathbf{M}_{T^{ij}} - \mathbf{M}_T\right\|_{\mathcal{F}}^2.
$$

Combining the two inequalities we obtain:

$$
\begin{aligned}
& \lambda \left\|\mathbf{M}_{T^{ij}} - \mathbf{M}_T\right\|_{\mathcal{F}}^2 \leq \frac{8k \left\|\mathbf{M}_{T^{ij}} - \mathbf{M}_T\right\|_{\mathcal{F}}}{n} \\
\Rightarrow \quad & \left\|\mathbf{M}_{T^{ij}} - \mathbf{M}_T\right\|_{\mathcal{F}} \leq \frac{8k}{\lambda n} \\
\Rightarrow \quad & \left|l\left(\mathbf{M}_{T^{ij}}, \mathbf{z}, \mathbf{z}'\right) - l\left(\mathbf{M}_T, \mathbf{z}, \mathbf{z}'\right)\right| \leq k \left\|\mathbf{M}_{T^{ij}} - \mathbf{M}_T\right\|_{\mathcal{F}} \leq \frac{8k^2}{\lambda n}.
\end{aligned}
$$

(C.1)

The last inequality is obtained thanks to the k-lipschitzness of the loss. Taking the expectation on both sides gives the lemma. □

C.2 Proof of Lemma 4.2

Lemma (Uniform stability)**.** *Given a positive, convex, k-lipschitz loss and a training sample T of n examples drawn i.i.d. from $\mathcal{D}_{\mathcal{T}}$, an algorithm solving Problem* (4.1) *has a uniform stability in $\beta = \frac{4k^2}{\lambda}$.*

Proof. Let $\Delta_{\mathbf{M}} = \mathbf{M}_T - \mathbf{M}_{T^i}$ where $\mathbf{M}_T$ is the optimal solution when learning with set T and $\mathbf{M}_{T^i}$ is the optimal solution when learning with set T^i. The empirical risk is convex by sum of convex functions, thus

$$\hat{L}_{T^i}(\mathbf{M}_T - t\Delta_{\mathbf{M}}) - \hat{L}_{T^i}(\mathbf{M}_T) \leq t(\hat{L}_{T^i}(\mathbf{M}_{T^i}) - \hat{L}_{T^i}(\mathbf{M}_T))$$
$$\hat{L}_{T^i}(\mathbf{M}_{T^i} + t\Delta_{\mathbf{M}}) - \hat{L}_{T^i}(\mathbf{M}_{T^i}) \leq t(\hat{L}_{T^i}(\mathbf{M}_T) - \hat{L}_{T^i}(\mathbf{M}_{T^i}))$$

Summing up the two inequalities gives:

$$\hat{L}_{T^i}(\mathbf{M}_T - t\Delta_{\mathbf{M}}) - \hat{L}_{T^i}(\mathbf{M}_T) + \hat{L}_{T^i}(\mathbf{M}_{T^i} + t\Delta_{\mathbf{M}}) - \hat{L}_{T^i}(\mathbf{M}_{T^i}) \leq 0. \tag{C.2}$$

Problem (4.1) is convex by sum of convex functions, thus:

$$\begin{aligned}\hat{L}_T(\mathbf{M}_T) + \lambda \left\|\mathbf{M}_T - \mathbf{M}_{\mathcal{S}}\right\|_{\mathcal{F}}^2 - \hat{L}_T(\mathbf{M}_T - t\Delta_{\mathbf{M}}) - \lambda \left\|\mathbf{M}_T - t\Delta_{\mathbf{M}} - \mathbf{M}_{\mathcal{S}}\right\|_{\mathcal{F}}^2 \\ + \hat{L}_{T^i}(\mathbf{M}_{T^i}) + \lambda \left\|\mathbf{M}_{T^i} - \mathbf{M}_{\mathcal{S}}\right\|_{\mathcal{F}}^2 - \hat{L}_{T^i}(\mathbf{M}_{T^i} + t\Delta_{\mathbf{M}}) - \lambda \left\|\mathbf{M}_{T^i} + t\Delta_{\mathbf{M}} - \mathbf{M}_{\mathcal{S}}\right\|_{\mathcal{F}}^2 \leq 0.\end{aligned} \tag{C.3}$$

Summing Inequalities (C.2) and (C.3) gives:

$$\begin{aligned}\hat{L}_T(\mathbf{M}_T) - \hat{L}_{T^i}(\mathbf{M}_T) + \hat{L}_{T^i}(\mathbf{M}_T - t\Delta_{\mathbf{M}}) - \hat{L}_T(\mathbf{M}_T - t\Delta_{\mathbf{M}}) \\ + \lambda \left\|\mathbf{M}_T - \mathbf{M}_{\mathcal{S}}\right\|_{\mathcal{F}}^2 - \lambda \left\|\mathbf{M}_T - t\Delta_{\mathbf{M}} - \mathbf{M}_{\mathcal{S}}\right\|_{\mathcal{F}}^2 \\ + \lambda \left\|\mathbf{M}_{T^i} - \mathbf{M}_{\mathcal{S}}\right\|_{\mathcal{F}}^2 - \lambda \left\|\mathbf{M}_{T^i} + t\Delta_{\mathbf{M}} - \mathbf{M}_{\mathcal{S}}\right\|_{\mathcal{F}}^2 \leq 0.\end{aligned} \tag{C.4}$$

From Inequality (C.4) we have:

$$\begin{aligned}\lambda \left\|\mathbf{M}_T - \mathbf{M}_{\mathcal{S}}\right\|_{\mathcal{F}}^2 - \lambda \left\|\mathbf{M}_T - t\Delta_{\mathbf{M}} - \mathbf{M}_{\mathcal{S}}\right\|_{\mathcal{F}}^2 \\ + \lambda \left\|\mathbf{M}_{T^i} - \mathbf{M}_{\mathcal{S}}\right\|_{\mathcal{F}}^2 - \lambda \left\|\mathbf{M}_{T^i} + t\Delta_{\mathbf{M}} - \mathbf{M}_{\mathcal{S}}\right\|_{\mathcal{F}}^2 \leq C.\end{aligned} \tag{C.5}$$

where $C = \hat{L}_{T^i}(\mathbf{M}_T) - \hat{L}_T(\mathbf{M}_T) + \hat{L}_T(\mathbf{M}_T - t\Delta_{\mathbf{M}}) - \hat{L}_{T^i}(\mathbf{M}_T - t\Delta_{\mathbf{M}})$. We are now looking for a bound on C:

$$\begin{aligned}C &\leq \left|\hat{L}_T(\mathbf{M}_T - t\Delta_{\mathbf{M}}) - \hat{L}_T(\mathbf{M}_T) + \hat{L}_{T^i}(\mathbf{M}_T) - \hat{L}_{T^i}(\mathbf{M}_T - t\Delta_{\mathbf{M}})\right| \\ &= \frac{1}{n(n-1)} \left| \sum_{\mathbf{z}\in T} \sum_{\substack{\mathbf{z}'\in T \\ \mathbf{z}\neq\mathbf{z}'}} l\left(\mathbf{M}_T - t\Delta_{\mathbf{M}}, \mathbf{z}, \mathbf{z}'\right) - l\left(\mathbf{M}_T, \mathbf{z}, \mathbf{z}'\right)\right.\end{aligned}$$

$$+ \sum_{\mathbf{z}^i \in T^i} \sum_{\substack{\mathbf{z}^{i\prime} \in T^i \\ \mathbf{z}^i \neq \mathbf{z}^{i\prime}}} l\left(\mathbf{M}_T, \mathbf{z}^i, \mathbf{z}^{i\prime}\right) - l\left(\mathbf{M}_T - t\Delta_{\mathbf{M}}, \mathbf{z}^i, \mathbf{z}^{i\prime}\right) \Bigg|$$

(T and T^i only differ by one example.)

$$= \frac{1}{n(n-1)} \Bigg| \sum_{\substack{\mathbf{z}' \in T \\ \mathbf{z}_i \neq \mathbf{z}'}} l\left(\mathbf{M}_T - t\Delta_{\mathbf{M}}, \mathbf{z}_i, \mathbf{z}'\right) - l\left(\mathbf{M}_T, \mathbf{z}_i, \mathbf{z}'\right) + \sum_{\substack{\mathbf{z} \in T \\ \mathbf{z} \neq \mathbf{z}'_i}} l\left(\mathbf{M}_T - t\Delta_{\mathbf{M}}, \mathbf{z}, \mathbf{z}'_i\right) - l\left(\mathbf{M}_T, \mathbf{z}, \mathbf{z}'_i\right)$$

$$+ \sum_{\substack{\mathbf{z}^{i\prime} \in T^i \\ \mathbf{z}^i_i \neq \mathbf{z}^{i\prime}}} l\left(\mathbf{M}_T, \mathbf{z}^i_i, \mathbf{z}^{i\prime}\right) - l\left(\mathbf{M}_T - t\Delta_{\mathbf{M}}, \mathbf{z}^i_i, \mathbf{z}^{i\prime}\right) + \sum_{\substack{\mathbf{z}^i \in T^i \\ \mathbf{z}^i \neq \mathbf{z}^{i\prime}_i}} l\left(\mathbf{M}_T, \mathbf{z}^i, \mathbf{z}^{i\prime}_i\right) - l\left(\mathbf{M}_T - t\Delta_{\mathbf{M}}, \mathbf{z}^i, \mathbf{z}^{i\prime}_i\right) \Bigg|$$

(Triangle inequality.)

$$\begin{aligned}
&= \frac{1}{n(n-1)} \sum_{\substack{\mathbf{z}' \in T \\ \mathbf{z}_i \neq \mathbf{z}'}} \left| l\left(\mathbf{M}_T - t\Delta_{\mathbf{M}}, \mathbf{z}_i, \mathbf{z}'\right) - l\left(\mathbf{M}_T, \mathbf{z}_i, \mathbf{z}'\right) \right| \\
&\quad + \frac{1}{n(n-1)} \sum_{\substack{\mathbf{z} \in T \\ \mathbf{z} \neq \mathbf{z}'_i}} \left| l\left(\mathbf{M}_T - t\Delta_{\mathbf{M}}, \mathbf{z}, \mathbf{z}'_i\right) - l\left(\mathbf{M}_T, \mathbf{z}, \mathbf{z}'_i\right) \right| \\
&\quad + \frac{1}{n(n-1)} \sum_{\substack{\mathbf{z}^{i\prime} \in T^i \\ \mathbf{z}^i_i \neq \mathbf{z}^{i\prime}}} \left| l\left(\mathbf{M}_T, \mathbf{z}^i_i, \mathbf{z}^{i\prime}\right) - l\left(\mathbf{M}_T - t\Delta_{\mathbf{M}}, \mathbf{z}^i_i, \mathbf{z}^{i\prime}\right) \right| \\
&\quad + \frac{1}{n(n-1)} \sum_{\substack{\mathbf{z}^i \in T^i \\ \mathbf{z}^i \neq \mathbf{z}^{i\prime}_i}} \left| l\left(\mathbf{M}_T, \mathbf{z}^i, \mathbf{z}^{i\prime}_i\right) - l\left(\mathbf{M}_T - t\Delta_{\mathbf{M}}, \mathbf{z}^i, \mathbf{z}^{i\prime}_i\right) \right|
\end{aligned}$$

(k-lipschitz continuous loss.)

$$\leq \frac{4(n-1)}{n(n-1)} k \left\| \mathbf{M}_T - \mathbf{M}_T + t\Delta_{\mathbf{M}} \right\|_{\mathcal{F}}$$

(Definition of $\|\cdot\|_{\mathcal{F}}$.)

$$\leq \frac{4kt}{n} \left\| \Delta_{\mathbf{M}} \right\|_{\mathcal{F}}$$

Furthermore, setting $t = \frac{1}{2}$ in the left hand side of Inequality (C.5), we have:

$$\begin{aligned}
&\lambda \left\| \mathbf{M}_T - \mathbf{M}_{\mathcal{S}} \right\|^2_{\mathcal{F}} - \lambda \left\| \mathbf{M}_T - \frac{1}{2}\Delta_{\mathbf{M}} - \mathbf{M}_{\mathcal{S}} \right\|^2_{\mathcal{F}} \\
&\qquad + \lambda \left\| \mathbf{M}_{T^i} - \mathbf{M}_{\mathcal{S}} \right\|^2_{\mathcal{F}} - \lambda \left\| \mathbf{M}_{T^i} + \frac{1}{2}\Delta_{\mathbf{M}} - \mathbf{M}_{\mathcal{S}} \right\|^2_{\mathcal{F}} = \frac{\lambda}{2} \left\| \Delta_{\mathbf{M}} \right\|^2_{\mathcal{F}}.
\end{aligned}$$

Following this we have:

$$\begin{aligned}
&\frac{\lambda}{2} \left\| \Delta_{\mathbf{M}} \right\|^2_{\mathcal{F}} \leq \frac{4k}{2n} \left\| \Delta_{\mathbf{M}} \right\|_{\mathcal{F}} \\
\Leftrightarrow \qquad &\left\| \Delta_{\mathbf{M}} \right\|_{\mathcal{F}} \leq \frac{4k}{\lambda n}. \qquad \text{(C.6)}
\end{aligned}$$

Using the k-lipschitz continuity of the loss, we have:

$$\sup_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_\mathcal{T}} \left| l\left(\mathbf{M}_T, \mathbf{z}, \mathbf{z}'\right) - l\left(\mathbf{M}_{T^i}, \mathbf{z}, \mathbf{z}'\right)\right| \leq k \left\|\Delta_\mathbf{M}\right\|_\mathcal{F} \leq \frac{4k^2}{\lambda n}.$$

Setting $\beta = \frac{4k^2}{\lambda}$ concludes the proof. □

C.3 Proof of Lemma 4.3

Lemma (Bound on $\mathbb{E}_{T\sim\mathcal{D}_\mathcal{T}}[R_T]$)**.** *For any β uniformly stable learning method of estimation error $R_T = L_\mathcal{T}(\mathbf{M}_T) - \hat{L}_T(\mathbf{M}_T)$ for a training set T, we have:*

$$\mathop{\mathbb{E}}_{T\sim\mathcal{D}_\mathcal{T}}[R_T] \leq \frac{2\beta}{n}.$$

Proof. First of all note that:

$$\begin{aligned}
\mathop{\mathbb{E}}_{T\sim\mathcal{D}_\mathcal{T}}\left[\hat{L}_T(\mathbf{M}_T)\right] &= \frac{1}{n(n-1)} \sum_{\mathbf{z}_i\in T} \sum_{\substack{\mathbf{z}_j\in T \\ \mathbf{z}_i\neq\mathbf{z}_j}} \mathop{\mathbb{E}}_{T\sim\mathcal{D}_\mathcal{T}} \left[l\left(\mathbf{M}_T, \mathbf{z}_i, \mathbf{z}_j\right)\right] \\
&= \frac{1}{n(n-1)} \sum_{\mathbf{z}_i\in T} \sum_{\substack{\mathbf{z}_j\in T \\ \mathbf{z}_i\neq\mathbf{z}_j}} \mathop{\mathbb{E}}_{T,\mathbf{z},\mathbf{z}'\sim\mathcal{D}_\mathcal{T}} \left[l\left(\mathbf{M}_{T^{ij}}, \mathbf{z}, \mathbf{z}'\right)\right] \\
&= \mathop{\mathbb{E}}_{T,\mathbf{z},\mathbf{z}'\sim\mathcal{D}_\mathcal{T}} \left[l\left(\mathbf{M}_{T^{ij}}, \mathbf{z}, \mathbf{z}'\right)\right]
\end{aligned}$$

The second to last equality comes from the fact that the examples are drawn independently from $\mathcal{D}_\mathcal{T}$ and thus changing one example with another twice does not change the expectation. From this equality we deduce:

$$\begin{aligned}
\mathop{\mathbb{E}}_{T\sim\mathcal{D}_\mathcal{T}}[R_T] &= \mathop{\mathbb{E}}_{T\sim\mathcal{D}_\mathcal{T}}\left[L_\mathcal{T}(\mathbf{M}_T) - \hat{L}_T(\mathbf{M}_T)\right] \\
&= \mathop{\mathbb{E}}_{T,\mathbf{z},\mathbf{z}'\sim\mathcal{D}_\mathcal{T}} \left[l\left(\mathbf{M}_T, \mathbf{z}, \mathbf{z}'\right) - l\left(\mathbf{M}_{T^{ij}}, \mathbf{z}, \mathbf{z}'\right)\right] \\
&= \mathop{\mathbb{E}}_{T,\mathbf{z},\mathbf{z}'\sim\mathcal{D}_\mathcal{T}} \left[l\left(\mathbf{M}_T, \mathbf{z}, \mathbf{z}'\right) - l\left(\mathbf{M}_{T^i}, \mathbf{z}, \mathbf{z}'\right) + l\left(\mathbf{M}_{T^i}, \mathbf{z}, \mathbf{z}'\right) - l\left(\mathbf{M}_{T^{ij}}, \mathbf{z}, \mathbf{z}'\right)\right] \\
&\qquad\qquad \text{(Triangle inequality.)} \\
&\leq \mathop{\mathbb{E}}_{T,\mathbf{z},\mathbf{z}'\sim\mathcal{D}_\mathcal{T}} \left|l\left(\mathbf{M}_T, \mathbf{z}, \mathbf{z}'\right) - l\left(\mathbf{M}_{T^i}, \mathbf{z}, \mathbf{z}'\right)\right| + \mathop{\mathbb{E}}_{T,\mathbf{z},\mathbf{z}'\sim\mathcal{D}_\mathcal{T}} \left|l\left(\mathbf{M}_{T^i}, \mathbf{z}, \mathbf{z}'\right) - l\left(\mathbf{M}_{T^{ij}}, \mathbf{z}, \mathbf{z}'\right)\right| \\
&\qquad\qquad \text{(Uniform stability (Lemma 4.2).)} \\
&\leq \frac{2\beta}{n}
\end{aligned}$$

□

C.4 Proof of Lemma 4.4

Lemma (Bound on $|R_T - R_{T^i}|$)**.** *For any β uniformly stable learning method of estimation error $R_T = L_\mathcal{T}(\mathbf{M}_T) - \hat{L}_T(\mathbf{M}_T)$ for a training set T and any (σ, m)-admissible loss function*

we have:

$$|R_T - R_{T^i}| \leq \frac{2\beta + 4\sigma + 2m}{n}. \tag{C.7}$$

Proof.

$$\begin{aligned}
|R_T - R_{T^i}| &= \left|L_{\mathcal{T}}(\mathbf{M}_T) - \hat{L}_T(\mathbf{M}_T) - (L_{\mathcal{T}}(\mathbf{M}_{T^i}) - \hat{L}_{T^i}(\mathbf{M}_{T^i}))\right| \\
&= \left|L_{\mathcal{T}}(\mathbf{M}_T) - L_{\mathcal{T}}(\mathbf{M}_{T^i}) + \hat{L}_{T^i}(\mathbf{M}_{T^i}) - \hat{L}_{T^i}(\mathbf{M}_T) + \hat{L}_{T^i}(\mathbf{M}_T) - \hat{L}_T(\mathbf{M}_T)\right| \\
&\qquad \text{(Triangle inequality.)} \\
&\leq |L_{\mathcal{T}}(\mathbf{M}_T) - L_{\mathcal{T}}(\mathbf{M}_{T^i})| + \left|\hat{L}_{T^i}(\mathbf{M}_{T^i}) - \hat{L}_{T^i}(\mathbf{M}_T)\right| + \left|\hat{L}_{T^i}(\mathbf{M}_T) - \hat{L}_T(\mathbf{M}_T)\right| \\
&\qquad \text{(Uniform stability (Lemma 4.2).)} \\
&\leq \frac{2\beta}{n} + \left|\hat{L}_{T^i}(\mathbf{M}_T) - \hat{L}_T(\mathbf{M}_T)\right| \\
&= \frac{2\beta}{n} + \frac{1}{n(n-1)}\left|\sum_{\mathbf{z}^i \in T^i}\sum_{\substack{\mathbf{z}^{i\prime} \in T^i \\ \mathbf{z}^i \neq \mathbf{z}^{i\prime}}} l\left(\mathbf{M}_T, \mathbf{z}^i, \mathbf{z}^{i\prime}\right) - \sum_{\mathbf{z} \in T}\sum_{\substack{\mathbf{z}' \in T \\ \mathbf{z} \neq \mathbf{z}'}} l\left(\mathbf{M}_T, \mathbf{z}, \mathbf{z}'\right)\right| \\
&\qquad (T \text{ and } T^i \text{ only differ by one example, } \forall j \neq i, \mathbf{z}_j = \mathbf{z}_j^i \text{ and } \mathbf{z}_j' = \mathbf{z}_j^{i\prime}.) \\
&= \frac{2\beta}{n} + \frac{1}{n(n-1)}\left|\sum_{\substack{\mathbf{z}' \in T^i \\ \mathbf{z}_i^i \neq \mathbf{z}'}} l\left(\mathbf{M}_T, \mathbf{z}_i^i, \mathbf{z}'\right) - \sum_{\substack{\mathbf{z}' \in T \\ \mathbf{z}_i \neq \mathbf{z}'}} l\left(\mathbf{M}_T, \mathbf{z}_i, \mathbf{z}'\right)\right. \\
&\quad \left. + \sum_{\substack{\mathbf{z} \in T^i \\ \mathbf{z} \neq \mathbf{z}_i^{i\prime}}} l\left(\mathbf{M}_T, \mathbf{z}, \mathbf{z}_i^{i\prime}\right) - \sum_{\substack{\mathbf{z} \in T \\ \mathbf{z} \neq \mathbf{z}_i'}} l\left(\mathbf{M}_T, \mathbf{z}, \mathbf{z}_i'\right)\right| \\
&\qquad \text{(Triangle inequality.)} \\
&\leq \frac{2\beta}{n} + \frac{1}{n(n-1)} \sum_{\substack{\mathbf{z}' \in T, T^i \\ \mathbf{z}_i \neq \mathbf{z}', \mathbf{z}_i^i \neq \mathbf{z}'}} \left|l\left(\mathbf{M}_T, \mathbf{z}_i^i, \mathbf{z}'\right) - l\left(\mathbf{M}_T, \mathbf{z}_i, \mathbf{z}'\right)\right| \\
&\quad + \frac{1}{n(n-1)} \sum_{\substack{\mathbf{z} \in T, T^i \\ \mathbf{z} \neq \mathbf{z}_i', \mathbf{z} \neq \mathbf{z}_i^{i\prime}}} \left|l\left(\mathbf{M}_T, \mathbf{z}, \mathbf{z}_i^{i\prime}\right) - l\left(\mathbf{M}_T, \mathbf{z}, \mathbf{z}_i'\right)\right| \\
&\qquad ((\sigma, m)\text{-admissible loss (Definition A.2).)} \\
&\leq \frac{2\beta}{n} + \frac{2(n-1)}{n(n-1)}\left(\sigma \sup_{\mathbf{z}, \mathbf{z}', \mathbf{z}'', \mathbf{z}''' \sim \mathcal{D}_{\mathcal{T}}} \left|yy' - y''y'''\right| + m\right) \\
&= \frac{2\beta}{n} + \frac{2\left(\sigma \sup_{\mathbf{z}, \mathbf{z}', \mathbf{z}'', \mathbf{z}''' \sim \mathcal{D}_{\mathcal{T}}} |yy' - y''y'''| + m\right)}{n}
\end{aligned}$$

Noting that by definition $\sup_{\mathbf{z}, \mathbf{z}', \mathbf{z}'', \mathbf{z}''' \sim \mathcal{D}_{\mathcal{T}}} |yy' - y''y'''| \leq 2$ gives the lemma. □

C.5 Proof of Lemma 4.6

Lemma (Bound on $\mathbb{E}_{T\sim\mathcal{D}_\mathcal{T}}[R_T]$)**.** *For any positive, convex and k-lipschitz (Definition 4.2) loss function and any algorithm with estimation error* $R_T = \sup_{\mathbf{M}\in\mathcal{M}_\mathcal{S}} \left[L_\mathcal{T}(\mathbf{M}) - \hat{L}_T(\mathbf{M})\right]$ *we have:*

$$\mathop{\mathbb{E}}_{T\sim\mathcal{D}_\mathcal{T}}[R_T] \leq 2kR_n(\mathcal{M}_\mathcal{S}).$$

Proof. Using standard properties on Rademacher variables Bartlett and Mendelson (2002); Shalev-Shwartz and Ben-David (2014a) and U-statistics Cao et al. (2016) we have that:

$$\begin{aligned}
&\mathop{\mathbb{E}}_{T\sim\mathcal{D}_\mathcal{T}} \sup_{\mathbf{M}\in\mathcal{M}_\mathcal{S}} \left[L_\mathcal{T}(\mathbf{M}) - \hat{L}_T(\mathbf{M})\right] \\
&\qquad = \mathop{\mathbb{E}}_{T\sim\mathcal{D}_\mathcal{T}} \sup_{\mathbf{M}\in\mathcal{M}_\mathcal{S}} \left[L_\mathcal{T}(\mathbf{M}) - \frac{1}{n(n-1)}\sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n} l\left(\mathbf{M}, \mathbf{z}_i, \mathbf{z}_j\right)\right] \\
&\qquad\qquad \text{(U-Statistics, Lemma 7 in Cao et al. (2016).)} \\
&\qquad \leq \mathop{\mathbb{E}}_{T\sim\mathcal{D}_\mathcal{T}} \sup_{\mathbf{M}\in\mathcal{M}_\mathcal{S}} \left[L_\mathcal{T}(\mathbf{M}) - \frac{1}{\lfloor\frac{n}{2}\rfloor}\sum_{i=1}^{\lfloor\frac{n}{2}\rfloor} l\left(\mathbf{M}, \mathbf{z}_i, \mathbf{z}_{\lfloor\frac{n}{2}\rfloor+i}\right)\right] \\
&\qquad \leq \mathop{\mathbb{E}}_{T\sim\mathcal{D}_\mathcal{T}} \sup_{\mathbf{M}\in\mathcal{M}_\mathcal{S}} \left[\mathop{\mathbb{E}}_{T'\sim\mathcal{D}_\mathcal{T}} \hat{L}_{T'}(\mathbf{M}) - \frac{1}{\lfloor\frac{n}{2}\rfloor}\sum_{i=1}^{\lfloor\frac{n}{2}\rfloor} l\left(\mathbf{M}, \mathbf{z}_i, \mathbf{z}_{\lfloor\frac{n}{2}\rfloor+i}\right)\right] \\
&\qquad \leq \mathop{\mathbb{E}}_{T,T'\sim\mathcal{D}_\mathcal{T}} \sup_{\mathbf{M}\in\mathcal{M}_\mathcal{S}} \left[\hat{L}_{T'}(\mathbf{M}) - \frac{1}{\lfloor\frac{n}{2}\rfloor}\sum_{i=1}^{\lfloor\frac{n}{2}\rfloor} l\left(\mathbf{M}, \mathbf{z}_i, \mathbf{z}_{\lfloor\frac{n}{2}\rfloor+i}\right)\right] \\
&\qquad\qquad \text{(U-Statistics, Lemma 7 in Cao et al. (2016).)} \\
&\qquad \leq \mathop{\mathbb{E}}_{T,T'\sim\mathcal{D}_\mathcal{T}} \sup_{\mathbf{M}\in\mathcal{M}_\mathcal{S}} \left[\frac{1}{\lfloor\frac{n}{2}\rfloor}\sum_{i=1}^{\lfloor\frac{n}{2}\rfloor} l\left(\mathbf{M}, \mathbf{z}'_i, \mathbf{z}'_{\lfloor\frac{n}{2}\rfloor+i}\right) - l\left(\mathbf{M}, \mathbf{z}_i, \mathbf{z}_{\lfloor\frac{n}{2}\rfloor+i}\right)\right] \\
&\qquad\qquad \text{(Equation (26.9) in Shalev-Shwartz and Ben-David (2014a).)} \\
&\qquad \leq \frac{1}{\lfloor\frac{n}{2}\rfloor} \mathop{\mathbb{E}}_{\substack{T,T'\sim\mathcal{D}_\mathcal{T}\\ \sigma}} \sup_{\mathbf{M}\in\mathcal{M}_\mathcal{S}} \sum_{i=1}^{\lfloor\frac{n}{2}\rfloor} \sigma_i \left(l\left(\mathbf{M}, \mathbf{z}'_i, \mathbf{z}'_{\lfloor\frac{n}{2}\rfloor+i}\right) - l\left(\mathbf{M}, \mathbf{z}_i, \mathbf{z}_{\lfloor\frac{n}{2}\rfloor+i}\right)\right) \\
&\qquad\qquad \text{(End of the proof of Lemma 26.2 in Shalev-Shwartz and Ben-David (2014a).)} \\
&\qquad \leq \frac{2}{\lfloor\frac{n}{2}\rfloor} \mathop{\mathbb{E}}_{T\sim\mathcal{D}_\mathcal{T},\sigma} \sup_{\mathbf{M}\in\mathcal{M}_\mathcal{S}} \left[\sum_{i=1}^{\lfloor\frac{n}{2}\rfloor} \sigma_i l\left(\mathbf{M}, \mathbf{z}_i, \mathbf{z}_{\lfloor\frac{n}{2}\rfloor+i}\right)\right] \\
&\qquad\qquad (k\text{-lipschitzness and Lemma 26.9 in Shalev-Shwartz and Ben-David (2014a).})
\end{aligned}$$

$$\leq \frac{2}{\lfloor \frac{n}{2} \rfloor} \mathop{\mathbb{E}}_{T\sim\mathcal{D}_\mathcal{T},\sigma} \sup_{\mathbf{M}\in\mathcal{M}_\mathcal{S}} \left[\sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} \sigma_i k k_\mathbf{M}\left(\mathbf{x}_i, \mathbf{x}_{\lfloor \frac{n}{2} \rfloor + i}\right) \right] \tag{C.8}$$

Note that $\forall \mathbf{b} \in \mathbb{R}^n, \mathbb{E}_\sigma \sum_{i=1}^n \sigma_i b_i = 0$. It implies that $\forall \mathbf{a} \in A \subseteq \mathbb{R}^n$ we have:

$$\mathop{\mathbb{E}}_{\sigma} \sup_{\mathbf{a}\in A} \sum_{i=1}^n \sigma_i a_i = \mathop{\mathbb{E}}_{\sigma} \sum_{i=1}^n \sigma_i b_i + \mathop{\mathbb{E}}_{\sigma} \sup_{\mathbf{a}\in A} \sum_{i=1}^n \sigma_i a_i$$

($\mathbf{b}$ does not depend on A.)

$$= \mathop{\mathbb{E}}_{\sigma} \sup_{\mathbf{a}\in A} \sum_{i=1}^n \sigma_i (a_i + b_i). \tag{C.9}$$

Applying (C.9) to (C.8) with $b_i = -k_{\mathbf{M}_\mathcal{S}}\left(\mathbf{x}_i, \mathbf{x}_{\lfloor \frac{n}{2} \rfloor + i}\right)$ gives:

$$\mathop{\mathbb{E}}_{T\sim\mathcal{D}_\mathcal{T}} \sup_{\mathbf{M}\in\mathcal{M}_\mathcal{S}} \left[L_\mathcal{T}(\mathbf{M}) - \hat{L}_T(\mathbf{M}) \right] \leq \frac{2}{\lfloor \frac{n}{2} \rfloor} \mathop{\mathbb{E}}_{T\sim\mathcal{D}_\mathcal{T},\sigma} \sup_{\mathbf{M}\in\mathcal{M}_\mathcal{S}} \left[\sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} \sigma_i k\, k_{\mathbf{M}-\mathbf{M}_\mathcal{S}}\left(\mathbf{x}_i, \mathbf{x}_{\lfloor \frac{n}{2} \rfloor + i}\right) \right]$$

(Definition 4.3.)

$$\leq 2k R_n(\mathcal{M}_\mathcal{S}).$$

□

C.6 Proof of Lemma 4.7

Lemma (Bound on $|R_T - R_{T^i}|$)**.** *For any positive, convex and k-lipschitz continuous (Definition 4.2) loss function, any metric satisfying Equation* (4.14) *and any algorithm of estimation error* $R_T = \sup_{\mathbf{M}\in\mathcal{M}_\mathcal{S}} \left[L_\mathcal{T}(\mathbf{M}) - \hat{L}_T(\mathbf{M}) \right]$ *we have:*

$$|R_T - R_{T^i}| \leq \frac{2G_3(\mathbf{M}_\mathcal{S}) + 2\sup_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_\mathcal{T}} \left[k \left\| g(\mathbf{x}, \mathbf{x}') \right\|_* \sqrt{\frac{G_3(\mathbf{M}_\mathcal{S})}{\lambda}} \right]}{n}$$

where $\|\cdot\|_*$ *is the dual norm of the regularization term (Definition A.4).*

Proof. First of all note that from Definition 4.2 and Equation (4.14) we have that for any two examples $\mathbf{z}, \mathbf{z}' \sim \mathcal{D}_\mathcal{T}$:

$$\left| l\left(\mathbf{M}, \mathbf{z}, \mathbf{z}'\right) - l\left(\mathbf{M}_\mathcal{S}, \mathbf{z}, \mathbf{z}'\right) \right| \leq k \left| \left\langle g(\mathbf{x}, \mathbf{x}'), \mathbf{M} - \mathbf{M}_\mathcal{S} \right\rangle \right|$$

(Cauchy-Schwartz's inequality (Theorem A.3).)

$$\Rightarrow \quad l\left(\mathbf{M}, \mathbf{z}, \mathbf{z}'\right) \leq l\left(\mathbf{M}_\mathcal{S}, \mathbf{z}, \mathbf{z}'\right) + k \left\| g(\mathbf{x}, \mathbf{x}') \right\|_* \|\mathbf{M} - \mathbf{M}_\mathcal{S}\|$$

($\mathbf{M} \in \mathcal{M}_\mathcal{S}$.)

$$\Rightarrow \quad l\left(\mathbf{M}, \mathbf{z}, \mathbf{z}'\right) \leq l\left(\mathbf{M}_\mathcal{S}, \mathbf{z}, \mathbf{z}'\right) + k \left\| g(\mathbf{x}, \mathbf{x}') \right\|_* \sqrt{\frac{G_3(\mathbf{M}_\mathcal{S})}{\lambda}}$$

(Taking the supremum over $\mathbf{z}, \mathbf{z}' \sim \mathcal{D}_{\mathcal{T}}$.)

$$\Rightarrow \qquad l\left(\mathbf{M}, \mathbf{z}, \mathbf{z}'\right) \leq G_3(\mathbf{M}_{\mathcal{S}}) + \sup_{\mathbf{z},\mathbf{z}' \sim \mathcal{D}_{\mathcal{T}}} \left[k \left\| g(\mathbf{x}, \mathbf{x}') \right\|_* \sqrt{\frac{G_3(\mathbf{M}_{\mathcal{S}})}{\lambda}} \right] \tag{C.10}$$

where $\|\cdot\|_*$ represents the dual norm of the regularization term (Definition A.4) and $G_3(\mathbf{M}_{\mathcal{S}}) = \sup_{\mathbf{z},\mathbf{z}' \sim \mathcal{D}_{\mathcal{T}}} l\left(\mathbf{M}_{\mathcal{S}}, \mathbf{z}, \mathbf{z}'\right)$.

$$\begin{aligned}
|R_T - R_{T^i}| &= \left| \sup_{\mathbf{M} \in \mathcal{M}_{\mathcal{S}}} \left[L_{\mathcal{T}}(\mathbf{M}) - \hat{L}_T(\mathbf{M}) \right] - \sup_{\mathbf{M} \in \mathcal{M}_{\mathcal{S}}} \left[L_{\mathcal{T}}(\mathbf{M}) - \hat{L}_{T^i}(\mathbf{M}) \right] \right| \\
&\leq \sup_{\mathbf{M} \in \mathcal{M}_{\mathcal{S}}} \left| \hat{L}_T(\mathbf{M}) - \hat{L}_{T^i}(\mathbf{M}) \right| \\
&= \frac{1}{n(n-1)} \sup_{\mathbf{M} \in \mathcal{M}_{\mathcal{S}}} \left| \sum_{\mathbf{z} \in T} \sum_{\substack{\mathbf{z}' \in T \\ \mathbf{z} \neq \mathbf{z}'}} l\left(\mathbf{M}, \mathbf{z}, \mathbf{z}'\right) - \sum_{\mathbf{z}^i \in T^i} \sum_{\substack{\mathbf{z}^{i\prime} \in T^i \\ \mathbf{z}^i \neq \mathbf{z}^{i\prime}}} l\left(\mathbf{M}, \mathbf{z}^i, \mathbf{z}^{i\prime}\right) \right|
\end{aligned}$$

(T and T^i only differ by one example, $\forall j \neq i, \mathbf{z}_j = \mathbf{z}_j^i$ and $\mathbf{z}'_j = \mathbf{z}_j^{i\prime}$.)

$$\begin{aligned}
&= \frac{1}{n(n-1)} \sup_{\mathbf{M} \in \mathcal{M}_{\mathcal{S}}} \left| \sum_{\substack{\mathbf{z}' \in T \\ \mathbf{z}_i \neq \mathbf{z}'}} l\left(\mathbf{M}, \mathbf{z}_i, \mathbf{z}'\right) - \sum_{\substack{\mathbf{z}' \in T^i \\ \mathbf{z}_i^i \neq \mathbf{z}'}} l\left(\mathbf{M}, \mathbf{z}_i^i, \mathbf{z}'\right) \right. \\
&\quad \left. + \sum_{\substack{\mathbf{z} \in T \\ \mathbf{z} \neq \mathbf{z}'_i}} l\left(\mathbf{M}, \mathbf{z}, \mathbf{z}'_i\right) - \sum_{\substack{\mathbf{z} \in T^i \\ \mathbf{z} \neq \mathbf{z}_i^{i\prime}}} l\left(\mathbf{M}, \mathbf{z}, \mathbf{z}_i^{i\prime}\right) \right|
\end{aligned}$$

(Triangle inequality.)

$$\begin{aligned}
&\leq \frac{1}{n(n-1)} \sup_{\mathbf{M} \in \mathcal{M}_{\mathcal{S}}} \left[\sum_{\substack{\mathbf{z}' \in T, T^i \\ \mathbf{z}_i \neq \mathbf{z}', \mathbf{z}_i^i \neq \mathbf{z}'}} \left| l\left(\mathbf{M}, \mathbf{z}_i^i, \mathbf{z}'\right) - l\left(\mathbf{M}, \mathbf{z}_i, \mathbf{z}'\right) \right| \right. \\
&\quad \left. + \sum_{\substack{\mathbf{z} \in T, T^i \\ \mathbf{z} \neq \mathbf{z}'_i, \mathbf{z} \neq \mathbf{z}_i^{i\prime}}} \left| l\left(\mathbf{M}, \mathbf{z}, \mathbf{z}_i^{i\prime}\right) - l\left(\mathbf{M}, \mathbf{z}, \mathbf{z}'_i\right) \right| \right]
\end{aligned}$$

(Positive loss and Inequality (C.10))

$$\leq \frac{2G_3(\mathbf{M}_{\mathcal{S}}) + 2\sup_{\mathbf{z},\mathbf{z}' \sim \mathcal{D}_{\mathcal{T}}} \left[k \left\| g(\mathbf{x}, \mathbf{x}') \right\|_* \sqrt{\frac{G_3(\mathbf{M}_{\mathcal{S}})}{\lambda}} \right]}{n}.$$

□

C.7 Proof of Lemma 4.9

Lemma (Bounded regularization)**.** *Let $\mathbf{M}_T$ be the optimal solution returned by Problem* (4.1) *with training set T and a positive and convex loss. We have:*

$$\|\mathbf{M}_T - \mathbf{M}_{\mathcal{S}}\| \leq \sqrt{\frac{\hat{L}_T(\mathbf{M}_{\mathcal{S}})}{\lambda}}.$$

Proof. By the convexity of the loss and the optimality of $\mathbf{M}_T$ we have:

$$\hat{L}_T(\mathbf{M}_T) + \lambda \|\mathbf{M}_T - \mathbf{M}_{\mathcal{S}}\|^2 \leq \hat{L}_T(\mathbf{M}_{\mathcal{S}}) \qquad \text{(Positive loss.)}$$

$$\Rightarrow \quad \lambda \|\mathbf{M}_T - \mathbf{M}_{\mathcal{S}}\|^2 \leq \hat{L}_T(\mathbf{M}_{\mathcal{S}})$$

$$\Rightarrow \quad \|\mathbf{M}_T - \mathbf{M}_{\mathcal{S}}\| \leq \sqrt{\frac{\hat{L}_T(\mathbf{M}_{\mathcal{S}})}{\lambda}}.$$

□

C.8 Proof of Example 4.1

Example (Positive, convex, L-lipschitz functions for dissimilarity learning)**.** *Let $f(a)$ be a positive, convex, L-lipschitz function. Given a dissimilarity (Definition 1.8) $k_{\mathbf{M}}$ parametrized by $\mathbf{M} \in \mathcal{M}$ and any two examples $\mathbf{z}, \mathbf{z}' \sim \mathcal{D}_{\mathcal{T}}$ we define a loss as:*

$$l\left(\mathbf{M}, \mathbf{z}, \mathbf{z}'\right) = f\big(\delta_{yy'}\left[k_{\mathbf{M}}\big(\mathbf{x}, \mathbf{x}'\big) - \gamma_{yy'}\right]\big) \tag{C.11}$$

where $\delta_{yy'} = 1$ if $y = y'$ and -1 otherwise and $\gamma_{yy'}$ is the desired margin between examples. This loss is:

- *Positive,*
- *Convex,*
- *k-lipschitz continuous with respect to the metric with $k = L$,*
- *k-lipschitz continuous with $k = L \sup_{\mathbf{z},\mathbf{z}' \sim \mathcal{D}_{\mathcal{T}}} \|g(\mathbf{x}, \mathbf{x}')\|_*$,*
- *(σ, m)-admissible with* $\begin{cases} \sigma = \sup_{\mathbf{z},\mathbf{z}' \sim \mathcal{D}_{\mathcal{T}}} \gamma_{yy'} \\ m = 2L \sup_{\mathbf{z},\mathbf{z}' \sim \mathcal{D}_{\mathcal{T}}} \|g(\mathbf{x}, \mathbf{x}')\|_* \left(\sqrt{\frac{\hat{L}_T(\mathbf{M}_{\mathcal{S}})}{\lambda}} + \|\mathbf{M}_{\mathcal{S}}\|\right) \end{cases}.$

Proof. First the loss is positive and convex by construction.

Then we prove that the loss function is k-lipschitz. Given two metrics $k_{\mathbf{M}}$ and $k_{\mathbf{M}'}$ we have:

$$\left|l\left(\mathbf{M}, \mathbf{z}, \mathbf{z}'\right) - l\left(\mathbf{M}', \mathbf{z}, \mathbf{z}'\right)\right| \leq \left|f\big(\delta_{yy'}[k_{\mathbf{M}}\big(\mathbf{x}, \mathbf{x}'\big) - \gamma_{yy'}]\big) - f\big(\delta_{yy'}[k_{\mathbf{M}'}\big(\mathbf{x}, \mathbf{x}'\big) - \gamma_{yy'}]\big)\right|$$

$$
\begin{aligned}
&(f \text{ is } L\text{-lipschitz.})\\
&\leq L\left|\delta_{yy'}[k_{\mathbf{M}}(\mathbf{x},\mathbf{x}')-\gamma_{yy'}]-\delta_{yy'}[k_{\mathbf{M}'}(\mathbf{x},\mathbf{x}')-\gamma_{yy'}]\right|\\
&(\delta_{yy'}\in\{-1,1\}.)\\
&\leq L\left|k_{\mathbf{M}}(\mathbf{x},\mathbf{x}')-k_{\mathbf{M}'}(\mathbf{x},\mathbf{x}')\right| \qquad \text{(C.12)}\\
&\text{(Equation (4.14) and Cauchy-Schwartz's inequality (Theorem A.3).)}\\
&\leq L\sup_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_{\mathcal{T}}}\left\|g(\mathbf{x},\mathbf{x}')\right\|_*\left\|\mathbf{M}-\mathbf{M}'\right\| \qquad \text{(C.13)}
\end{aligned}
$$

Inequalities (C.12) and (C.13) respectively prove the lipschitzness with respect to the metric and the matrix.

Lastly we show that the loss is (σ, m)-admissible. Given four examples $\mathbf{z},\mathbf{z}',\mathbf{z}'',\mathbf{z}'''\sim\mathcal{D}_{\mathcal{T}}$ and $\mathbf{M}_T$ the learned metric when learning with T we have:

$$
\begin{aligned}
&\left|l\left(\mathbf{M}_T,\mathbf{z},\mathbf{z}'\right)-l\left(\mathbf{M}_T,\mathbf{z}'',\mathbf{z}'''\right)\right|\\
&\quad\leq\left|f\left(\delta_{yy'}[k_{\mathbf{M}_T}(\mathbf{x},\mathbf{x}')-\gamma_{yy'}]\right)-f\left(\delta_{y''y'''}[k_{\mathbf{M}_T}(\mathbf{x}'',\mathbf{x}''')-\gamma_{y''y'''}]\right)\right|\\
&\quad(f \text{ is } L\text{-lipschitz.})\\
&\quad\leq L\left|\delta_{yy'}[k_{\mathbf{M}_T}(\mathbf{x},\mathbf{x}')-\gamma_{yy'}]-\delta_{y''y'''}[k_{\mathbf{M}_T}(\mathbf{x}'',\mathbf{x}''')-\gamma_{y''y'''}]\right|\\
&\quad\text{(Triangle inequality.)}\\
&\quad\leq L\left|\delta_{yy'}k_{\mathbf{M}_T}(\mathbf{x},\mathbf{x}')-\delta_{y''y'''}k_{\mathbf{M}_T}(\mathbf{x}'',\mathbf{x}''')\right|+\left|\delta_{yy'}\gamma_{yy'}-\delta_{y''y'''}\gamma_{y''y'''}\right|\\
&\quad(\delta_{yy'},\delta_{y''y'''}\in\{-1,1\}.)\\
&\quad\leq 2L\sup_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_{\mathcal{T}}}\left|k_{\mathbf{M}_T}(\mathbf{x},\mathbf{x}')\right|+\left|\delta_{yy'}-\delta_{y''y'''}\right|\sup_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_{\mathcal{T}}}\gamma_{yy'}\\
&\quad\text{(Equation (4.14) and Cauchy-Schwartz's inequality (Theorem A.3).)}\\
&\quad\leq 2L\sup_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_{\mathcal{T}}}\left\|g(\mathbf{x},\mathbf{x}')\right\|_*\left\|\mathbf{M}_T\right\|+\left|\delta_{yy'}-\delta_{y''y'''}\right|\sup_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_{\mathcal{T}}}\gamma_{yy'}\\
&\quad\text{(Triangle inequality.)}\\
&\quad\leq 2L\sup_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_{\mathcal{T}}}\left\|g(\mathbf{x},\mathbf{x}')\right\|_*\left(\left\|\mathbf{M}_T-\mathbf{M}_{\mathcal{S}}\right\|+\left\|\mathbf{M}_{\mathcal{S}}\right\|\right)+\left|\delta_{yy'}-\delta_{y''y'''}\right|\sup_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_{\mathcal{T}}}\gamma_{yy'}\\
&\quad\text{(Bounded regularization (Lemma 4.9).)}\\
&\quad\leq 2L\sup_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_{\mathcal{T}}}\left\|g(\mathbf{x},\mathbf{x}')\right\|_*\left(\sqrt{\frac{\hat{L}_T(\mathbf{M}_{\mathcal{S}})}{\lambda}}+\left\|\mathbf{M}_{\mathcal{S}}\right\|\right)+\left|\delta_{yy'}-\delta_{y''y'''}\right|\sup_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_{\mathcal{T}}}\gamma_{yy'}
\end{aligned}
$$

Setting $\sigma=\sup_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_{\mathcal{T}}}\gamma_{yy'}$ and $m=2L\sup_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_{\mathcal{T}}}\left\|g(\mathbf{x},\mathbf{x}')\right\|_*\left(\sqrt{\frac{\hat{L}_T(\mathbf{M}_{\mathcal{S}})}{\lambda}}+\left\|\mathbf{M}_{\mathcal{S}}\right\|\right)$ gives the example. □

C.9 Proof of Example 4.2

Example (Positive, convex, L-lipschitz functions for similarity learning)**.** *Let $f(a)$ be a positive, convex, L-lipschitz function. Given a similarity (Definition 1.8) $k_{\mathbf{M}}$ parametrized by*

$\mathbf{M} \in \mathcal{M}$ and any two examples $\mathbf{z}, \mathbf{z}' \sim \mathcal{D}_{\mathcal{T}}$ we define a loss as:

$$l\left(\mathbf{M}, \mathbf{z}, \mathbf{z}'\right) = f\left(1 - \delta_{yy'} \frac{k_{\mathbf{M}}(\mathbf{x}, \mathbf{x}')}{\gamma_{yy'}}\right) \tag{C.14}$$

where $\delta_{yy'} = 1$ if $y = y'$ and -1 otherwise and $\gamma_{yy'}$ is the desired margin between examples. This loss is:

- *Positive,*
- *Convex,*
- *k-lipschitz continuous with respect to the metric with $k = \frac{L}{\inf_{\mathbf{z},\mathbf{z}' \sim \mathcal{D}_{\mathcal{T}}} |\gamma_{yy'}|}$,*
- *k-lipschitz continuous with $k = \frac{L}{\inf_{\mathbf{z},\mathbf{z}' \sim \mathcal{D}_{\mathcal{T}}} |\gamma_{yy'}|} \sup_{\mathbf{z},\mathbf{z}' \sim \mathcal{D}_{\mathcal{T}}} \|g(\mathbf{x}, \mathbf{x}')\|_*$,*
- *(σ, m)-admissible with $\begin{cases} \sigma = 0 \\ m = 2\frac{L}{\inf_{\mathbf{z},\mathbf{z}' \sim \mathcal{D}_{\mathcal{T}}} |\gamma_{yy'}|} \sup_{\mathbf{z},\mathbf{z}' \sim \mathcal{D}_{\mathcal{T}}} \|g(\mathbf{x}, \mathbf{x}')\|_* \left(\sqrt{\frac{\hat{L}_T(\mathbf{M}_S)}{\lambda}} + \|\mathbf{M}_S\|\right) \end{cases}$.*

Proof. First the loss is positive and convex by construction.

Then we prove that the loss function is k-lipschitz. Given two metrics $k_{\mathbf{M}}$ and $k_{\mathbf{M}'}$ we have:

$$\begin{aligned}
\left|l\left(\mathbf{M}, \mathbf{z}, \mathbf{z}'\right) - l\left(\mathbf{M}', \mathbf{z}, \mathbf{z}'\right)\right| &\leq \left|f\left(1 - \delta_{yy'} \frac{k_{\mathbf{M}}(\mathbf{x}, \mathbf{x}')}{\gamma_{yy'}}\right) - f\left(1 - \delta_{yy'} \frac{k_{\mathbf{M}'}(\mathbf{x}, \mathbf{x}')}{\gamma_{yy'}}\right)\right| \\
&\qquad (f \text{ is } L\text{-lipschitz.}) \\
&\leq L \left|1 - \delta_{yy'} \frac{k_{\mathbf{M}}(\mathbf{x}, \mathbf{x}')}{\gamma_{yy'}} - \left(1 - \delta_{yy'} \frac{k_{\mathbf{M}'}(\mathbf{x}, \mathbf{x}')}{\gamma_{yy'}}\right)\right| \\
&\qquad (\delta_{yy'} \in \{-1, 1\}.) \\
&\leq \frac{L}{\inf_{\mathbf{z},\mathbf{z}' \sim \mathcal{D}_{\mathcal{T}}} \left|\gamma_{yy'}\right|} \left|k_{\mathbf{M}}\left(\mathbf{x}, \mathbf{x}'\right) - k_{\mathbf{M}'}\left(\mathbf{x}, \mathbf{x}'\right)\right| \qquad \text{(C.15)} \\
&\qquad \text{(Equation (4.14) and Cauchy-Schwartz's inequality (Theorem A.3).)} \\
&\leq \frac{L}{\inf_{\mathbf{z},\mathbf{z}' \sim \mathcal{D}_{\mathcal{T}}} \left|\gamma_{yy'}\right|} \sup_{\mathbf{z},\mathbf{z}' \sim \mathcal{D}_{\mathcal{T}}} \left\|g(\mathbf{x}, \mathbf{x}')\right\|_* \left\|\mathbf{M} - \mathbf{M}'\right\| \qquad \text{(C.16)}
\end{aligned}$$

Inequalities (C.15) and (C.16) respectively prove the lipschitzness with respect to the metric and the matrix.

Lastly we show that the loss is (σ, m)-admissible. Given four examples $\mathbf{z}, \mathbf{z}', \mathbf{z}'', \mathbf{z}''' \sim \mathcal{D}_{\mathcal{T}}$ and $\mathbf{M}_T$ the learned metric when learning with T we have:

$$\begin{aligned}
\left|l\left(\mathbf{M}_T, \mathbf{z}, \mathbf{z}'\right) - l\left(\mathbf{M}_T, \mathbf{z}'', \mathbf{z}'''\right)\right| &\leq \left|f\left(1 - \delta_{yy'} \frac{k_{\mathbf{M}_T}(\mathbf{x}, \mathbf{x}')}{\gamma_{yy'}}\right) - f\left(1 - \delta_{y''y'''} \frac{k_{\mathbf{M}_T}(\mathbf{x}'', \mathbf{x}''')}{\gamma_{y''y'''}}\right)\right| \\
&\qquad (f \text{ is } L\text{-lipschitz.}) \\
&\leq L \left|\delta_{yy'} \frac{k_{\mathbf{M}_T}(\mathbf{x}, \mathbf{x}')}{\gamma_{yy'}} - \delta_{y''y'''} \frac{k_{\mathbf{M}_T}(\mathbf{x}'', \mathbf{x}''')}{\gamma_{y''y'''}}\right|
\end{aligned}$$

$$(\delta_{yy'}, \delta_{y''y'''} \in \{-1, 1\}.)$$

$$\leq 2\frac{L}{\inf_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_\mathcal{T}} \left|\gamma_{yy'}\right|} \sup_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_\mathcal{T}} \left|k_{\mathbf{M}_T}\left(\mathbf{x}, \mathbf{x}'\right)\right|$$

(Equation (4.14) and Cauchy-Schwartz's inequality (Theorem A.3).)

$$\leq 2\frac{L}{\inf_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_\mathcal{T}} \left|\gamma_{yy'}\right|} \sup_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_\mathcal{T}} \left\|g(\mathbf{x}, \mathbf{x}')\right\|_* \|\mathbf{M}_T\|$$

(Triangle inequality.)

$$\leq 2\frac{L}{\inf_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_\mathcal{T}} \left|\gamma_{yy'}\right|} \sup_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_\mathcal{T}} \left\|g(\mathbf{x}, \mathbf{x}')\right\|_* \left(\|\mathbf{M}_T - \mathbf{M}_\mathcal{S}\| + \|\mathbf{M}_\mathcal{S}\|\right)$$

(Bounded regularization (Lemma 4.9).)

$$\leq 2\frac{L}{\inf_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_\mathcal{T}} \left|\gamma_{yy'}\right|} \sup_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_\mathcal{T}} \left\|g(\mathbf{x}, \mathbf{x}')\right\|_* \left(\sqrt{\frac{\hat{L}_T(\mathbf{M}_\mathcal{S})}{\lambda}} + \|\mathbf{M}_\mathcal{S}\|\right).$$

Setting $\sigma = 0$ and $m = 2\frac{L}{\inf_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_\mathcal{T}}|\gamma_{yy'}|} \sup_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_\mathcal{T}} \|g(\mathbf{x}, \mathbf{x}')\|_* \left(\sqrt{\frac{\hat{L}_T(\mathbf{M}_\mathcal{S})}{\lambda}} + \|\mathbf{M}_\mathcal{S}\|\right)$ gives the example. □

C.10 Proof of Example 4.3

Example (Positive, convex, H-smooth, B-bounded functions for dissimilarity learning)**.** *Let $f(a)$ be a positive, convex, H-smooth, B-bounded function. Given a dissimilarity (Definition 1.8) $k_\mathbf{M}$ parametrized by $\mathbf{M} \in \mathcal{M}$ and any two examples $\mathbf{z}, \mathbf{z}' \sim \mathcal{D}_\mathcal{T}$ we define a loss as:*

$$l\left(\mathbf{M}, \mathbf{z}, \mathbf{z}'\right) = f\left(\delta_{yy'}\left[k_\mathbf{M}\left(\mathbf{x}, \mathbf{x}'\right) - \gamma_{yy'}\right]\right) \tag{C.17}$$

where $\delta_{yy'} = 1$ if $y = y'$ and -1 otherwise and $\gamma_{yy'}$ is the desired margin between examples. This loss is:

- *Positive,*
- *Convex,*
- *k-lipschitz continuous with respect to the metric with $k = \sqrt{12HB}$,*
- *k-lipschitz continuous with $k = \sqrt{12HB}\sup_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_\mathcal{T}} \|g(\mathbf{x}, \mathbf{x}')\|_*$,*
- *(σ, m)-admissible with $\begin{cases}\sigma = 0\\ m = B\end{cases}$.*

Proof. First the loss is positive and convex by construction.

Then we prove that the loss function is k-lipschitz. Given two metrics $k_\mathbf{M}$ and $k_{\mathbf{M}'}$ we have:

$$\left|l\left(\mathbf{M}, \mathbf{z}, \mathbf{z}'\right) - l\left(\mathbf{M}', \mathbf{z}, \mathbf{z}'\right)\right| \leq \left|f\left(\delta_{yy'}[k_\mathbf{M}\left(\mathbf{x}, \mathbf{x}'\right) - \gamma_{yy'}]\right) - f\left(\delta_{yy'}[k_{\mathbf{M}'}\left(\mathbf{x}, \mathbf{x}'\right) - \gamma_{yy'}]\right)\right|$$

$$
\begin{aligned}
&\text{(Lemma A.1 with } f \text{ which is } H\text{-smooth and } B \text{ bounded.)}\\
&\leq \sqrt{12HB}\left|\delta_{yy'}[\gamma_{yy'} - k_{\mathbf{M}}(\mathbf{x},\mathbf{x}')] - \delta_{yy'}[\gamma_{yy'} - k_{\mathbf{M}'}(\mathbf{x},\mathbf{x}')]\right|\\
&\qquad\qquad (\delta_{yy'} \in \{-1,1\}.)\\
&\leq \sqrt{12HB}\left|k_{\mathbf{M}}(\mathbf{x},\mathbf{x}') - k_{\mathbf{M}'}(\mathbf{x},\mathbf{x}')\right| \qquad \text{(C.18)}\\
&\text{(Equation (4.14) and Cauchy-Schwartz's inequality (Theorem A.3).)}\\
&\leq \sqrt{12HB}\sup_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_{\mathcal{T}}}\left\|g(\mathbf{x},\mathbf{x}')\right\|_*\left\|\mathbf{M}-\mathbf{M}'\right\| \qquad \text{(C.19)}
\end{aligned}
$$

Inequalities (C.18) and (C.19) respectively prove the lipschitzness with respect to the metric and the matrix.

Lastly we show that the loss is (σ, m)-admissible. Given four examples $\mathbf{z}, \mathbf{z}', \mathbf{z}'', \mathbf{z}''' \sim \mathcal{D}_{\mathcal{T}}$, $\mathbf{M}_T$ the learned metric when learning with T and the fact that the loss function is positive and B-bounded we have:

$$\left|l\left(\mathbf{M}_T, \mathbf{z}, \mathbf{z}'\right) - l\left(\mathbf{M}_T, \mathbf{z}'', \mathbf{z}'''\right)\right| \leq B.$$

Setting $\sigma = 0$ and $m = B$ gives the example. □

C.11 Proof of Example 4.4

Example (Positive, convex, H-smooth, B-bounded functions for similarity learning)**.** *Let $f(a)$ be a positive, convex, H-smooth, B-bounded function. Given a similarity (Definition 1.8) $k_{\mathbf{M}}$ parametrized by $\mathbf{M} \in \mathcal{M}$ and any two examples $\mathbf{z}, \mathbf{z}' \sim \mathcal{D}_{\mathcal{T}}$ we define a loss as:*

$$l\left(\mathbf{M}, \mathbf{z}, \mathbf{z}'\right) = f\left(1 - \delta_{yy'}\frac{k_{\mathbf{M}}(\mathbf{x},\mathbf{x}')}{\gamma_{yy'}}\right) \qquad \text{(C.20)}$$

where $\delta_{yy'} = 1$ if $y = y'$ and -1 otherwise and $\gamma_{yy'}$ is the desired margin between examples. This loss is:

- *Positive,*
- *Convex,*
- *k-lipschitz continuous with respect to the metric with $k = \frac{\sqrt{12HB}}{\inf_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_{\mathcal{T}}}|\gamma_{yy'}|}$,*
- *k-lipschitz continuous with $k = \frac{\sqrt{12HB}}{\inf_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_{\mathcal{T}}}|\gamma_{yy'}|}\sup_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_{\mathcal{T}}}\|g(\mathbf{x},\mathbf{x}')\|_*$,*
- *(σ, m)-admissible with $\begin{cases}\sigma = 0\\ m = B\end{cases}$.*

Proof. First the loss is positive and convex by construction.

Then we prove that the loss function is k-lipschitz. Given two metrics $k_{\mathbf{M}}$ and $k_{\mathbf{M}'}$ we have:

$$\left|l\left(\mathbf{M}, \mathbf{z}, \mathbf{z}'\right) - l\left(\mathbf{M}', \mathbf{z}, \mathbf{z}'\right)\right| \leq \left|f\left(1 - \delta_{yy'}\frac{k_{\mathbf{M}}(\mathbf{x},\mathbf{x}')}{\gamma_{yy'}}\right) - f\left(1 - \delta_{yy'}\frac{k_{\mathbf{M}'}(\mathbf{x},\mathbf{x}')}{\gamma_{yy'}}\right)\right|$$

(Lemma A.1 with f which is H-smooth and B bounded.)

$$\leq \sqrt{12HB}\left|1-\delta_{yy'}\frac{k_{\mathbf{M}}(\mathbf{x},\mathbf{x}')}{\gamma_{yy'}}-\left(1-\delta_{yy'}\frac{k_{\mathbf{M}'}(\mathbf{x},\mathbf{x}')}{\gamma_{yy'}}\right)\right|$$

($\delta_{yy'} \in \{-1,1\}$.)

$$\leq \frac{\sqrt{12HB}}{\inf_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_\mathcal{T}}\left|\gamma_{yy'}\right|}\left|k_{\mathbf{M}}(\mathbf{x},\mathbf{x}')-k_{\mathbf{M}'}(\mathbf{x},\mathbf{x}')\right| \tag{C.21}$$

(Equation (4.14) and Cauchy-Schwartz's inequality (Theorem A.3).)

$$\leq \frac{\sqrt{12HB}}{\inf_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_\mathcal{T}}\left|\gamma_{yy'}\right|}\sup_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_\mathcal{T}}\left\|g(\mathbf{x},\mathbf{x}')\right\|_*\left\|\mathbf{M}-\mathbf{M}'\right\| \tag{C.22}$$

Inequalities (C.21) and (C.22) respectively prove the lipschitzness with respect to the metric and the matrix.

Lastly we show that the loss is (σ, m)-admissible. Given four examples $\mathbf{z},\mathbf{z}',\mathbf{z}'',\mathbf{z}''' \sim \mathcal{D}_\mathcal{T}$, $\mathbf{M}_T$ the learned metric when learning with T and the fact that the loss function is positive and B-bounded we have:

$$\left|l\left(\mathbf{M}_T,\mathbf{z},\mathbf{z}'\right)-l\left(\mathbf{M}_T,\mathbf{z}'',\mathbf{z}'''\right)\right| \leq B$$

Setting $\sigma = 0$ and $m = B$ gives the example. □

C.12 Proofs of Table 4.2

Example (Bound on the Rademacher Average)**.** *The dual norms of* $\|\cdot\|_\mathcal{F}$, $\|\cdot\|_1$, $\|\cdot\|_{2,1}$ *and* $\|\cdot\|_{Tr}$ *are respectively* $\|\cdot\|_\mathcal{F}$, $\|\cdot\|_\infty$, $\|\cdot\|_{2,\infty}$ *and* $\|\cdot\|_{Spec}$ *whose Rademacher Average is bounded:*

$$R_n(\|\cdot\|_*) \leq \frac{2\sup_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_\mathcal{T}}\|g(\mathbf{x},\mathbf{x}')\|_\mathcal{F}^2}{\sqrt{n}}. \tag{C.23}$$

Proof. The results presented here have already been proven in (Cao et al., 2016) in a slightly less general setting where $g(\mathbf{x},\mathbf{x}') = (\mathbf{x}-\mathbf{x}')(\mathbf{x}-\mathbf{x}')^T$. We recall the proof below for the sake of completeness.

For the Frobenius norm, from the definition of Rademacher Averages we have:

$$R_n(\|\cdot\|_\mathcal{F}) = \mathop{\mathbb{E}}_{T\sim\mathcal{D}_\mathcal{T}}\frac{1}{\lfloor\frac{n}{2}\rfloor}\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left\|\sum_{i=1}^{\lfloor\frac{n}{2}\rfloor}\sigma_i g(\mathbf{x}_i,\mathbf{x}_{\lfloor\frac{n}{2}\rfloor+i})\right\|_\mathcal{F}$$

(Jensen's inequality (Theorem A.4).)

$$\leq \mathop{\mathbb{E}}_{T\sim\mathcal{D}_\mathcal{T}}\frac{1}{\lfloor\frac{n}{2}\rfloor}\sqrt{\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left\|\sum_{i=1}^{\lfloor\frac{n}{2}\rfloor}\sigma_i g(\mathbf{x}_i,\mathbf{x}_{\lfloor\frac{n}{2}\rfloor+i})\right\|_\mathcal{F}^2}$$

(Definition of $\|\cdot\|_\mathcal{F}$.)

$$\leq \mathop{\mathbb{E}}_{T\sim\mathcal{D}_\mathcal{T}} \frac{1}{\lfloor \frac{n}{2} \rfloor} \sqrt{\mathop{\mathbb{E}}_{\boldsymbol{\sigma}} \sum_{j,k} \left(\sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} \sigma_i g(\mathbf{x}_i, \mathbf{x}_{\lfloor \frac{n}{2} \rfloor + i}) \right)_{j,k}^2}$$

(Standard properties on Rademacher Variables.)

$$\leq \mathop{\mathbb{E}}_{T\sim\mathcal{D}_\mathcal{T}} \frac{1}{\lfloor \frac{n}{2} \rfloor} \sqrt{\mathop{\mathbb{E}}_{\boldsymbol{\sigma}} \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} \sum_{j,k} \left(\sigma_i g(\mathbf{x}_i, \mathbf{x}_{\lfloor \frac{n}{2} \rfloor + i}) \right)_{j,k}^2}$$

(Definition of $\|\cdot\|_\mathcal{F}$.)

$$\leq \mathop{\mathbb{E}}_{T\sim\mathcal{D}_\mathcal{T}} \frac{1}{\lfloor \frac{n}{2} \rfloor} \sqrt{\sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} \left\| g(\mathbf{x}_i, \mathbf{x}_{\lfloor \frac{n}{2} \rfloor + i}) \right\|_\mathcal{F}^2}$$

$$\leq \mathop{\mathbb{E}}_{T\sim\mathcal{D}_\mathcal{T}} \frac{1}{\lfloor \frac{n}{2} \rfloor} \sqrt{\left\lfloor \frac{n}{2} \right\rfloor \sup_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_\mathcal{T}} \|g(\mathbf{x}, \mathbf{x}')\|_\mathcal{F}^2}$$

$$\leq \frac{2 \sup_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_\mathcal{T}} \|g(\mathbf{x}, \mathbf{x}')\|_\mathcal{F}}{\sqrt{n}}.$$

Noting that the ℓ_∞ norm, the $\ell_{2,\infty}$ norm and the spectral norm are always smaller than the Frobenius norm and that the Rademacher Average is increasing when the value of the norm is increasing gives the example. □

Appendix D

Proofs of Chapter 5

D.1 Proof of Theorem 5.1

Theorem (Optimal solution of Problem (5.1))**.** *The optimal solution of Problem* (5.1) *can be found in closed form. Furthermore, we can derive two equivalent solutions:*

$$\mathbf{L}_V = \left(\mathbf{X}^T\mathbf{X} + \lambda n\mathbf{I}\right)^{-1}\mathbf{X}^T\mathbf{V} \tag{5.4}$$

$$\Leftrightarrow \qquad \mathbf{L}_V = \mathbf{X}^T\left(\mathbf{X}\mathbf{X}^T + \lambda n\mathbf{I}\right)^{-1}\mathbf{V}. \tag{5.5}$$

Proof. Problem (5.1) is a classic regularized regression problem admitting a closed form solution Cortes et al. (2007). We recall the derivation here for the sake of completeness. Let $F_V(\mathbf{L}) = \hat{L}_V(\mathbf{L}) + \lambda \|\mathbf{L}\|_{\mathcal{F}}^2$ be the function optimised in Problem (5.1). First we consider its derivative with respect to $\mathbf{L}$:

$$\frac{\partial F_V(\mathbf{L})}{\partial \mathbf{L}} = 2\left(\frac{1}{n}\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}\right)\mathbf{L} - \frac{2}{n}\mathbf{X}^T\mathbf{V}.$$

Then we set this derivative to zero to obtain:

$$\mathbf{L}_V = \left(\mathbf{X}^T\mathbf{X} + \lambda n\mathbf{I}\right)^{-1}\mathbf{X}^T\mathbf{V}.$$

Finally Equation (5.5) comes from using Taylor expansions as proposed in Cortes et al. (2007). □

D.2 Proof of Lemma 5.1

Before proving Lemma 5.1 we need the following technical lemma showing that the Frobenius norm of the optimal solution of Problem (5.1) is bounded.

Lemma D.1 (Bounded Frobenius norm)**.** *Let* $\mathbf{L}_V$ *be an optimal solution of Problem* (5.1)*, we have:*

$$\|\mathbf{L}_V\|_{\mathcal{F}} \leq \frac{B_{\mathbf{v}}}{\sqrt{\lambda}}.$$

Proof. Since $\mathbf{L}$ is an optimal solution of Problem (5.1) and by convexity of the loss we have:

$$\hat{L}_V(\mathbf{L}_V) + \lambda \|\mathbf{L}_V\|_{\mathcal{F}}^2 \leq \hat{L}_V(\mathbf{0}) + \lambda \|\mathbf{0}\|_{\mathcal{F}}^2$$

$$\Leftrightarrow \quad \frac{1}{n} \sum_{(\mathbf{x},\mathbf{v})\in V} l\left(\mathbf{L}_V, (\mathbf{x},\mathbf{v})\right) + \lambda \|\mathbf{L}_V\|_{\mathcal{F}}^2 \leq \frac{1}{n} \sum_{(\mathbf{x},\mathbf{v})\in V} l\left(\mathbf{0}, (\mathbf{x},\mathbf{v})\right) + \lambda \|\mathbf{0}\|_{\mathcal{F}}^2$$

(Positive loss.)

$$\Rightarrow \quad \lambda \|\mathbf{L}_V\|_{\mathcal{F}}^2 \leq \frac{1}{n} \sum_{(\mathbf{x},\mathbf{v})\in V} \|\mathbf{v}\|_2^2$$

($\|\mathbf{v}\|_2 \leq B_{\mathbf{v}}$.)

$$\Rightarrow \quad \lambda \|\mathbf{L}_V\|_{\mathcal{F}}^2 \leq B_{\mathbf{v}}^2$$

$$\Rightarrow \quad \|\mathbf{L}_V\|_{\mathcal{F}} \leq \frac{B_{\mathbf{v}}}{\sqrt{\lambda}}.$$

□

Lemma (Bounded loss function)**.** *Let $\mathbf{L}_V$ be the metric learned with Problem* (5.1) *with training set V, we have that for any example $(\mathbf{x},\mathbf{v}) \sim \mathcal{D}_{\mathcal{V}}$:*

$$l\left(\mathbf{L}_V, (\mathbf{x},\mathbf{v})\right) \leq B$$

with $B = B_{\mathbf{v}}^2 \left(1 + \frac{B_{\mathbf{x}}}{\sqrt{\lambda}}\right)^2$.

Proof.

$$l\left(\mathbf{L}, (\mathbf{x},\mathbf{v})\right) = \left\|\mathbf{x}^T\mathbf{L} - \mathbf{v}^T\right\|_2^2$$

(Triangle inequality and standard norm properties.)

$$\leq \left(\left\|\mathbf{x}^T\right\|_2 \|\mathbf{L}\|_{\mathcal{F}} + \left\|\mathbf{v}^T\right\|_2\right)^2$$

($\|\mathbf{v}\|_2 \leq B_{\mathbf{v}}$, $\|\mathbf{x}\|_2 \leq B_{\mathbf{x}}$ and Lemma D.1.)

$$\leq \left(B_{\mathbf{x}} \frac{B_{\mathbf{v}}}{\sqrt{\lambda}} + B_{\mathbf{v}}\right)^2$$

$$\leq B_{\mathbf{v}}^2 \left(1 + \frac{B_{\mathbf{x}}}{\sqrt{\lambda}}\right)^2.$$

Setting $B = B_{\mathbf{v}}^2 \left(1 + \frac{B_{\mathbf{x}}}{\sqrt{\lambda}}\right)^2$ gives the lemma. □

D.3 Proof of Lemma 5.2

Lemma (k-lipschitz continuity)**.** *Our loss is k-lipschitz with $k = 2B_{\mathbf{v}}B_{\mathbf{x}}\left(1 + \frac{B_{\mathbf{x}}}{\sqrt{\lambda}}\right)$.*

Proof.

$$\left|\left\|\mathbf{x}^T\mathbf{L} - \mathbf{v}^T\right\|_2^2 - \left\|\mathbf{x}^T\mathbf{L}' - \mathbf{v}^T\right\|_2^2\right|$$

$$
\begin{aligned}
&= \left| \left\| \mathbf{x}^T\mathbf{L} - \mathbf{v}^T \right\|_2 - \left\| \mathbf{x}^T\mathbf{L}' - \mathbf{v}^T \right\|_2 \right| \left| \left\| \mathbf{x}^T\mathbf{L} - \mathbf{v}^T \right\|_2 + \left\| \mathbf{x}^T\mathbf{L}' - \mathbf{v}^T \right\|_2 \right| \\
&\qquad\qquad \text{(Triangle inequality.)} \\
&\leq \left\| \mathbf{x}^T\mathbf{L} - \mathbf{v}^T - \mathbf{x}^T\mathbf{L}' + \mathbf{v}^T \right\|_2 \left| \left\| \mathbf{x}^T\mathbf{L} - \mathbf{v}^T \right\|_2 + \left\| \mathbf{x}^T\mathbf{L}' - \mathbf{v}^T \right\|_2 \right| \\
&\qquad\qquad \text{(Bounded loss (Lemma 5.1).)} \\
&\leq \left\| \mathbf{L} - \mathbf{L}' \right\|_{\mathcal{F}} 2B_{\mathbf{v}}B_{\mathbf{x}} \left(1 + \frac{B_{\mathbf{x}}}{\sqrt{\lambda}} \right).
\end{aligned}
$$

Setting $k = 2B_{\mathbf{v}}B_{\mathbf{x}} \left(1 + \frac{B_{\mathbf{x}}}{\sqrt{\lambda}} \right)$ gives the lemma. □

D.4 Proof of Lemma 5.3

To prove Lemma 5.3 we need the following technical lemma.

Lemma D.2. *Let $F_V(\mathbf{L}) = \hat{L}_V(\mathbf{L}) + \lambda \left\| \mathbf{L} \right\|_{\mathcal{F}}^2$ and $F_V^i(\mathbf{L}) = \hat{L}_V(\mathbf{L}) + \lambda \left\| \mathbf{L} \right\|_{\mathcal{F}}^2$ be the functions minimized in Problem* (5.1) *where V and V^i are two training samples of n examples. V^i is obtained by replacing example i from V by another example drawn independently from $\mathcal{D}_{\mathcal{V}}$. Let $\mathbf{L}_V$ and $\mathbf{L}_{V^i}$ be their respective minimizers, and λ be the regularization parameter used in our algorithm. Let $\Delta_{\mathbf{L}} = \mathbf{L}_V - \mathbf{L}_{V^i}$, then, we have, for any $t \in [0, 1]$,*

$$
\left\| \mathbf{L}_V \right\|_{\mathcal{F}}^2 - \left\| \mathbf{L}_V - t\Delta_{\mathbf{L}} \right\|_{\mathcal{F}}^2 + \left\| \mathbf{L}_{V^i} \right\|_{\mathcal{F}}^2 - \left\| \mathbf{L}_{V^i} + t\Delta_{\mathbf{L}} \right\|_{\mathcal{F}}^2 \leq \frac{4tB_{\mathbf{v}}B_{\mathbf{x}}}{\lambda n} \left(1 + \frac{B_{\mathbf{x}}}{\sqrt{\lambda}} \right) \left\| \Delta_{\mathbf{L}} \right\|_{\mathcal{F}}. \quad \text{(D.1)}
$$

Proof. This proof is similar to the proof of Lemma 20 in Bousquet and Elisseeff (2002a) which we recall here for the sake of completeness. First, note that $\hat{L}_V$ is a convex function, thus, for any $t \in [0, 1]$, we have:

$$
\hat{L}_{V^i}(\mathbf{L}_V - t\Delta_{\mathbf{L}}) - \hat{L}_{V^i}(\mathbf{L}_V) \leq t(\hat{L}_{V^i}(\mathbf{L}_{V^i}) - \hat{L}_{V^i}(\mathbf{L}_V)) \quad \text{(D.2)}
$$

$$
\hat{L}_{V^i}(\mathbf{L}_{V^i} + t\Delta_{\mathbf{L}}) - \hat{L}_{V^i}(\mathbf{L}_{V^i}) \leq t(\hat{L}_{V^i}(\mathbf{L}_V) - \hat{L}_{V^i}(\mathbf{L}_{V^i})) \quad \text{(D.3)}
$$

Summing Inequalities (D.2) and (D.3) gives:

$$
\hat{L}_{V^i}(\mathbf{L}_V - t\Delta_{\mathbf{L}}) - \hat{L}_{V^i}(\mathbf{L}_V) + \hat{L}_{V^i}(\mathbf{L}_{V^i} + t\Delta_{\mathbf{L}}) - \hat{L}_{V^i}(\mathbf{L}_{V^i}) \leq 0 \quad \text{(D.4)}
$$

$\mathbf{L}_V$ and $\mathbf{L}_{V^i}$ respectively minimize F_V and $F_{V^i}(\mathbf{L})$, we have:

$$
F_V(\mathbf{L}_V) - F_V(\mathbf{L}_V - t\Delta_{\mathbf{L}}) \leq 0 \quad \text{(D.5)}
$$

$$
F_{V^i}(\mathbf{L}_{V^i}) - F_{V^i}(\mathbf{L}_{V^i} + t\Delta_{\mathbf{L}}) \leq 0 \quad \text{(D.6)}
$$

Summing Inequalities (D.4), (D.5) and (D.6) gives:

$$
\begin{aligned}
&\hat{L}_{V^i}(\mathbf{L}_V - t\Delta_{\mathbf{L}}) - \hat{L}_{V^i}(\mathbf{L}_V) + \hat{L}_V(\mathbf{L}_V) - \hat{L}_V(\mathbf{L}_V - t\Delta_{\mathbf{L}}) \\
&\qquad + \lambda \left\| \mathbf{L}_V \right\|_{\mathcal{F}}^2 - \lambda \left\| \mathbf{L}_V - t\Delta_{\mathbf{L}} \right\|_{\mathcal{F}}^2 + \lambda \left\| \mathbf{L}_{V^i} \right\|_{\mathcal{F}}^2 - \lambda \left\| \mathbf{L}_{V^i} + t\Delta_{\mathbf{L}} \right\|_{\mathcal{F}}^2 \leq 0. \quad \text{(D.7)}
\end{aligned}
$$

From Equation (D.7), we can write:

$$
\lambda \left\| \mathbf{L}_V \right\|_{\mathcal{F}}^2 - \lambda \left\| \mathbf{L}_V - t\Delta_{\mathbf{L}} \right\|_{\mathcal{F}}^2 + \lambda \left\| \mathbf{L}_{V^i} \right\|_{\mathcal{F}}^2 - \lambda \left\| \mathbf{L}_{V^i} + t\Delta_{\mathbf{L}} \right\|_{\mathcal{F}}^2 \leq C \quad \text{(D.8)}
$$

with

$$C = \hat{L}_{V^i}(\mathbf{L}_V) - \hat{L}_{V^i}(\mathbf{L}_V - t\Delta_{\mathbf{L}}) + \hat{L}_V(\mathbf{L}_V - t\Delta_{\mathbf{L}}) - \hat{L}_V(\mathbf{L}_V).$$

Using Lemma 5.2 we can bound C:

$$\begin{aligned}
C &\leq \left| \hat{L}_{V^i}(\mathbf{L}_V) - \hat{L}_{V^i}(\mathbf{L}_V - t\Delta_{\mathbf{L}}) + \hat{L}_V(\mathbf{L}_V - t\Delta_{\mathbf{L}}) - \hat{L}_V(\mathbf{L}_V) \right| \\
&= \left| \frac{1}{n} \sum_{(\mathbf{x},\mathbf{v})\in V} l\left(\mathbf{L}_V - t\Delta_{\mathbf{L}}, (\mathbf{x},\mathbf{v})\right) - \frac{1}{n} \sum_{(\mathbf{x}^i,\mathbf{v}^i)\in V^i} l\left(\mathbf{L}_V - t\Delta_{\mathbf{L}}, (\mathbf{x}^i,\mathbf{v}^i)\right) \right. \\
&\quad \left. + \frac{1}{n} \sum_{(\mathbf{x}^i,\mathbf{v}^i)\in V^i} l\left(\mathbf{L}_V, (\mathbf{x}^i,\mathbf{v}^i)\right) - \frac{1}{n} \sum_{(\mathbf{x},\mathbf{v})\in V} l\left(\mathbf{L}_V, (\mathbf{x},\mathbf{v})\right) \right| \\
&\hspace{20em} \text{(V and V^i only differ by one example.)} \\
&= \frac{1}{n} \left| l\left(\mathbf{L}_V - t\Delta_{\mathbf{L}}, (\mathbf{x}_i,\mathbf{v}_i)\right) - l\left(\mathbf{L}_V - t\Delta_{\mathbf{L}}, (\mathbf{x}_i^i,\mathbf{v}_i^i)\right) + l\left(\mathbf{L}_V, (\mathbf{x}_i^i,\mathbf{v}_i^i)\right) - l\left(\mathbf{L}_V, (\mathbf{x}_i,\mathbf{v}_i)\right) \right| \\
&\hspace{20em} \text{(Triangle inequality.)} \\
&\leq \frac{1}{n} \left| l\left(\mathbf{L}_V - t\Delta_{\mathbf{L}}, (\mathbf{x}_i,\mathbf{v}_i)\right) - l\left(\mathbf{L}_V, (\mathbf{x}_i,\mathbf{v}_i)\right) \right| + \frac{1}{n} \left| l\left(\mathbf{L}_V, (\mathbf{x}_i^i,\mathbf{v}_i^i)\right) - l\left(\mathbf{L}_V - t\Delta_{\mathbf{L}}, (\mathbf{x}_i^i,\mathbf{v}_i^i)\right) \right| \\
&\hspace{20em} \text{(Loss k-lipschitz (Lemma 5.2).)} \\
&\leq \frac{4tB_{\mathbf{v}}B_{\mathbf{x}}}{n} \left(1 + \frac{B_{\mathbf{x}}}{\sqrt{\lambda}}\right) \left\|\Delta_{\mathbf{L}}\right\|_{\mathcal{F}}.
\end{aligned}$$

□

We can now prove the lemma.

Lemma (Uniform stability)**.** *Our algorithm has a uniform stability in* $\beta = \frac{8B_{\mathbf{v}}^2B_{\mathbf{x}}^2}{\lambda n}\left(1 + \frac{B_{\mathbf{x}}}{\sqrt{\lambda}}\right)^2$.

Proof. By setting $t = \frac{1}{2}$ in Lemma D.2, one can obtain for the left hand side:

$$\left\|\mathbf{L}_V\right\|_{\mathcal{F}}^2 - \left\|\mathbf{L}_V - \frac{1}{2}\Delta_{\mathbf{L}}\right\|_{\mathcal{F}}^2 + \left\|\mathbf{L}_{V^i}\right\|_{\mathcal{F}}^2 - \left\|\mathbf{L}_{V^i} + \frac{1}{2}\Delta_{\mathbf{L}}\right\|_{\mathcal{F}}^2 = \frac{1}{2}\left\|\Delta_{\mathbf{L}}\right\|_{\mathcal{F}}^2$$

and thus:

$$\begin{aligned}
& & \frac{1}{2}\left\|\Delta_{\mathbf{L}}\right\|_{\mathcal{F}}^2 &\leq \frac{2B_{\mathbf{v}}B_{\mathbf{x}}}{\lambda n}\left(1 + \frac{B_{\mathbf{x}}}{\sqrt{\lambda}}\right)\left\|\Delta_{\mathbf{L}}\right\|_{\mathcal{F}} \\
&\Rightarrow & \left\|\Delta_{\mathbf{L}}\right\|_{\mathcal{F}} &\leq \frac{4B_{\mathbf{v}}B_{\mathbf{x}}}{\lambda n}\left(1 + \frac{B_{\mathbf{x}}}{\sqrt{\lambda}}\right)
\end{aligned}$$

From Lemma 5.2 we have:

$$\begin{aligned}
\left| l\left(\mathbf{L}_V, (\mathbf{x},\mathbf{v})\right) - l\left(\mathbf{L}_{V^i}, (\mathbf{x},\mathbf{v})\right) \right| &\leq 2B_{\mathbf{v}}B_{\mathbf{x}}\left(1 + \frac{B_{\mathbf{x}}}{\sqrt{\lambda}}\right)\left\|\Delta_{\mathbf{L}}\right\|_{\mathcal{F}} \\
&\leq \frac{8B_{\mathbf{v}}^2B_{\mathbf{x}}^2}{\lambda n}\left(1 + \frac{B_{\mathbf{x}}}{\sqrt{\lambda}}\right)^2
\end{aligned}$$

Setting $\beta = \frac{8B_{\mathbf{v}}^2B_{\mathbf{x}}^2}{\lambda n}\left(1 + \frac{B_{\mathbf{x}}}{\sqrt{\lambda}}\right)^2$ gives the lemma. □

Appendix E

French Translations

Table des Matières

E.1 Introduction

L'*Apprentissage Automatique* est un domaine de l'*Intelligence Artificielle* dont le but est d'acquérir de nouvelles connaissances à partir de données. Ces nouvelles connaissances prennent généralement la forme d'un modèle, appris à partir d'un nombre limité d'exemples observés et capable de bien généraliser à de futures requêtes. En d'autres termes, le but est d'apprendre comment résoudre un problème de façon automatique à partir d'un nombre fini d'observations. Par exemple, l'objectif de la détection de spams est d'utiliser la boite mail annotée d'un utilisateur pour apprendre comment séparer les emails désirés des autres; en suivi d'objets le problème est de suivre un élément donné dans une vidéo; en reconnaissance de visages le but est d'identifier une personne dans un ensemble d'images.... La diversité des problèmes posés en apprentissage automatique a attiré beaucoup d'attention dans le passé et mérite que l'on continue à s'y intéresser activement.

Dans cette thèse nous sommes principalement intéressés par les problèmes d'*Apprentissage Supervisé*. L'idée derrière ce paradigme est que les exemples sont accompagnés d'une étiquette. Celle-ci peut être une valeur ou une classe et correspond à la solution du problème pour l'exemple associé. Pour illustrer cela, nous considérons le problème de prédiction du prix des habitations et celui de la reconnaissance de champignons empoisonnés. Dans le premier cas, le but est de prédire le prix d'une maison, chaque exemple correspond alors à un ensemble de caractéristiques du bâtiment tandis que l'étiquette correspond à son prix. Dans le second cas, le but est de reconnaitre, à partir d'images, les champignons mangeables de ceux qui sont empoisonnés. Chaque exemple est alors la photo d'un champignon tandis que l'étiquette correspond à sa classe, i.e. empoissonné ou pas. De ces exemples, nous remarquons l'importance, en apprentissage supervisé, de la généralisation aux nouvelles données. En effet, les étiquettes des exemples d'apprentissage étant données, l'intérêt d'un modèle qui n'est pas capable de

prédire la bonne étiquette pour un nouvel exemple est limité. Notons que les deux exemples précédents correspondent à des problèmes largement étudiés en apprentissage supervisé : la régression et la classification. La différence entre les deux est que le but du premier est de prédire une valeur continue tandis que l'objectif du second est de déterminer une classe discrète.

L'apprentissage supervisé n'est pas le seul paradigme existant en apprentissage automatique. Il peut, en fait, être opposé à celui de l'apprentissage non supervisé où les exemples ne sont pas associés à des étiquettes. Par exemple un problème largement étudié est celui du partitionnement de données où l'objectif est d'obtenir une séparation sensée de l'espace, c'est à dire de regrouper les exemples qui partagent des propriétés communes. La performance des algorithmes d'apprentissage non supervisé est difficile à évaluer en pratique. En effet, contrairement à l'apprentissage supervisé, il n'y a pas d'étiquettes donnant un retour évident sur le modèle appris.

S'inspirant de ces deux paradigmes, l'idée derrière l'*Apprentissage Semi-Supervisé* est de considérer deux ensembles d'apprentissage, le premier est étiqueté tandis que le second ne l'est pas. Dans ce cas, le but est souvent d'utiliser les exemples étiquetés pour aider à résoudre une tâche d'apprentissage non supervisé ou d'utiliser les exemples non étiquetés pour aider à résoudre une tâche d'apprentissage supervisé.

Jusque là nous avons considéré que le but des différentes approches d'apprentissage automatique est de résoudre une tâche unique. Prenant un point de vue différent, l'idée derrière l'*Apprentissage par Transfert* est de transférer la connaissance apprise sur un problème source à un problème cible. De façon similaire, l'idée derrière l'*Adaptation de Domaine* est de transférer le modèle appris sur une tâche source pour résoudre un problème cible qui est différent mais relié. Par exemple, dans le problème de la détection de spams, les deux tâches peuvent être de détecter les mails non désirés dans les boîtes de deux utilisateurs différents. Ces deux utilisateurs rencontrent le même problème mais la distribution de leurs mails diffère, e.g. ils ne sont pas abonnés aux mêmes listes de diffusion. Dans ce cas le but est d'adapter le modèle appris pour un des utilisateurs à l'autre.

Dans ce manuscrit nous verrons que même si nous nous intéressons principalement à des problèmes d'apprentissage supervisé, plusieurs de nos contributions sont aussi liées aux différents paradigmes présentés ici.

Lors de la présentation du paradigme de l'apprentissage supervisé nous avons insisté sur le fait qu'un modèle, appris en utilisant un nombre limité d'exemples d'apprentissage, devrait pouvoir généraliser à de nouveaux exemples. Une façon de vérifier cette propriété est d'évaluer le modèle appris sur un nouvel ensemble d'exemples de test indépendants des exemples d'apprentissage et pour lesquels la solution au problème est connue. Cependant, le nombre d'exemples qui peuvent être obtenus est souvent limité. Cela implique que cette approche ne suffit bien souvent pas à assurer que le modèle généralise bien. D'autres méthodes sont alors nécessaires. Pour cela, notons d'abord qu'une supposition commune en apprentissage automatique est que la tâche que nous cherchons à résoudre est complètement définie

par une distribution inconnue à partir de laquelle les exemples d'apprentissage sont tirés. Par suite, une solution possible est d'utiliser une procédure de validation croisée où l'idée est de séparer l'ensemble d'apprentissage en k parties. Le modèle est alors appris sur $k-1$ parties et testé sur la dernière. Cette procédure est alors répétée k fois, i.e. jusqu'à ce que chaque partie est été utilisée comme partie de test, et la précision est obtenue comme moyenne des différentes répétitions. Cette procédure requiert aussi un nombre significatif d'exemples pour être pertinente. Une autre possibilité qui suit de la supposition évoquée précédemment consiste à procéder à une analyse théorique de l'algorithme d'apprentissage et de dériver des bornes appelées bornes en généralisation. L'idée de ces bornes est de montrer que l'erreur réelle du modèle appris, i.e. son erreur sur la distribution inconnue, est bornée par son erreur empirique, i.e. son erreur sur l'ensemble d'apprentissage, à laquelle s'ajoute un terme qui décroit avec l'augmentation de la taille de l'ensemble d'apprentissage. L'obtention de telles bornes est une garantie que les modèles appris par l'algorithme concerné généralisent raisonnablement bien.

De nombreuses approches ont été proposées pour résoudre les problèmes posés par l'apprentissage supervisé. Parmi celles-ci plusieurs dépendent fortement d'une notion de distance ou de similarité entre les exemples pour apprendre un modèle. Un exemple très représentatif est le classificateur des plus proches voisins qui est basé sur l'idée que deux exemples similaires devraient partager la même étiquette. Un autre exemple est l'algorithme des machines à vecteurs de support qui propose de classer les exemples en fonction de leur similarité à des points spécifiques nommés vecteurs de support. Dans ces deux exemples la notion de similarité utilisée est d'une importance critique. Cependant des tâches différentes requièrent souvent des mesures de similarité différentes. Par exemple, considérant les exemples évoqués précédemment dans cette introduction, il semble mal venu de comparer les habitations et les champignons de la même façon. Manuellement choisir une mesure de similarité appropriée peut être fastidieux et difficile. Cependant il devrait être possible de l'inférer de façon automatique à partir des données. C'est l'idée derrière l'*Apprentissage de Métriques* qui correspond au problème auquel nous allons nous intéresser dans cette thèse.

Nous identifions plusieurs limites des approches actuelles en apprentissage de métriques. Tout d'abord plusieurs méthodes proposent d'utiliser des informations supplémentaires pour aider durant le processus d'apprentissage. Cependant il n'y a pas de compréhension théorique de l'impact de ces informations sur la métrique apprise. Ensuite les propriétés intrinsèques des métriques apprises sont souvent les mêmes. En effet celles-ci sont généralement apprises avec l'idée de rapprocher les exemples similaires et d'éloigner les exemples dissimilaires. Dans certains cas il pourrait être intéressant de considérer d'autres types de contraintes. Un exemple est l'obtention d'une métrique dont le comportement n'est pas limité aux exemples mais est plus global dans le sens où elle est, par exemple, capable de bouger des blocs d'exemples en tant que tels. Enfin, une troisième limite des approches actuelles est qu'elles ne sont, bien souvent, pas justifiées théoriquement, i.e. aucune garantie n'est proposée concernant la

capacité à généraliser des métriques apprises.

Contributions: Apprendre des Métriques avec un Comportement Contrôlé

Dans cette thèse nous proposons plusieurs approches pour apprendre des métriques dont le comportement est contrôlé. Dans une première partie nous proposons d'utiliser une information supplémentaire qui prend la forme d'une métrique de référence ou métrique source pour guider de façon stricte ou plus relâchée la métrique apprise. Ainsi, dans notre première contribution, nous nous intéressons au problème de la régression des valeurs d'une métrique de référence uniquement accessible à travers un ensemble d'apprentissage de taille limité. Dans notre seconde contribution nous étudions de façon théorique comment utiliser une métrique de référence venant d'un problème lié mais différent peut aider lors du processus d'apprentissage. En particulier nous dérivons plusieurs mesures de l'apport de la métrique source pour le problème considéré. Dans une seconde partie nous proposons deux approches capables de considérer de nouvelles formes de contraintes pour l'apprentissage de métriques. Ainsi, dans notre troisième contribution, nous considérons que les exemples d'apprentissage ne devraient pas bouger les uns par rapport aux autres mais plutôt par rapport à des points virtuels qui se trouvent déjà dans l'espace induit par la métrique apprise. Cette méthode nous permet de contrôler de manière précise le mouvement de chaque exemple. Dans notre quatrième contribution nous étendons notre troisième contribution et considérons de récentes avancées dans le domaine du Transport Optimal pour proposer une nouvelle approche pour apprendre une métrique capable de bouger des blocs d'exemples dans l'espace. Enfin, remarquons que dans cette thèse nous proposons, autant que possible, des approches qui sont justifiées théoriquement.

Plan général

Dans la première partie de cette thèse nous présentons plusieurs éléments préliminaires. Dans le premier chapitre nous introduisons des concepts qui seront utilisés tout au long de ce manuscrit tandis que dans le second chapitre nous proposons une revue de l'état de l'art en apprentissage de métriques.

Chapitre 1 Le premier chapitre de cette thèse est dédié à la présentation de plusieurs notions et outils utilisés dans celle-ci. La première partie de ce chapitre présente le cadre de travail de la minimisation du risque sur lequel sont basées toutes nos contributions algorithmiques. La seconde partie est dédiée à l'analyse théorique des algorithmes. Plus précisément nous présentons deux cadres théoriques utilisés pour dériver des bornes en généralisation et basés respectivement sur la stabilité uniforme et la complexité de Rademacher. La troisième partie s'intéresse à la notion de fonction de perte et de terme de régularisation qui sont des éléments clés de l'apprentissage par minimisation du risque. Au travers de plusieurs exemples nous montrons qu'il existe de nombreux choix avec des propriétés différentes. Cette troisième partie s'intéresse aussi à la présentation d'une définition formelle de la notion de métrique

comme terme général pour désigner une similarité, une dissimilarité ou une distance. De la même façon que pour les fonctions de perte et les termes de régularisation plusieurs exemples sont présentés. La dernière partie de ce premier chapitre introduit plusieurs autres notions utiles telles que le classificateur des plus proches voisins, que nous utiliserons souvent avec les métriques que nous apprenons, et le problème de l'adaptation de domaine que nous utilisons pour évaluer deux de nos contributions.

Chapitre 2 Le second chapitre de cette thèse correspond à une revue de l'état de l'art en apprentissage de métriques. Dans celle-ci nous présentons les approches principales qui ont fait le succès de ce domaine. Nous proposons de diviser cette revue en quatre parties qui correspondent aux réponses à quatre questions basiques sur les problèmes de l'apprentissage de métriques. Dans la première partie nous considérons les différentes sortes de métriques pouvant être apprises. Ensuite, dans la seconde partie, nous répondons à la question de savoir comment ces métriques peuvent être apprises en pratique. Dans la troisième partie de ce chapitre nous présentons plusieurs approches qui s'intéressent aux questions théoriques liées à l'apprentissage de métriques. Enfin, dans la dernière partie, nous présentons plusieurs travaux qui s'intéressent à l'utilisation de l'apprentissage de métriques dans des applications qui vont de la classification au partitionnement en passant par l'adaptation de domaine.

Dans la seconde partie de cette thèse nous présentons nos deux premières contributions. Elles s'intéressent à l'utilisation d'une métrique de référence comme aide lors du processus d'apprentissage.

Chapitre 3 Dans le troisième chapitre de cette thèse nous présentons notre première contribution. Elle correspond à une méthode d'apprentissage capable d'approximer une métrique existante. La première partie de ce chapitre est dédiée à la présentation du problème d'optimisation qui correspond à une régression des valeurs d'une métrique. De plus nous montrons que quand la métrique de référence est trop complexe, il est possible d'utiliser une approche locale pour obtenir une meilleure approximation. Dans la deuxième partie nous analysons théoriquement notre approche dans le cas global mais aussi dans le cas local. Cela montre que les métriques apprises par notre approche généralisent bien. Dans les troisième et quatrième parties de ce chapitre nous considérons le problème de l'apprentissage de distances couleur perceptuelles pour montrer l'intérêt de notre approche dans une application réelle.

Chapitre 4 Le quatrième chapitre de cette thèse est dédié à notre seconde contribution. Comme dans le troisième chapitre il s'agit d'une approche d'apprentissage de métriques capable de prendre en compte la connaissance donnée par une métrique de référence. La principale différence est que, dans ce chapitre, le but n'est pas d'approximer cette métrique mais plutôt de l'utiliser pour aider au cours du processus d'apprentissage. Cette contribution est ainsi fortement liée aux domaines de l'apprentissage par transfert et de l'adaptation de domaine. Ce chapitre est divisé en sept parties. Dans la première nous présentons le cadre de travail

de l'apprentissage de métriques par transfert d'hypothèses qui correspond à un problème de minimisation avec terme de régularisation biaisé. Dans les deuxième, troisième et quatrième parties nous proposons une analyse théorique du cadre de travail proposé en utilisant trois approches théoriques différentes. Cela nous permet de dériver plusieurs mesures de l'apport de la métrique de référence. Dans la cinquième partie de ce chapitre nous résumons les différentes bornes et dans la sixième partie nous présentons plusieurs fonctions de perte et termes de régularisation pouvant être utilisés dans notre cadre de travail. Dans la dernière partie nous montrons que ce cadre peut être utilisé en pratique pour obtenir des résultats compétitifs sur plusieurs tâches d'apprentissage par transfert.

Dans la dernière partie de cette thèse nous introduisons nos deux dernières contributions où nous proposons de nouvelles façons de contrôler le comportement des métriques apprises.

Chapitre 5 Dans ce cinquième chapitre nous présentons notre troisième contribution. Dans celle-ci, plutôt que d'utiliser les contraintes classiques de similarités et dissimilarités nous proposons de considérer que la métrique devrait rapprocher les exemples d'apprentissage de points virtuels définis à priori. Cela nous permet d'apprendre une métrique à l'aide d'une régression et de réduire le nombre de contraintes considérées. Dans la première partie de ce chapitre nous présentons notre algorithme. Dans la seconde nous adressons le problème de sélectionner les points virtuels et de définir les contraintes. Dans la troisième partie nous proposons une analyse théorique de l'algorithme proposé et nous montrons, d'une part, qu'apprendre une métrique avec notre approche est fondé et, d'autre part, qu'il est possible de dériver des liens avec une approche plus classique d'apprentissage de métriques. Dans la dernière partie nous validons empiriquement l'intérêt de notre approche.

Chapitre 6 Le sixième chapitre de cette thèse introduit la dernière contribution de celle-ci. Il s'agit d'une nouvelle méthode capable d'apprendre une métrique pouvant bouger des blocs d'exemples en approximant la transformation correspondant à la solution d'un problème de transport optimal. Dans la première partie de ce chapitre nous introduisons de manière formelle le problème du transport optimal. Dans la deuxième partie nous présentons notre formulation tandis que dans la troisième nous proposons une approche efficace pour l'optimiser. Dans la quatrième partie de ce chapitre nous proposons une discussion théorique qui montre que si les suppositions classiques faites dans le domaine du transport optimal sont correctes alors notre approche est fondée. Dans la dernière partie nous proposons une validation empirique de notre méthode sur des problèmes d'adaptation de domaine et d'édition d'images.

E.2 Résumé du Chapitre 1

Dans ce chapitre nous présentons plusieurs notions essentielles à la bonne compréhension de cette thèse. En particulier nous formalisons le cadre de travail de l'apprentissage par

minimisation empirique du risque sur lequel sont basées nos différentes contributions algorithmiques. De la même façon nous présentons deux cadres théoriques qui permettent de dériver des bornes en généralisation pour la minimisation du risque. Comme nous le verrons dans le second chapitre ces deux cadres théoriques ont été étendus avec succès au problème de l'apprentissage de métriques. Dans ce manuscrit nous les utiliserons pour démontrer que nos algorithmes sont capables d'apprendre des métriques qui généralisent bien. D'un point de vue plus pratique nous présentons plusieurs fonctions de perte et termes de régularisation pouvant être utilisés dans le cadre de la minimisation du risque. Nous proposons aussi une définition formelle de la notion de métrique considérée dans cette thèse. Pour finir nous présentons l'algorithme de classification des plus proches voisins et le problème de l'adaptation de domaine qui seront utilisés pour empiriquement démontrer l'intérêt de la plupart de nos contributions.

E.3 Résumé du Chapitre 2

Dans ce chapitre nous proposons une revue, non exhaustive, de l'état de l'art en apprentissage de métriques. Ainsi nous nous intéressons tout particulièrement aux méthodes proches de nos contributions. Cela correspond à des approches qui apprennent le même genre de métriques, considèrent des façons similaires d'effectuer l'étape d'apprentissage, dérivent le même genre de bornes en généralisation ou apprennent une métrique pour résoudre les mêmes tâches.

E.4 Résumé du Chapitre 3

Dans ce chapitre nous nous intéressons au problème de l'estimation d'une métrique de référence inconnue à partir d'un ensemble de paires d'exemples. Une solution à ce problème est d'utiliser l'apprentissage de métriques pour approximer de façon automatique les valeurs de cette métrique de référence. Cependant, la plupart des algorithmes d'apprentissage de métriques s'intéressent à l'estimation de la proximité relative des exemples d'apprentissage plutôt qu'à la distance effective qui les sépare. Dans ce chapitre nous proposons un nouvel algorithme d'apprentissage de métriques locales nous permettant, à l'aide d'une distance de Mahalanobis, d'approximer de façon précise une métrique de référence. En utilisant le cadre théorique de la stabilité uniforme nous dérivons des bornes en généralisation sur le modèle appris qui montrent que notre méthode est fondée théoriquement. De plus nous évaluons notre approche sur un problème de vision par ordinateur consistant à calculer des différences de couleurs qui soient perceptuellement uniformes. Avoir des distances qui reflètent la perception humaine des couleurs de la scène est essentiel dans les applications de vision par ordinateur comme la segmentation d'images ou la détection d'objets saillants. Cependant, dans la plupart des cas, il est uniquement possible d'avoir accès aux couleurs de l'image sans aucun moyen de revenir aux couleurs de la scène. Il existe deux approches principales permettant de résoudre ce problème. D'un côté, il est possible de calculer directement une distance perceptuelle entre

les couleurs de l'image considérée. Cependant cette distance est coûteuse à calculer et dépend des conditions d'acquisition ce qui implique qu'elle est bien souvent loin des différences entre couleurs de la scène. D'un autre côté, il est possible d'estimer les couleurs de la scène à partir de celles de l'image puis de calculer une distance perceptuellement uniforme. Cependant, cela implique une connaissance sur les conditions d'acquisition qui n'est pas raisonnable pour la plupart des applications. Notre approche nous permet d'apprendre une métrique qui est à la fois invariante aux conditions d'acquisition et calculable à partir des couleurs des images. Nous évaluons l'intérêt de cette dernière en montrant sa capacité à (i) généraliser à de nouvelles couleurs et de nouveaux appareils photographiques et (ii) aider dans un problème de segmentation.

E.5 Résumé du Chapitre 4

Nous considérons le problème du transfert de connaissances à priori dans le contexte de l'apprentissage supervisé de métriques. De façon plus précise nous considérons des problèmes à régularisation biaisée qui utilisent un métrique de référence, une métrique source, venant d'un problème différent mais relié et pouvant potentiellement aider lors de l'apprentissage d'une métrique avec peu de données. Si ce cadre a déjà été appliqué avec succès de manière empirique, il n'existe pas de cadre théorique justifiant une telle approche. Dans ce chapitre, nous proposons de résoudre ce problème en proposant une analyse théorique basée sur trois approches différentes. Tout d'abord nous présentons une nouvelle définition de la stabilité, on-average-replace-two-stability, qui nous permet de montrer des bornes en généralisation en moyenne avec un taux de convergence rapide lorsqu'une métrique source auxiliaire est utilisée pour biaiser le terme de régularisation. Ensuite nous considérons une notion de stabilité algorithmique adaptée au cadre de l'apprentissage de métriques régularisé et nous prouvons une borne en généralisation probabiliste montrant l'intérêt d'utiliser un terme de régularisation biaisé avec pondération de la métrique source. Nous proposons une solution algorithmique à ce problème de pondération que nous évaluons (i) dans un problème d'apprentissage de métriques classique et (ii) dans un problème d'apprentissage par transfert avec peu de données cibles. Enfin nous dérivons une borne en généralisation basée sur la complexité de Rademacher de la classe de métriques considérée en prenant notamment en compte la métrique de référence. Cette borne souligne l'intérêt d'utiliser une bonne métrique source en montrant que, lorsque celle-ci est une solution idéale au problème, l'apprentissage n'est plus nécessaire. Pour justifier l'intérêt de ce cadre de travail nous proposons plusieurs exemples de fonctions de perte et de termes de régularisation qui peuvent être utilisés dans le cadre d'une ou plusieurs des approches théoriques considérées.

E.6 Résumé du Chapitre 5

Dans ce chapitre nous nous intéressons à l'apprentissage supervisé de distances de type Mahalanobis. Les approches existantes cherchent principalement à apprendre un nouvel espace de représentation en fonction de contraintes prenant en compte des informations de similarité et de dissimilarité entre les exemples. Ici, au lieu de rapprocher où d'éloigner les exemples selon ce type de contraintes, nous proposons d'introduire le concept de points virtuels nous servant de supports pour le déplacement des exemples d'apprentissage. Ainsi, les exemples d'apprentissage sont rapprochés d'un point virtuel qui leur a été affecté à priori permettant alors de réduire le nombre de contraintes à satisfaire et de contrôler de façon explicite le comportement de la métrique pour chaque exemple. Nous montrons que l'approche proposée peut être résolue en forme close et qu'il est alors possible de travailler dans l'espace induit par un noyau. Nous proposons deux analyses théoriques, la première prouvant la capacité de généralisation des métriques apprises avec notre méthode et la seconde établissant des liens avec une approche d'apprentissage de métriques classique. De plus nous proposons deux solutions efficaces au difficile problème de la sélection des points virtuels, l'une d'elle étant basée sur de récentes avancées dans le domaine du transport optimal. Pour finir, nous évaluons notre approche sur plusieurs jeux de données classiques en apprentissage de métriques.

E.7 Résumé du Chapitre 6

Dans ce chapitre nous proposons d'adresser le problème de l'apprentissage d'une transformation, induisant une distance de Mahalanobis, qui approxime une transformation géométrique particulière. Une telle métrique pourrait être très bénéfique dans le contexte de l'adaptation de domaine où le but est d'aligner les domaines sources et cibles. Ici nous proposons de considérer des transformations géométriques induites par la résolution d'un problème de transport optimal. En effet, il s'agit d'une procédure raisonnable pour aligner des distributions et sa capacité à résoudre des problèmes d'adaptation de domaine a déjà été démontrée. La plupart des approches en transport optimal utilisent la formulation donnée par Kantorovich et apprennent un couplage probabiliste $\mathbf{\Gamma}$ entre les différents exemples d'apprentissage. Cependant elles n'abordent pas le problème de l'apprentissage de la transformation $f_{\mathcal{S}\to\mathcal{T}}$ liée au problème de Monge. En conséquence le couplage appris ne peut-être utilisé que sur les exemples d'apprentissage et pas sur de nouveaux exemples ce qui réduit l'intérêt potentiel de telles approches. Dans ce chapitre nous proposons de combiner l'apprentissage de métriques et le transport optimal dans un nouveau cadre de travail nous permettant d'apprendre conjointement le couplage et une approximation de la transformation correspondante. Cette approximation prend la forme d'une matrice $\mathbf{L}$ correspondant à une nouvelle métrique dans le domaine source. Dans ce cas nous montrons que notre approche est liée à RVML, présenté dans le Chapitre 5, où les points virtuels associés à chaque exemple sont définis comme le résultat du couplage induit par le transport. Cependant, plutôt que de considérer que le couplage est défini a priori, nous proposons de l'apprendre en même temps que la métrique.

Ainsi, nous obtenons une formulation jointe et convexe pouvant être optimisée de façon efficace et ayant le bénéfice de lisser le résultat du transport optimal. En pratique nous montrons l'intérêt de notre méthode pour deux tâches, l'une en adaptation de domaine et l'autre en édition d'images.

E.8 Conclusion

Dans cette thèse nous avons adressé le problème de l'apprentissage de métriques à comportement contrôlé. Nous avons considéré deux types de contrôle sur la métrique apprise. D'une part, nous avons considéré le problème de l'apprentissage par rapport à une métrique de référence donnée soit sous la forme d'une distance pour un nombre limité de paires d'exemples soit directement sous la forme d'un modèle. D'autre part, nous avons considéré le problème de l'apprentissage de la transformation induite par une distance de Mahalanobis soit pour contrôler de façon précise le mouvement de chaque exemple soit pour approximer une transformation géométrique. Nos différentes contributions sont à la fois algorithmiques et théoriques.

Résumé des Contributions

La plupart des algorithmes d'apprentissage de métriques s'intéressent à l'obtention de métriques capables de rapprocher les exemples similaires tout en éloignant les exemples dissimilaires. Cependant, il peut parfois être intéressant de prédire une valeur précise entre deux exemples. C'est par exemple le cas lorsque l'on a accès à un nombre limité de paires d'exemples pour lesquelles la valeur d'une métrique de référence est connue. Dans notre première contribution nous avons adressé le problème de l'approximation de cette métrique de référence. Nous avons proposé une approche d'apprentissage de métriques locales que nous avons analysé théoriquement pour montrer que si le modèle a été appris avec un nombre suffisant d'exemples, il généralise bien. De plus nous avons évalué notre approche sur le problème de vision par ordinateur qu'est l'estimation de distances couleur perceptuelles. Pour cela nous avons créé un nouveau jeu de données spécialement dédié à cette tâche. Nos résultats empiriques ont montré le bon comportement de notre approche ainsi que sa capacité à approximer la métrique de référence. Le nouveau jeu de données ainsi que la distance perceptuellement uniforme apprise sont distribués gratuitement (Perrot et al., 2014a).

Plusieurs approches d'apprentissage de métriques montrent de façon empirique l'intérêt d'utiliser une information supplémentaire, sous forme d'une métrique source, mais ne prouvent pas ces bénéfices de façon théorique. Dans notre deuxième contribution nous avons proposé de résoudre ce problème. Ainsi, nous avons formalisé le cadre de travail de l'apprentissage de métriques par transfert d'hypothèse où l'idée est de prendre en compte une métrique source dans un terme de régularisation biaisé. Nous avons proposé une analyse théorique de ce cadre nous permettant de dériver trois mesures différentes de l'apport d'une métrique source. Ces mesures représentent différents moyens d'évaluer l'intérêt de la métrique de référence pour le problème considéré. Deux de ces mesures sont théoriques et donc difficiles à utiliser en

pratique. La troisième, cependant, est empirique ce qui signifie qu'elle peut être utilisée pour sélectionner la meilleur métrique source dans un ensemble. Pour illustrer cela nous avons proposé un algorithme de pondération de l'importance de la métrique source. Nous avons de plus démontré l'intérêt de notre cadre de travail en montrant que de nombreuses fonctions de perte et termes de régularisations pouvaient être utilisés. Enfin nous l'avons empiriquement évalué sur un problème d'apprentissage de métriques classique mais aussi sur une tâche d'adaptation de domaine semi-supervisé.

La plupart des approches d'apprentissage de métriques utilisent des contraintes de similarité et dissimilarité pour apprendre une métrique mais ne contrôlent pas de façon explicite le comportement de la transformation induite. Dans notre troisième contribution nous avons adressé ce problème en proposant une nouvelle approche où la destination des exemples, après projection par la transformation, est choisie de manière explicite à l'aide de points virtuels. Cela nous a permis de contrôler de manière précise la métrique apprise et donc d'apprendre des modèles plus adaptés à la tâche considérée. Par exemple, pour un problème de classification, nous avons proposé des points virtuels basés sur les différentes classes de telle façon que chaque axe de l'espace de projection de la métrique apprise soit discriminant pour une classe particulière. Nous avons montré que notre approche peut facilement apprendre à partir de l'espace induit par un noyau et donc apprendre des métriques très expressives. Nous avons aussi proposé une étude théorique montrant des liens entre notre méthode et une approche classique d'apprentissage de métriques. Enfin, nous avons démontré empiriquement ses bonnes performances sur plusieurs jeux de données classiques.

Dans notre quatrième contribution nous avons abordé un problème similaire à celui de notre troisième contribution. Cependant, au lieu de contrôler de façon explicite le comportement de chaque exemple individuellement, nous avons proposé de forcer la métrique à suivre une transformation géométrique particulière. Ainsi, nous avons considéré des transformations induites par le couplage appris par un problème de transport optimal discret, ce qui est d'un intérêt tout particulier pour des tâches d'adaptation de domaine. Nous avons proposé une solution pour apprendre de façon jointe ce couplage et la transformation induite par une métrique. Nous avons dérivé une méthode d'optimisation efficace et nous avons montré que cette approche pouvait être reliée à notre troisième contribution où les points virtuels et la transformation sont appris de façon jointe. Nous avons empiriquement démontré le bon comportement de notre approche pour un problème d'adaptation de domaine non supervisé ainsi que pour une tâche d'édition d'images.

Perspectives

Nous avons déjà présenté des perspectives spécifiques pour chacune de nos contributions. Dans cette partie, nous proposons plutôt de considérer des travaux futurs généraux qui peuvent représenter de nouvelles directions de recherche découlant des éléments présentés dans cette thèse.

D'un point de vue algorithmique nos contributions sont principalement basées sur des

problèmes d'optimisation directs. Une première perspective serait d'étendre les concepts présentés dans ce manuscrit au contexte de l'apprentissage en ligne. Ainsi, il pourrait être intéressant de développer des mécanismes capables de détecter de potentiels changements dans la distribution des exemples et d'alors changer automatiquement le comportement de la métrique considérée. Une telle approche pourrait, par exemple, être utilisée dans un contexte de suivi d'objets dans des vidéos où les variations dans la scène peuvent potentiellement appeler à des comportements différents de la métrique. Une autre perspective serait de considérer l'apprentissage actif pour améliorer le contrôle de la métrique. Par exemple, lorsque l'on apprend une transformation, il pourrait être intéressant d'obtenir un retour de l'utilisateur pour vérifier que les exemples sont correctement déplacés. Un domaine d'application pourrait être celui de l'adaptation de domaine, où l'apprentissage actif à déjà fait ses preuves (Berlind and Urner, 2015), où obtenir des retours sur des exemples bien choisis pourrait assurer que la métrique estime de façon correcte les différences entre les distributions.

D'un point de vue plus théorique notons que dans cette thèse nous nous sommes principalement intéressés à la capacité de généralisation des métriques apprises et pas à leur impact sur l'application dans laquelle elles sont utilisées. Partant de cette dernière idée, Balcan et al. (2008) ont montré que l'erreur d'un classificateur linéaire était liée à une mesure de l'apport de la similarité utilisée pour l'apprendre. Cette mesure de l'apport d'une métrique est reliée à sa capacité à rapprocher les exemples similaires et à éloigner les exemples dissimilaires. Cependant, lors de l'apprentissage d'une métrique à comportement contrôlé cette mesure ne sera pas forcément adaptée. Par exemple, lors de l'apprentissage d'une métrique à l'aide d'une métrique de référence (Chapitres 3 et 4) il serait probablement plus intéressant de considérer une mesure prenant en compte cette information supplémentaire. De la même façon, lors de l'apprentissage d'une transformation pour une tâche d'adaptation de domaine (Chapitres 4 et 6) il serait probablement plus intéressant de se focaliser sur la capacité de la métrique à aligner la source et la cible. Cela implique que la mesure de l'apport de la métrique dépend de la tâche considérée. Une perspective intéressante pourrait être de considérer des cadres théoriques capables de prendre en compte une notion d'apport reliée à la tâche considérée et de montrer qu'une bonne métrique est en effet bénéfique.

Une autre perspective théorique est la dérivation de bornes à convergence rapides en présence d'informations supplémentaires. Dans le Chapitre 4 nous avons proposé une première solution à ce problème en utilisant la complexité de Rademacher et l'information supplémentaire qu'est la mesure de l'apport de la métrique source dérivée. Cependant, cette solution n'est pas satisfaisante dans le sens où la contrainte imposée sur la métrique source était plus forte que le résultat obtenu sur la métrique apprise. Dans tous les cas, cela reste un résultat encourageant puisqu'il montre qu'en utilisant des suppositions fortes il est possible d'obtenir un taux de convergence rapide. Ainsi, s'il est possible d'obtenir des suppositions plus faibles (Voir e.g. Srebro et al. (2010c)) il pourrait être possible de dériver des résultats plus significatifs.

Bibliography

Radhakrishna Achanta and Sabine Süsstrunk. Saliency detection using maximum symmetric surround. In *2010 IEEE International Conference on Image Processing*, pages 2653–2656. IEEE, 2010. 64, 72, 73

Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, 2011. 73

Julien Audiffren and Hachem Kadri. Stability of multi-task kernel regression algorithms. In *Proc. of ACML*, pages 1–16, 2013. 125

Maria-Florina Balcan, Avrim Blum, and Nathan Srebro. Improved guarantees for learning via similarity functions. *Computer Science Department*, page 126, 2008. 56, 136, 159, 206

Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002. 26, 27, 179

Aurélien Bellet and Amaury Habrard. Robustness and generalization for metric learning. *Neurocomputing*, 151:259–267, 2015. 54

Aurélien Bellet, Amaury Habrard, and Marc Sebban. Learning good edit similarities with generalization guarantees. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 188–203. Springer, 2011. 56

Aurélien Bellet, Amaury Habrard, and Marc Sebban. Similarity learning for provably accurate sparse linear classification. In *ICML 2012*, pages 1871–1878, 2012. 44, 56, 105, 107, 128, 136

Aurélien Bellet, Amaury Habrard, and Marc Sebban. *Metric Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2015. 35

Shai Ben-David and Ruth Urner. Domain adaptation as learning with auxiliary information. In *New Directions in Transfer and Multi-Task-Workshop@ NIPS*, 2013. 86

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine Learning*, 79(1-2):151–175, 2010. 40, 86, 94

J.-D. Benamou, G. Carlier, M. Cuturi, L. Nenna, and G. Peyré. Iterative bregman projections for regularized transportation problems. *SISC*, 2015. 140

George Bennett. Probability inequalities for the sum of independent random variables. *Journal of the American Statistical Association*, 57(297):33–45, 1962. 26

Christopher Berlind and Ruth Urner. Active nearest neighbors in changing environments. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 1870–1879, 2015. 159, 206

Jinbo Bi, Dijia Wu, Le Lu, Meizhu Liu, Yimo Tao, and Matthias Wolf. Adaboost on low-rank psd matrices for metric learning. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2617–2624. IEEE, 2011. 53

Konstantinos Bitsakos, Cornelia Fermüller, and Yiannis Aloimonos. An experimental study of color-based segmentation algorithms based on the mean-shift concept. In *European conference on Computer vision*, pages 506–519. Springer, 2010. 64, 72, 73, 79, 80

Julien Bohné, Yiming Ying, Stéphane Gentric, and Massimiliano Pontil. Large margin local metric learning. In *European Conference on Computer Vision*, pages 679–694. Springer, 2014. 45, 57, 86, 107

Karsten M Borgwardt, Arthur Gretton, Malte J Rasch, Hans-Peter Kriegel, Bernhard Schölkopf, and Alex J Smola. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, 22(14):e49–e57, 2006. 58

Stéphane Boucheron, Gábor Lugosi, and Olivier Bousquet. Concentration inequalities. In *Advanced Lectures on Machine Learning*, pages 208–240. Springer, 2004. 26

O. Bousquet. *Concentration inequalities and empirical processes theory applied to the analysis of learning algorithms.* PhD thesis, Ecole Polytechnique, 2002. 102

Olivier Bousquet and André Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2:499–526, 2002a. 191

Olivier Bousquet and André Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2(Mar):499–526, 2002b. 26, 27, 55, 67, 125

L. Bruzzone and M. Marconcini. Domain adaptation problems: A DASVM classification technique and a circular validation strategy. *IEEE PAMI*, 32(5), 2010. 40, 150, 151

Gertjan J Burghouts and Jan-Mark Geusebroek. Performance evaluation of local colour invariants. *Computer Vision and Image Understanding*, 113(1):48–62, 2009. 72

G. Canas and L. Rosasco. Learning probability measures with respect to optimal transport metrics. In *NIPS*. 2012. 139, 140

Qiong Cao, Yiming Ying, and Peng Li. Similarity metric learning for face recognition. In *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, pages 2408–2415, 2013a. 86, 107

Qiong Cao, Yiming Ying, and Peng Li. Similarity metric learning for face recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2408–2415, 2013b. 57

Qiong Cao, Zheng-Chu Guo, and Yiming Ying. Generalization bounds for metric and similarity learning. *Machine Learning*, 102(1):115–132, 2016. 54, 55, 87, 95, 97, 107, 179, 187

Hong Chang and Dit-Yan Yeung. Locally linear metric adaptation for semi-supervised clustering. In *Proceedings of the twenty-first international conference on Machine learning*, page 20. ACM, 2004. 45, 56

Hong Chang and Dit-Yan Yeung. Locally smooth metric learning with application to image retrieval. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–7. IEEE, 2007. 45, 57

Gal Chechik, Varun Sharma, Uri Shalit, and Samy Bengio. Large scale online learning of image similarity through ranking. In *Iberian Conference on Pattern Recognition and Image Analysis*, pages 11–14. Springer, 2009. 44, 53

Gal Chechik, Varun Sharma, Uri Shalit, and Samy Bengio. Large scale online learning of image similarity through ranking. *Journal of Machine Learning Research*, 11(Mar):1109–1135, 2010. 44, 53, 57

Stéphan Clémençon, Gábor Lugosi, and Nicolas Vayatis. Ranking and empirical minimization of u-statistics. *The Annals of Statistics*, pages 844–874, 2008. 55

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3): 273–297, 1995. 30, 33, 34, 36

Corinna Cortes, Mehryar Mohri, and Jason Weston. A general regression technique for learning transductions. In *Proc. of ICML*, pages 153–160, 2005. 120

Corinna Cortes, Mehryar Mohri, and Jason Weston. A general regression framework for learning string-to-string mappings. In *Predicting Structured Data*. MIT Press, 2007. 189

N. Courty, R. Flamary, and D. Tuia. Domain adaptation with regularized optimal transport. In *ECML PKDD*, 2014a. 140, 142, 150

Nicolas Courty, Rémi Flamary, and Devis Tuia. Domain adaptation with regularized optimal transport. In *Proc. of ECML/PKDD*, pages 274–289, 2014b. 121, 122, 123, 138

Thomas M. Cover and Peter E. Hart. Nearest neighbor pattern classification. *IEEE Trans. Information Theory*, 13(1):21–27, 1967. 34, 38

M. Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *NIPS*, 2013a. 139, 140, 142, 150

M. Cuturi and A. Doucet. Fast computation of wasserstein barycenters. In *ICML*, 2014. 140

M. Cuturi and G. Peyré. A smoothed dual approach for variational wasserstein problems. *SIIMS*, 2016. 140

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Proc. of NIPS*, pages 2292–2300, 2013b. 123

Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893. IEEE, 2005. 72

Jason V Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S Dhillon. Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning*, pages 209–216. ACM, 2007. 43, 45, 48, 49, 53, 56, 57, 108, 109, 131

F. Deng, S. J. Kim, Y.-W. Tai, and M. Brown. *ACCV*, chapter Color-Aware Regularization for Gradient Domain Image Manipulation. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. 153

Terry E Dielman. Least absolute value regression: recent contributions. *Journal of Statistical Computation and Simulation*, 75(4):263–286, 2005. 32

J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*, 2014. 150

B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *ICCV*, 2013. 150

S. Ferradans, N. Papadakis, G. Peyré, and J.-F. Aujol. Regularized discrete optimal transport. *SIIMS*, 2014. 140, 142, 144, 152

M. Frank and P. Wolfe. An algorithm for quadratic programming. *NRL*, 3(1-2), 1956. 144

Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, 55(1):119–139, 1997. 31

Yoav Freund and Robert E Schapire. Large margin classification using the perceptron algorithm. *Machine learning*, 37(3):277–296, 1999. 53

Jerome Friedman, Trevor Hastie, Robert Tibshirani, et al. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*, 28(2):337–407, 2000. 31

C. Frogner, C. Zhang, H. Mobahi, M. Araya, and T. Poggio. Learning with a wasserstein loss. In *NIPS*. 2015. 140

Keinosuke Fukunaga and Larry Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on information theory*, 21(1): 32–40, 1975. 79

Bo Geng, Dacheng Tao, and Chao Xu. DAML: domain adaptation metric learning. *IEEE Transactions on Image Processing*, 20(10):2980–2989, 2011. 57

Amir Globerson and Sam T Roweis. Metric learning by collapsing classes. In *Advances in neural information processing systems*, pages 451–458, 2005. 48, 131

Jacob Goldberger, Geoffrey E Hinton, Sam T Roweis, and Ruslan Salakhutdinov. Neighbourhood components analysis. In *Advances in neural information processing systems*, pages 513–520, 2004. 43, 47, 48, 56

Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012*, pages 2066–2073, 2012. 39, 40, 110, 150

Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. 2007. 39

Zheng-Chu Guo and Yiming Ying. Guaranteed classification via regularized similarity learning. *Neural computation*, 26(3):497–522, 2014. 56

Judy Hoffman, Erik Rodner, Jeff Donahue, Kate Saenko, and Trevor Darrell. Efficient learning of domain-invariant image representations. *CoRR*, abs/1301.3224, 2013. 40, 110

Min Huang, Haoxue Liu, Guihua Cui, M Ronnier Luo, and Manuel Melgosa. Evaluation of threshold color differences using printed samples. *JOSA A*, 29(6):883–891, 2012. 72

Peter J. Huber. Robust estimation of a location parameter. *Annals of Mathematical Statistics*, 35(1):73–101, March 1964. 32

Zhouyuan Huo, Feiping Nie, and Heng Huang. Robust and effective metric learning using capped trace norm: Metric learning via capped trace norm. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1605–1614. ACM, 2016. 44, 52, 57

Adrian Ilie and Greg Welch. Ensuring color consistency across multiple cameras. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1268–1275. IEEE, 2005. 78

M. Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *ICML*, 2013. 144

Prateek Jain, Brian Kulis, Inderjit S Dhillon, and Kristen Grauman. Online metric learning and fast similarity search. In *Advances in neural information processing systems*, pages 761–768, 2009. 53

Rong Jin, Shijun Wang, and Yang Zhou. Regularized distance metric learning: Theory and algorithm. In *Advances in neural information processing systems*, pages 862–870, 2009. 43, 49, 54, 55, 57, 87, 90, 91, 105, 107, 126

L. Kantorovich. On the translocation of masses. *C.R. (Doklady) Acad. Sci. URSS (N.S.)*, 37, 1942. 141

Purushottam Kar and Prateek Jain. Similarity-based learning via data driven embeddings. In *Advances in neural information processing systems*, pages 1998–2006, 2011. 38, 121, 129

Dor Kedem, Stephen Tyree, Fei Sha, Gert R Lanckriet, and Kilian Q Weinberger. Non-linear metric learning. In *Advances in Neural Information Processing Systems*, pages 2573–2581, 2012. 45, 50, 128, 129

Rahat Khan, Joost Van de Weijer, Fahad Shahbaz Khan, Damien Muselet, Christophe Ducottet, and Cecile Barat. Discriminative color descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2866–2873, 2013. 73

Seon Joo Kim, Hai Ting Lin, Zheng Lu, Sabine Süsstrunk, Stephen Lin, and Michael S. Brown. A new in-camera imaging model for color computer vision and its application. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(12):2289–2302, 2012a. 73

Seon Joo Kim, Hai Ting Lin, Zheng Lu, Sabine Süsstrunk, Stephen Lin, and Michael S Brown. A new in-camera imaging model for color computer vision and its application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(12):2289–2302, 2012b. 73

Philip A Knight. The sinkhorn-knopp algorithm: convergence and applications. *SIAM Journal on Matrix Analysis and Applications*, 30(1):261–275, 2008. 123

Brian Kulis, Kate Saenko, and Trevor Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1785–1792. IEEE, 2011. 44, 57, 138

Matt Kusner, Stephen Tyree, Kilian Q Weinberger, and Kunal Agrawal. Stochastic neighbor compression. In *Proceedings of the 31st international conference on machine learning (ICML-14)*, pages 622–630, 2014. 124

Ilja Kuzborskij and Francesco Orabona. Stability and hypothesis transfer learning. In *Proc. of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, pages 942–950, 2013. 86, 94

Ilja Kuzborskij and Francesco Orabona. Learning by transferring from auxiliary hypotheses. *CoRR*, abs/1412.1619, 2014. 101, 106, 112

John Langford. Tutorial on practical prediction theory for classification. *Journal of machine learning research*, 6(Mar):273–306, 2005. 26

RE Larraín, DM Schaefer, and JD Reed. Use of digital images to estimate cie color coordinates of beef. *Food Research International*, 41(4):380–385, 2008. 73, 74

Marc T Law, Nicolas Thome, and Matthieu Cord. Quadruplet-wise image similarity learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 249–256, 2013. 51

Marc T Law, Nicolas Thome, and Matthieu Cord. Fantope regularization in metric learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1051–1058, 2014. 44, 51, 52, 57

Y. Lei and Y. Ying. Generalization analysis of multi-modal metric learning. *Analysis and Applications*, 2015. 96

Katherine Leon, Domingo Mery, Franco Pedreschi, and Jorge Leon. Color measurement in l? a? b? units from rgb digital images. *Food research international*, 39(10):1084–1091, 2006. 73, 74

M. Lichman. UCI machine learning repository, 2013. 109, 128

Stuart P. Lloyd. Least squares quantization in PCM. *IEEE Trans. Information Theory*, 28 (2):129–136, 1982. 34

David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 72

Prasanta Chandra Mahalanobis. On the generalized distance in statistics. *Proceedings of the National Institute of Sciences (Calcutta)*, 2:49–55, 1936. 37, 86, 109

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. In *COLT 2009 - The 22nd Conference on Learning Theory, Montreal, Quebec, Canada, June 18-21, 2009*, 2009. 40

R. McCann. A convexity principle for interacting gases. *Advances in Mathematics*, 128(1), 1997. 140

Colin McDiarmid. *Surveys in Combinatorics*, chapter On the method of bounded differences, pages 148–188. Cambridge University Press, 1989. 26, 54, 164

Manuel Melgosa, Rafael Huertas, and Roy S Berns. Performance of recent advanced color-difference formulas using the standardized residual sum of squares index. *JOSA A*, 25(7): 1828–1834, 2008. 77

Aleksandra Mojsilovic. A computational model for color naming and describing color composition of images. *IEEE Transactions on Image processing*, 14(5):690–699, 2005. 73

J. Mueller and T. Jaakkola. Principal differences analysis: Interpretable characterization of differences between distributions. In *NIPS*. 2015. 140

Albert H Munsell. A pigment color system and notation. *The American Journal of Psychology*, 23(2):236–244, 1912. 75

Maria-Irina Nicolae, Éric Gaussier, Amaury Habrard, and Marc Sebban. Joint semi-supervised similarity learning for linear classification. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 594–609. Springer, 2015. 56, 105, 107

S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.*, 22 (10), 2010. 39

Shibin Parameswaran and Kilian Q Weinberger. Large margin multi-task metric learning. In *Advances in neural information processing systems*, pages 1867–1875, 2010. 58, 86, 107

P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. *ACM Trans. on Graphics*, 22 (3), 2003. 152, 153, 155

Michaël Perrot and Amaury Habrard. Apprentissage de mtriques par rgression. In *Conférence francophone sur l'Apprentissage Automatique (CAp-15)*, 2015a.

Michaël Perrot and Amaury Habrard. Transfert d'informations en apprentissage de mtriques : une analyse thorique. In *Conférence francophone sur l'Apprentissage Automatique (CAp-15)*, 2015b.

Michaël Perrot and Amaury Habrard. Regressive virtual metric learning. In *Advances in Neural Information Processing Systems (NIPS-15)*, pages 1810–1818, 2015c.

Michaël Perrot and Amaury Habrard. A theoretical analysis of metric hypothesis transfer learning. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 1708–1717, 2015d.

Michaël Perrot and Amaury Habrard. Bornes en généralisation à convergence rapide pour le transfert d'hypothèses en apprentissage de métriques. In *Conférence francophone sur l'Apprentissage Automatique (CAp-16)*, 2016.

Michaël Perrot, Amaury Habrard, Damien Muselet, and Marc Sebban. Freely avaible on the authors' personal web pages., 2014a. 157, 204

Michaël Perrot, Amaury Habrard, Damien Muselet, and Marc Sebban. Modeling perceptual color differences by local metric learning. In *European Conference on Computer Vision (ECCV-15)*, pages 96–111. Springer International Publishing, 2014b.

Michaël Perrot, Amaury Habrard, Damien Muselet, and Marc Sebban. Modélisation de distances couleur uniformes par apprentissage de métriques locales. In *Conférence francophone sur l'Apprentissage Automatique (CAp-14)*, 2014c.

Michaël Perrot, Nicolas Courty, Rémi Flamary, and Amaury Habrard. Mapping estimation of discrete optimal transport. In *Advances in Neural Information Processing Systems (NIPS-16)*, 2016.

Ali Mustafa Qamar and Eric Gaussier. Online and batch learning of generalized cosine similarities. In *2009 Ninth IEEE International Conference on Data Mining*, pages 926–931. IEEE, 2009. 44

Ali Mustafa Qamar, Eric Gaussier, Jean-Pierre Chevallet, and Joo Hwee Lim. Similarity learning for nearest neighbor classification. In *2008 Eighth IEEE International Conference on Data Mining*, pages 983–988. IEEE, 2008. 44, 53, 56

S. Reich. A nonparametric ensemble transform method for bayesian inference. *SISC*, 2013. 140, 142

Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European conference on computer vision*, pages 213–226. Springer, 2010. 57, 138

Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Kernel principal component analysis. In *International Conference on Artificial Neural Networks*, pages 583–588. Springer, 1997. 58

Bernhard Schölkopf, Alexander J. Smola, and Klaus-Robert Müller. Kernel principal component analysis. In *Artificial Neural Networks - ICANN '97, 7th International Conference, Lausanne, Switzerland, October 8-10, 1997, Proceedings*, pages 583–588, 1997. 36

V. Seguy and M. Cuturi. Principal geodesic analysis for probability measures under the optimal transport metric. In *NIPS*. 2015. 140

Murat Semerci and Ethem Alpaydın. Mixtures of large margin nearest neighbor classifiers. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 675–688. Springer, 2013. 45, 50, 56, 82

Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning - From Theory to Algorithms*, chapter Regularization and Stability, pages 137–149. Cambridge University Press, 2014a. 88, 164, 179

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge University Press, 2014b. 28

Shai Shalev-Shwartz, Yoram Singer, and Andrew Y Ng. Online and batch learning of pseudo-metrics. In *Proceedings of the twenty-first international conference on Machine learning*, page 94. ACM, 2004. 52

Gaurav Sharma, Wencheng Wu, and Edul N Dalal. The ciede2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations. *Color Research & Application*, 30(1):21–30, 2005. 72, 73, 74, 75, 77, 78

Chunhua Shen, Junae Kim, Lei Wang, and Anton Hengel. Positive semidefinite metric learning with boosting. In *Advances in neural information processing systems*, pages 1651–1659, 2009. 43, 53

Chunhua Shen, Junae Kim, Lei Wang, and Anton van den Hengel. Positive semidefinite metric learning using boosting-like algorithms. *Journal of Machine Learning Research*, 13 (Apr):1007–1036, 2012. 43, 53, 107

Yuan Shi, Aurélien Bellet, and Fei Sha. Sparse compositional metric learning. *AAAI 2014*, pages 2078–2084, 2014. 43, 49, 50, 56, 128, 129

J. Solomon, R. Rustamov, G. Leonidas, and A. Butscher. Wasserstein propagation for semi-supervised learning. In *ICML*, 2014. 139, 140

J. Solomon, F. De Goes, G. Peyré, M. Cuturi, A. Butscher, A. Nguyen, T. Du, and L. Guibas. Convolutional wasserstein distances. *ACM Trans. on Graphics*, 34(4), 2015. 152

N. Srebro, K. Sridharan, and A. Tewari. Smoothness, low noise and fast rates. In *Proc. of NIPS*, 2010a. 96, 106

N. Srebro, K. Sridharan, and A. Tewari. Optimistic rates for learning with a smooth loss. *CoRR*, abs/1009.3896, 2010b. 105, 163

Nathan Srebro, Karthik Sridharan, and Ambuj Tewari. Smoothness, low noise and fast rates. In *Advances in neural information processing systems*, pages 2199–2207, 2010c. 159, 163, 206

M. Stokes, M. Anderson, S. Chandrasekar, and R. Motta. A standard default color space for the internet: sRGB. Technical report, Hewlett-Packard and Microsoft, November 1996. 73

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996. 31, 33

P. Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of optimization theory and applications*, 109(3), 2001. 143

Int. Telecommunications Union. Parameter values for the hdtv standards for production and international programme exchange, itu-r recommendation bt.709-4. Technical report, March 2000. 73

Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984. 25

Koen Van De Sande, Theo Gevers, and Cees Snoek. Evaluating color descriptors for object and scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 32 (9):1582–1596, 2010. 72

Joost Van de Weijer, Theo Gevers, and J-M Geusebroek. Edge and corner detection by photometric quasi-invariants. *IEEE transactions on pattern analysis and machine intelligence*, 27(4):625–630, 2005. 72

Joost Van de Weijer, Theo Gevers, and Andrew D Bagdanov. Boosting color saliency in image feature detection. *IEEE transactions on pattern analysis and machine intelligence*, 28(1):150–156, 2006. 72

Aad W Van Der Vaart and Jon A Wellner. Weak convergence. In *Weak Convergence and Empirical Processes*, pages 16–28. Springer, 1996. 70, 164

V. Vapnik. *Estimation of Dependences Based on Empirical Data*. Springer-Verlag, 1982. 26

V. N. Vapnik and A. Ya. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2):264–280, 1971. 26

Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998. 56

C. Villani. *Optimal transport: old and new*. Grund. der mathematischen Wissenschaften. Springer, 2009. ISBN 9783540710493. 139, 140, 141

Cédric Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008. 122

Jun Wang, Alexandros Kalousis, and Adam Woznica. Parametric local metric learning for nearest neighbor classification. In *Advances in Neural Information Processing Systems*, pages 1601–1609, 2012. 45, 56

Qianqian Wang, Fang Chen, Quanxue Gao, Xinbo Gao, and Feiping Nie. On the schatten norm for matrix based subspace learning and classification. *Neurocomputing*, 2016. 34

Kilian Q Weinberger and Lawrence K Saul. Fast solvers and efficient implementations for distance metric learning. In *Proceedings of the 25th international conference on Machine learning*, pages 1160–1167. ACM, 2008. 45, 50, 64

Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10(Feb):207–244, 2009. 50

Kilian Q Weinberger and Gerald Tesauro. Metric learning for kernel regression. In *AISTATS*, pages 612–619, 2007. 58

Kilian Q Weinberger, John Blitzer, and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. In *Advances in neural information processing systems*, pages 1473–1480, 2005. 43, 45, 49, 50, 56, 109, 129

G. Wyszecki and W. S. Stiles. *Color Science: Concepts and Methods, Quantitative Data and Formulas*. John Wiley & Sons Inc, 2nd revised ed., New York, 2000. 72

Eric P Xing, Andrew Y Ng, Michael I Jordan, and Stuart Russell. Distance metric learning with application to clustering with side-information. *Advances in neural information processing systems*, 15:505–512, 2002. 43, 47, 56

Caiming Xiong, David Johnson, Ran Xu, and Jason J Corso. Random forests for metric learning with implicit pairwise position dependence. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 958–966. ACM, 2012a. 45

Ying Xiong, Kate Saenko, Trevor Darrell, and Todd Zickler. From pixels to physics: Probabilistic color de-rendering. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 358–365. IEEE, 2012b. 73

Huan Xu and Shie Mannor. Robustness and generalization. In *COLT 2010 - The 23rd Conference on Learning Theory, Haifa, Israel, June 27-29, 2010*, pages 503–515, 2010. 26

Huan Xu and Shie Mannor. Robustness and generalization. *Machine learning*, 86(3):391–423, 2012. 26, 70

Huan Xu, Constantine Caramanis, and Shie Mannor. Sparse algorithms are not stable: A no-free-lunch theorem. *IEEE transactions on pattern analysis and machine intelligence*, 34 (1):187–193, 2012. 27

Wufeng Xue, Xuanqin Mou, Lei Zhang, and Xiangchu Feng. Perceptual fidelity aware mean squared error. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 705–712, 2013. 72

Yiming Ying, Kaizhu Huang, and Colin Campbell. Sparse metric learning via smooth optimization. In *Advances in neural information processing systems*, pages 2214–2222, 2009. 44, 49, 50, 56, 107

Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006. 34

Valentina Zantedeschi, Remi Emonet, and Marc Sebban. Metric learning as convex combinations of local models with generalization guarantees. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 82

Zheng-Jun Zha, Tao Mei, Meng Wang, Zengfu Wang, and Xian-Sheng Hua. Robust distance metric learning with auxiliary knowledge. In *IJCAI*, pages 1327–1332, 2009. 49, 57

ErHeng Zhong, Wei Fan, Qiang Yang, Olivier Verscheure, and Jiangtao Ren. Cross validation framework to choose amongst models and datasets for transfer learning. In *Proc. of European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD)*, volume 6323 of *LNCS*, pages 547–562. Springer, 2010. 151

Abstract Many Machine Learning algorithms make use of a notion of distance or similarity between examples to solve various problems such as classification, clustering or domain adaptation. Depending on the tasks considered these metrics should have different properties but manually choosing an adapted comparison function can be tedious and difficult. A natural trend is then to automatically tailor such metrics to the task at hand. This is known as Metric Learning and the goal is mainly to find the best parameters of a metric under some specific constraints. Standard approaches in this field usually focus on learning Mahalanobis distances or Bilinear similarities and one of the main limitations is that the control over the behaviour of the learned metrics is often limited. Furthermore if some theoretical works exist to justify the generalization ability of the learned models, most of the approaches do not come with such guarantees. In this thesis we propose new algorithms to learn metrics with a controlled behaviour and we put a particular emphasis on the theoretical properties of these algorithms. We propose four distinct contributions which can be separated in two parts, namely (i) controlling the metric with respect to a reference metric and (ii) controlling the underlying transformation corresponding to the learned metric. Our first contribution is a local metric learning method where the goal is to regress a distance proportional to the human perception of colors. Our approach is backed up by theoretical guarantees on the generalization ability of the learned metrics. In our second contribution we are interested in theoretically studying the interest of using a reference metric in a biased regularization term to help during the learning process. We propose to use three different theoretical frameworks allowing us to derive three different measures of goodness for the reference metric. These measures give us some insights on the impact of the reference metric on the learned one. In our third contribution we propose a metric learning algorithm where the underlying transformation is controlled. The idea is that instead of using similarity and dissimilarity constraints we associate each learning example to a so-called virtual point belonging to the output space associated with the learned metric. We theoretically show that metrics learned in this way generalize well but also that our approach is linked to a classic metric learning method based on pairs constraints. In our fourth contribution we also try to control the underlying transformation of a learned metric. However instead of considering a point-wise control we consider a global one by forcing the transformation to follow the geometrical transformation associated to an optimal transport problem. From a theoretical standpoint we propose a discussion on the link between the transformation associated with the learned metric and the transformation associated with the optimal transport problem. On a more practical side we show the interest of our approach for domain adaptation but also for a task of seamless copy in images.

Résumé De nombreux algorithmes en Apprentissage Automatique utilisent une notion de distance ou de similarité entre les exemples pour résoudre divers problèmes tels que la classification, le partitionnement ou l'adaptation de domaine. En fonction des tâches considérées ces métriques devraient avoir des propriétés différentes mais les choisir manuellement peut-être fastidieux et difficile. Une solution naturelle est alors d'adapter automatiquement ces métriques à la tâche considérée. Il s'agit alors d'un problème connu sous le nom d'Apprentissage de Métriques et où le but est principalement de trouver les meilleurs paramètres d'une métrique respectant des contraintes spécifiques. Les approches classiques dans ce domaine se focalisent habituellement sur l'apprentissage de distances de Mahalanobis ou de similarités Bilinéaires et l'une des principales limitations est le fait que le contrôle du comportement de ces métriques est souvent limité. De plus, si des travaux théoriques existent pour justifier de la capacité de généralisation des modèles appris, la plupart des approches ne présentent pas de telles garanties. Dans cette thèse nous proposons de nouveaux algorithmes pour apprendre des métriques à comportement contrôlé et nous mettons l'accent sur les propriétés théoriques de ceux-ci. Nous proposons quatre contributions distinctes qui peuvent être séparées en deux parties: (i) contrôler la métrique apprise en utilisant une métrique de référence et (ii) contrôler la transformation induite par la métrique apprise. Notre première contribution est une approche locale d'apprentissage de métriques où le but est de régresser une distance proportionnelle à la perception humaine des couleurs. Notre approche est justifiée théoriquement par des garanties en généralisation sur les métriques apprises. Dans notre deuxième contribution nous nous sommes intéressés à l'analyse théorique de l'intérêt d'utiliser une métrique de référence dans un terme de régularisation biaisé pour aider lors du processus d'apprentissage. Nous proposons d'utiliser trois cadres théoriques différents qui nous permettent de dériver trois mesures différentes de l'apport de la métrique de référence. Ces mesures nous donnent un aperçu de l'impact de la métrique de référence sur celle apprise. Dans notre troisième contribution nous proposons un algorithme d'apprentissage de métriques où la transformation induite est contrôlée. L'idée est que, plutôt que d'utiliser des contraintes de similarité et de dissimilarité, chaque exemple est associé à un point virtuel qui appartient déjà à l'espace induit par la métrique apprise. D'un point de vue théorique nous montrons que les métriques apprises de cette façon généralisent bien mais aussi que notre approche est liée à une méthode plus classique d'apprentissage de métriques basée sur des contraintes de paires. Dans notre quatrième contribution nous essayons aussi de contrôler la transformation induite par une métrique apprise. Cependant, plutôt que considérer un contrôle individuel pour chaque exemple, nous proposons une approche plus globale en forçant la transformation à suivre une transformation géométrique associée à un problème de transport optimal. D'un point de vue théorique nous proposons une discussion sur le lien entre la transformation associée à la métrique apprise et la transformation associée au problème de transport optimal. D'un point de vue plus pratique nous montrons l'intérêt de notre approche pour l'adaptation de domaine mais aussi pour l'édition d'images.