# A Survey on Metric Learning for Feature Vectors and Structured Data

Aurélien Bellet, Amaury Habrard, Marc Sebban

Technical report

# A Survey on Metric Learning for Feature Vectors and Structured Data

**Aurélien Bellet**[*] BELLET@USC.EDU
*Department of Computer Science*
*University of Southern California*
*Los Angeles, CA 90089, USA*

**Amaury Habrard** AMAURY.HABRARD@UNIV-ST-ETIENNE.FR
**Marc Sebban** MARC.SEBBAN@UNIV-ST-ETIENNE.FR
*Laboratoire Hubert Curien UMR 5516*
*Université de Saint-Etienne*
*18 rue Benoit Lauras, 42000 St-Etienne, France*

## Abstract

The need for appropriate ways to measure the distance or similarity between data is ubiquitous in machine learning, pattern recognition and data mining, but handcrafting such good metrics for specific problems is generally difficult. This has led to the emergence of metric learning, which aims at automatically learning a metric from data and has attracted a lot of interest in machine learning and related fields for the past ten years. This survey paper proposes a systematic review of the metric learning literature, highlighting the pros and cons of each approach. We pay particular attention to Mahalanobis distance metric learning, a well-studied and successful framework, but additionally present a wide range of methods that have recently emerged as powerful alternatives, including nonlinear metric learning, similarity learning and local metric learning. Recent trends and extensions, such as semi-supervised metric learning, metric learning for histogram data and the derivation of generalization guarantees, are also covered. Finally, this survey addresses metric learning for structured data, in particular edit distance learning, and attempts to give an overview of the remaining challenges in metric learning for the years to come.

**Keywords:** Metric Learning, Similarity Learning, Mahalanobis Distance, Edit Distance

## 1. Introduction

The notion of *pairwise metric*—used throughout this survey as a generic term for distance, similarity or dissimilarity function—between data points plays an important role in many machine learning, pattern recognition and data mining techniques.[1] For instance, in classification, the $k$-Nearest Neighbor classifier (Cover and Hart, 1967) uses a metric to identify the nearest neighbors; many clustering algorithms, such as the prominent $K$-Means (Lloyd, 1982), rely on distance measurements between data points; in information retrieval, doc-

∗. Most of the work in this paper was carried out while the author was affiliated with Laboratoire Hubert Curien UMR 5516, Université de Saint-Etienne, France.

1. Metric-based learning methods were the focus of the recent SIMBAD European project (ICT 2008-FET 2008-2011). Website: http://simbad-fp7.eu/

uments are often ranked according to their relevance to a given query based on similarity scores. Clearly, the performance of these methods depends on the quality of the metric: as in the saying "birds of a feather flock together", we hope that it identifies as similar (resp. dissimilar) the pairs of instances that are indeed semantically close (resp. different). General-purpose metrics exist (e.g., the Euclidean distance and the cosine similarity for feature vectors or the Levenshtein distance for strings) but they often fail to capture the idiosyncrasies of the data of interest. Improved results are expected when the metric is designed specifically for the task at hand. Since manual tuning is difficult and tedious, a lot of effort has gone into *metric learning*, the research topic devoted to automatically learning metrics from data.

### 1.1 Metric Learning in a Nutshell

Although its origins can be traced back to some earlier work (e.g., Short and Fukunaga, 1981; Fukunaga, 1990; Friedman, 1994; Hastie and Tibshirani, 1996; Baxter and Bartlett, 1997), metric learning really emerged in 2002 with the pioneering work of Xing et al. (2002) that formulates it as a convex optimization problem. It has since been a hot research topic, being the subject of tutorials at ICML 2010[2] and ECCV 2010[3] and workshops at ICCV 2011,[4] NIPS 2011[5] and ICML 2013.[6]

The goal of metric learning is to adapt some pairwise real-valued metric function, say the Mahalanobis distance $d_{\boldsymbol{M}}(\boldsymbol{x}, \boldsymbol{x'}) = \sqrt{(\boldsymbol{x} - \boldsymbol{x'})^T \boldsymbol{M} (\boldsymbol{x} - \boldsymbol{x'})}$, to the problem of interest using the information brought by training examples. Most methods learn the metric (here, the positive semi-definite matrix $\boldsymbol{M}$ in $d_{\boldsymbol{M}}$) in a weakly-supervised way from pair or triplet-based constraints of the following form:

- Must-link / cannot-link constraints (sometimes called positive / negative pairs):

$$\begin{aligned} \mathcal{S} &= \{(x_i, x_j) : x_i \text{ and } x_j \text{ should be similar}\}, \\ \mathcal{D} &= \{(x_i, x_j) : x_i \text{ and } x_j \text{ should be dissimilar}\}. \end{aligned}$$

- Relative constraints (sometimes called training triplets):

$$\mathcal{R} = \{(x_i, x_j, x_k) : x_i \text{ should be more similar to } x_j \text{ than to } x_k\}.$$

A metric learning algorithm basically aims at finding the parameters of the metric such that it best agrees with these constraints (see Figure 1 for an illustration), in an effort to approximate the underlying semantic metric. This is typically formulated as an optimization problem that has the following general form:

$$\min_{\boldsymbol{M}} \quad \ell(\boldsymbol{M}, \mathcal{S}, \mathcal{D}, \mathcal{R}) \quad + \quad \lambda R(\boldsymbol{M})$$

where $\ell(\boldsymbol{M}, \mathcal{S}, \mathcal{D}, \mathcal{R})$ is a loss function that incurs a penalty when training constraints are violated, $R(\boldsymbol{M})$ is some regularizer on the parameters $\boldsymbol{M}$ of the learned metric and

2. http://www.icml2010.org/tutorials.html
3. http://www.ics.forth.gr/eccv2010/tutorials.php
4. http://www.iccv2011.org/authors/workshops/
5. http://nips.cc/Conferences/2011/Program/schedule.php?Session=Workshops
6. http://icml.cc/2013/?page_id=41

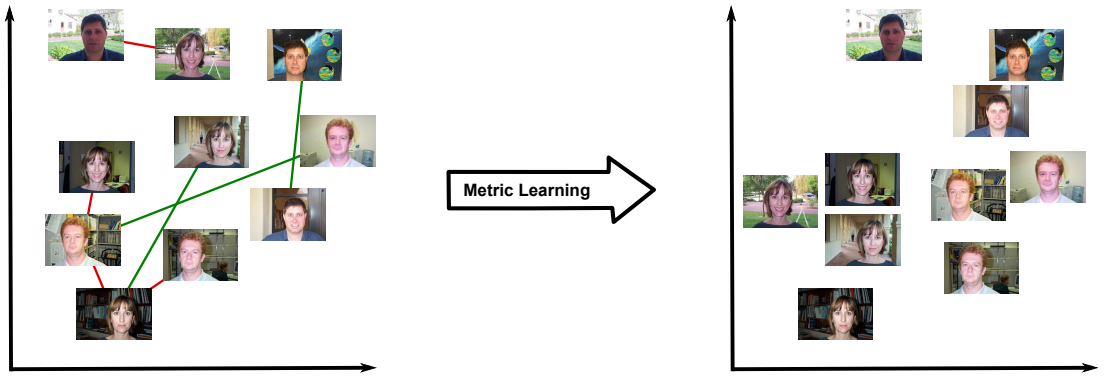


Figure 1: Illustration of metric learning applied to a face recognition task. For simplicity, images are represented as points in 2 dimensions. Pairwise constraints, shown in the left pane, are composed of images representing the same person (must-link, shown in green) or different persons (cannot-link, shown in red). We wish to adapt the metric so that there are fewer constraint violations (right pane). Images are taken from the Caltech Faces dataset.[8]

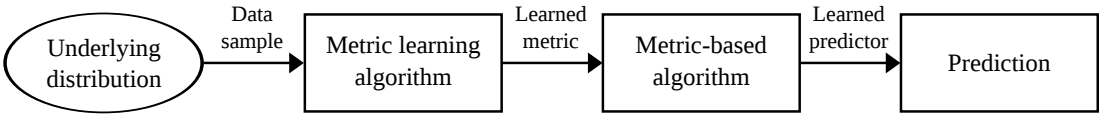


Figure 2: The common process in metric learning. A metric is learned from training data and plugged into an algorithm that outputs a predictor (e.g., a classifier, a regressor, a recommender system...) which hopefully performs better than a predictor induced by a standard (non-learned) metric.

$\lambda \geq 0$ is the regularization parameter. As we will see in this survey, state-of-the-art metric learning formulations essentially differ by their choice of metric, constraints, loss function and regularizer.

After the metric learning phase, the resulting function is used to improve the performance of a metric-based algorithm, which is most often $k$-Nearest Neighbors ($k$-NN), but may also be a clustering algorithm such as $K$-Means, a ranking algorithm, etc. The common process in metric learning is summarized in Figure 2.

### 1.2 Applications

Metric learning can potentially be beneficial whenever the notion of metric between instances plays an important role. Recently, it has been applied to problems as diverse as link prediction in networks (Shaw et al., 2011), state representation in reinforcement learning (Taylor et al., 2011), music recommendation (McFee et al., 2012), partitioning problems

8. http://www.vision.caltech.edu/html-files/archive.html

(Lajugie et al., 2014), identity verification (Ben et al., 2012), webpage archiving (Law et al., 2012), cartoon synthesis (Yu et al., 2012) and even assessing the efficacy of acupuncture (Liang et al., 2012), to name a few. In the following, we list three large fields of application where metric learning has been shown to be very useful.

**Computer vision** There is a great need of appropriate metrics in computer vision, not only to compare images or videos in ad-hoc representations—such as bags-of-visual-words (Li and Perona, 2005)—but also in the pre-processing step consisting in building this very representation (for instance, visual words are usually obtained by means of clustering). For this reason, there exists a large body of metric learning literature dealing specifically with computer vision problems, such as image classification (Mensink et al., 2012), object recognition (Frome et al., 2007; Verma et al., 2012), face recognition (Guillaumin et al., 2009b; Lu et al., 2012), visual tracking (Li et al., 2012; Jiang et al., 2012) or image annotation (Guillaumin et al., 2009a).

**Information retrieval** The objective of many information retrieval systems, such as search engines, is to provide the user with the most relevant documents according to his/her query. This ranking is often achieved by using a metric between two documents or between a document and a query. Applications of metric learning to these settings include the work of Lebanon (2006); Lee et al. (2008); McFee and Lanckriet (2010); Lim et al. (2013).

**Bioinformatics** Many problems in bioinformatics involve comparing sequences such as DNA, protein or temporal series. These comparisons are based on structured metrics such as edit distance measures (or related string alignment scores) for strings or Dynamic Time Warping distance for temporal series. Learning these metrics to adapt them to the task of interest can greatly improve the results. Examples include the work of Xiong and Chen (2006); Saigo et al. (2006); Kato and Nagano (2010); Wang et al. (2012a).

### 1.3 Related Topics

We mention here three research topics that are related to metric learning but outside the scope of this survey.

**Kernel learning** While metric learning is parametric (one learns the parameters of a given form of metric, such as a Mahalanobis distance), kernel learning is usually nonparametric: one learns the kernel matrix without any assumption on the form of the kernel that implicitly generated it. These approaches are thus very powerful but limited to the transductive setting and can hardly be applied to new data. The interested reader may refer to the recent survey on kernel learning by Abbasnejad et al. (2012).

**Multiple kernel learning** Unlike kernel learning, Multiple Kernel Learning (MKL) is parametric: it learns a combination of predefined base kernels. In this regard, it can be seen as more restrictive than metric or kernel learning, but as opposed to kernel learning, MKL has very efficient formulations and can be applied in the inductive setting. The interested reader may refer to the recent survey on MKL by Gönen and Alpaydin (2011).

**Dimensionality reduction** Supervised dimensionality reduction aims at finding a low-dimensional representation that maximizes the separation of labeled data and in this respect

has connections with metric learning,[9] although the primary objective is quite different. Unsupervised dimensionality reduction, or manifold learning, usually assume that the (unlabeled) data lie on an embedded low-dimensional manifold within the higher-dimensional space and aim at "unfolding" it. These methods aim at capturing or preserving some properties of the original data (such as the variance or local distance measurements) in the low-dimensional representation.[10] The interested reader may refer to the surveys by Fodor (2002) and van der Maaten et al. (2009).

### 1.4 Why this Survey?

As pointed out above, metric learning has been a hot topic of research in machine learning for a few years and has now reached a considerable level of maturity both practically and theoretically. The early review due to Yang and Jin (2006) is now largely outdated as it misses out on important recent advances: more than 75% of the work referenced in the present survey is post 2006. A more recent survey, written independently and in parallel to our work, is due to Kulis (2012). Despite some overlap, it should be noted that both surveys have their own strengths and complement each other well. Indeed, the survey of Kulis takes a more general approach, attempting to provide a unified view of a few core metric learning methods. It also goes into depth about topics that are only briefly reviewed here, such as kernelization, optimization methods and applications. On the other hand, the present survey is a detailed and comprehensive review of the existing literature, covering more than 50 approaches (including many recent works that are missing from Kulis' paper) with their relative merits and drawbacks. Furthermore, we give particular attention to topics that are not covered by Kulis, such as metric learning for structured data and the derivation of generalization guarantees.

We think that the present survey may foster novel research in metric learning and be useful to a variety of audiences, in particular: (i) machine learners wanting to get introduced to or update their knowledge of metric learning will be able to quickly grasp the pros and cons of each method as well as the current strengths and limitations of the research area as a whole, and (ii) machine learning practitioners interested in applying metric learning to their own problem will find information to help them choose the methods most appropriate to their needs, along with links to source codes whenever available.

Note that we focus on general-purpose methods, i.e., that are applicable to a wide range of application domains. The abundant literature on metric learning designed specifically for computer vision is not addressed because the understanding of these approaches requires a significant amount of background in that area. For this reason, we think that they deserve a separate survey, targeted at the computer vision audience.

### 1.5 Prerequisites

This survey is almost self-contained and has few prerequisites. For metric learning from feature vectors, we assume that the reader has some basic knowledge of linear algebra

9. Some metric learning methods can be seen as finding a new feature space, and a few of them actually have the additional goal of making this feature space low-dimensional.
10. These approaches are sometimes referred to as "unsupervised metric learning", which is somewhat misleading because they do not optimize a notion of metric.

| Notation | Description |
|---|---|
| $\mathbb{R}$ | Set of real numbers |
| $\mathbb{R}^d$ | Set of $d$-dimensional real-valued vectors |
| $\mathbb{R}^{c\times d}$ | Set of $c \times d$ real-valued matrices |
| $\mathbb{S}_+^d$ | Cone of symmetric PSD $d \times d$ real-valued matrices |
| $\mathcal{X}$ | Input (instance) space |
| $\mathcal{Y}$ | Output (label) space |
| $\mathcal{S}$ | Set of must-link constraints |
| $\mathcal{D}$ | Set of cannot-link constraints |
| $\mathcal{R}$ | Set of relative constraints |
| $z = (x, y) \in \mathcal{X} \times \mathcal{Y}$ | An arbitrary labeled instance |
| $\boldsymbol{x}$ | An arbitrary vector |
| $\boldsymbol{M}$ | An arbitrary matrix |
| $\boldsymbol{I}$ | Identity matrix |
| $\boldsymbol{M} \succeq 0$ | PSD matrix $\boldsymbol{M}$ |
| $\|\cdot\|_p$ | $p$-norm |
| $\|\cdot\|_{\mathcal{F}}$ | Frobenius norm |
| $\|\cdot\|_*$ | Nuclear norm |
| $\mathrm{tr}(\boldsymbol{M})$ | Trace of matrix $\boldsymbol{M}$ |
| $[t]_+ = \max(0, 1 - t)$ | Hinge loss function |
| $\xi$ | Slack variable |
| $\Sigma$ | Finite alphabet |
| $\times$ | String of finite size |

Table 1: Summary of the main notations.

and convex optimization (if needed, see Boyd and Vandenberghe, 2004, for a brush-up). For metric learning from structured data, we assume that the reader has some familiarity with basic probability theory, statistics and likelihood maximization. The notations used throughout this survey are summarized in Table 1.

### 1.6 Outline

The rest of this paper is organized as follows. We first assume that data consist of vectors lying in some feature space $\mathcal{X} \subseteq \mathbb{R}^d$. Section 2 describes key properties that we will use to provide a taxonomy of metric learning algorithms. In Section 3, we review the large body of work dealing with supervised Mahalanobis distance learning. Section 4 deals with recent advances and trends in the field, such as linear similarity learning, nonlinear and local methods, histogram distance learning, the derivation of generalization guarantees and semi-supervised metric learning methods. We cover metric learning for structured data in Section 5, with a focus on edit distance learning. Lastly, we conclude this survey in Section 6 with a discussion on the current limitations of the existing literature and promising directions for future research.

## 2. Key Properties of Metric Learning Algorithms

Except for a few early methods, most metric learning algorithms are essentially "competitive" in the sense that they are able to achieve state-of-the-art performance on some problems. However, each algorithm has its intrinsic properties (e.g., type of metric, ability to leverage unsupervised data, good scalability with dimensionality, generalization guaran-

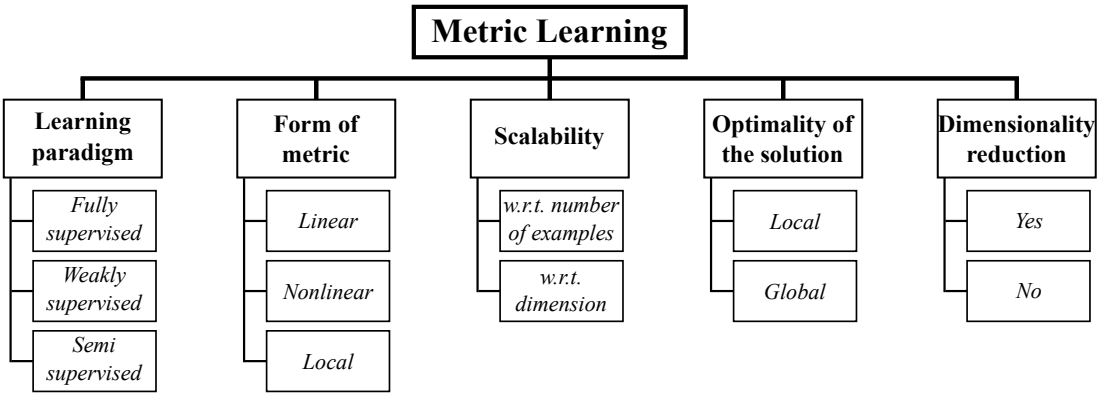


Figure 3: Five key properties of metric learning algorithms.

tees, etc) and emphasis should be placed on those when deciding which method to apply to a given problem. In this section, we identify and describe five key properties of metric learning algorithms, summarized in Figure 3. We use them to provide a taxonomy of the existing literature: the main features of each method are given in Table 2.[11]

**Learning Paradigm** We will consider three learning paradigms:

- *Fully supervised*: the metric learning algorithm has access to a set of labeled training instances $\{z_i = (x_i, y_i)\}_{i=1}^n$, where each training example $z_i \in \mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ is composed of an instance $x_i \in \mathcal{X}$ and a label (or class) $y_i \in \mathcal{Y}$. $\mathcal{Y}$ is a discrete and finite set of $|\mathcal{Y}|$ labels (unless stated otherwise). In practice, the label information is often used to generate specific sets of pair/triplet constraints $\mathcal{S}, \mathcal{D}, \mathcal{R}$, for instance based on a notion of neighborhood.[12]

- *Weakly supervised*: the algorithm has no access to the labels of individual training instances: it is only provided with side information in the form of sets of constraints $\mathcal{S}, \mathcal{D}, \mathcal{R}$. This is a meaningful setting in a variety of applications where labeled data is costly to obtain while such side information is cheap: examples include users' implicit feedback (e.g., clicks on search engine results), citations among articles or links in a network. This can be seen as having label information only at the pair/triplet level.

- *Semi-supervised*: besides the (full or weak) supervision, the algorithm has access to a (typically large) sample of unlabeled instances for which no side information is available. This is useful to avoid overfitting when the labeled data or side information are scarce.

**Form of Metric** Clearly, the form of the learned metric is a key choice. One may identify three main families of metrics:

11. Whenever possible, we use the acronyms provided by the authors of the studied methods. When there is no known acronym, we take the liberty of choosing one.
12. These constraints are usually derived from the labels prior to learning the metric and never challenged. Note that Wang et al. (2012b) propose a more refined (but costly) approach to the problem of building the constraints from labels. Their method alternates between selecting the most relevant constraints given the current metric and learning a new metric based on the current constraints.

- *Linear metrics*, such as the Mahalanobis distance. Their expressive power is limited but they are easier to optimize (they usually lead to convex formulations, and thus global optimality of the solution) and less prone to overfitting.
- *Nonlinear metrics*, such as the $\chi^2$ histogram distance. They often give rise to nonconvex formulations (subject to local optimality) and may overfit, but they can capture nonlinear variations in the data.
- *Local metrics*, where multiple (linear or nonlinear) local metrics are learned (typically simultaneously) to better deal with complex problems, such as heterogeneous data. They are however more prone to overfitting than global methods since the number of parameters they learn can be very large.

**Scalability** With the amount of available data growing fast, the problem of scalability arises in all areas of machine learning. First, it is desirable for a metric learning algorithm to scale well with the number of training examples $n$ (or constraints). As we will see, learning the metric in an online way is one of the solutions. Second, metric learning methods should also scale reasonably well with the dimensionality $d$ of the data. However, since metric learning is often phrased as learning a $d \times d$ matrix, designing algorithms that scale reasonably well with this quantity is a considerable challenge.

**Optimality of the Solution** This property refers to the ability of the algorithm to find the parameters of the metric that satisfy best the criterion of interest. Ideally, the solution is guaranteed to be the *global optimum*—this is essentially the case for convex formulations of metric learning. On the contrary, for nonconvex formulations, the solution may only be a *local optimum*.

**Dimensionality Reduction** As noted earlier, metric learning is sometimes formulated as finding a projection of the data into a new feature space. An interesting byproduct in this case is to look for a low-dimensional projected space, allowing faster computations as well as more compact representations. This is typically achieved by forcing or regularizing the learned metric matrix to be low-rank.

## 3. Supervised Mahalanobis Distance Learning

This section deals with (fully or weakly) supervised Malahanobis distance learning (sometimes simply referred to as distance metric learning), which has attracted a lot of interest due to its simplicity and nice interpretation in terms of a linear projection. We start by presenting the Mahalanobis distance and two important challenges associated with learning this form of metric.

**The Mahalanobis distance** This term comes from Mahalanobis (1936) and originally refers to a distance measure that incorporates the correlation between features:

$$d_{maha}(\boldsymbol{x}, \boldsymbol{x'}) = \sqrt{(\boldsymbol{x} - \boldsymbol{x'})^T \boldsymbol{\Omega}^{-1} (\boldsymbol{x} - \boldsymbol{x'})},$$

where $\mathbf{x}$ and $\mathbf{x}'$ are random vectors from the same distribution with covariance matrix $\boldsymbol{\Omega}$. By an abuse of terminology common in the metric learning literature, we will in fact use

| Page | Name | Year | Source Code | Supervision | Form of Metric | Scalability w.r.t. $n$ | Scalability w.r.t. $d$ | Optimum | Dimension Reduction | Regularizer | Additional Information |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 11 | MMC | 2002 | Yes | Weak | Linear | ★☆☆ | ☆☆☆ | Global | No | None | — |
| 11 | S&J | 2003 | No | Weak | Linear | ★★☆ | ★★★ | Global | No | Frobenius norm | — |
| 12 | NCA | 2004 | Yes | Full | Linear | ★☆☆ | ★★☆ | Local | Yes | None | For $k$-NN |
| 12 | MCML | 2005 | Yes | Full | Linear | ★☆☆ | ☆☆☆ | Global | No | None | For $k$-NN |
| 13 | LMNN | 2005 | Yes | Full | Linear | ★★☆ | ★☆☆ | Global | No | None | For $k$-NN |
| 13 | RCA | 2003 | Yes | Weak | Linear | ★★☆ | ★★☆ | Global | No | None | — |
| 14 | ITML | 2007 | Yes | Weak | Linear | ★☆☆ | ★★☆ | Global | No | LogDet | Online version |
| 14 | SDML | 2009 | No | Weak | Linear | ★☆☆ | ★★☆ | Global | No | LogDet+$L_1$ | $n \ll d$ |
| 15 | POLA | 2004 | No | Weak | Linear | ★★★ | ★☆☆ | Global | No | None | Online |
| 15 | LEGO | 2008 | No | Weak | Linear | ★★★ | ★★☆ | Global | No | LogDet | Online |
| 15 | RDML | 2009 | No | Weak | Linear | ★★★ | ★★☆ | Global | No | Frobenius norm | Online |
| 16 | MDML | 2012 | No | Weak | Linear | ★★★ | ★☆☆ | Global | Yes | Nuclear norm | Online |
| 16 | mt-LMNN | 2010 | Yes | Full | Linear | ★★☆ | ☆☆☆ | Global | No | Frobenius norm | Multi-task |
| 17 | MLCS | 2011 | No | Weak | Linear | ★☆☆ | ★★☆ | Local | Yes | N/A | Multi-task |
| 17 | GPML | 2012 | No | Weak | Linear | ★☆☆ | ★★☆ | Global | Yes | von Neumann | Multi-task |
| 18 | TML | 2010 | Yes | Weak | Linear | ★★☆ | ★★☆ | Global | No | Frobenius norm | Transfer learning |
| 18 | LPML | 2006 | No | Weak | Linear | ★★☆ | ★★☆ | Global | Yes | $L_1$ norm | — |
| 19 | SML | 2009 | No | Weak | Linear | ★☆☆ | ☆☆☆ | Global | Yes | $L_{2,1}$ norm | — |
| 19 | BoostMetric | 2009 | Yes | Weak | Linear | ★☆☆ | ★★☆ | Global | Yes | None | — |
| 19 | DML-$p$ | 2012 | No | Weak | Linear | ★☆☆ | ★☆☆ | Global | No | None | — |
| 20 | RML | 2010 | No | Weak | Linear | ★★☆ | ☆☆☆ | Global | No | Frobenius norm | Noisy constraints |
| 20 | MLR | 2010 | Yes | Full | Linear | ★★☆ | ☆☆☆ | Global | Yes | Nuclear norm | For ranking |
| 21 | SiLA | 2008 | No | Full | Linear | ★★☆ | ★★☆ | N/A | No | None | Online |
| 22 | gCosLA | 2009 | No | Weak | Linear | ★★★ | ☆☆☆ | Global | No | None | Online |
| 22 | OASIS | 2009 | Yes | Weak | Linear | ★★★ | ★★☆ | Global | No | Frobenius norm | Online |
| 23 | SLLC | 2012 | No | Full | Linear | ★★☆ | ★★☆ | Global | No | Frobenius norm | For linear classif. |
| 23 | RSL | 2013 | No | Full | Linear | ★☆☆ | ★★☆ | Local | No | Frobenius norm | Rectangular matrix |
| 25 | LSMD | 2005 | No | Weak | Nonlinear | ★☆☆ | ★★☆ | Local | Yes | None | — |
| 25 | NNCA | 2007 | No | Full | Nonlinear | ★☆☆ | ★★☆ | Local | Yes | Recons. error | — |
| 25 | SVML | 2012 | No | Full | Nonlinear | ★☆☆ | ★★☆ | Local | Yes | Frobenius norm | For SVM |
| 25 | GB-LMNN | 2012 | No | Full | Nonlinear | ★★☆ | ★★☆ | Local | Yes | None | — |
| 26 | HDML | 2012 | Yes | Weak | Nonlinear | ★★☆ | ★★☆ | Local | Yes | $L_2$ norm | Hamming distance |
| 27 | $M^2$-LMNN | 2008 | Yes | Full | Local | ★★☆ | ★☆☆ | Global | No | None | — |
| 27 | GLML | 2010 | No | Full | Local | ★★★ | ★★☆ | Global | No | Diagonal | Generative |
| 27 | Bk-means | 2009 | No | Weak | Local | ★☆☆ | ★★★ | Global | No | RKHS norm | Bregman dist. |
| 28 | PLML | 2012 | Yes | Weak | Local | ★★☆ | ☆☆☆ | Global | No | Manifold+Frob | — |
| 29 | RFD | 2012 | Yes | Weak | Local | ★★☆ | ★★★ | N/A | No | None | Random forests |
| 30 | $\chi^2$-LMNN | 2012 | No | Full | Nonlinear | ★★☆ | ★★☆ | Local | Yes | None | Histogram data |
| 30 | GML | 2011 | No | Weak | Linear | ★☆☆ | ★★☆ | Local | No | None | Histogram data |
| 31 | EMDL | 2012 | No | Weak | Linear | ★☆☆ | ★★☆ | Local | No | Frobenius norm | Histogram data |
| 33 | LRML | 2008 | Yes | Semi | Linear | ★☆☆ | ☆☆☆ | Global | No | Laplacian | — |
| 34 | M-DML | 2009 | No | Semi | Linear | ★☆☆ | ☆☆☆ | Local | No | Laplacian | Auxiliary metrics |
| 34 | SERAPH | 2012 | Yes | Semi | Linear | ★☆☆ | ☆☆☆ | Local | Yes | Trace+entropy | Probabilistic |
| 35 | CDML | 2011 | No | Semi | N/A | N/A | N/A | N/A | N/A | N/A | Domain adaptation |
| 35 | DAML | 2011 | No | Semi | Nonlinear | ★☆☆ | ☆☆☆ | Global | No | MMD | Domain adaptation |

Table 2: Main features of metric learning methods for feature vectors. Scalability levels are relative and given as a rough guide.

the term Mahalanobis distance to refer to generalized quadratic distances, defined as

$$d_{\boldsymbol{M}}(\boldsymbol{x}, \boldsymbol{x'}) = \sqrt{(\boldsymbol{x} - \boldsymbol{x'})^T \boldsymbol{M} (\boldsymbol{x} - \boldsymbol{x'})}$$

and parameterized by $\boldsymbol{M} \in \mathbb{S}_+^d$, where $\mathbb{S}_+^d$ is the cone of symmetric positive semi-definite (PSD) $d \times d$ real-valued matrices (see Figure 4).[13] $\boldsymbol{M} \in \mathbb{S}_+^d$ ensures that $d_{\boldsymbol{M}}$ satisfies the properties of a pseudo-distance: $\forall \boldsymbol{x}, \boldsymbol{x'}, \boldsymbol{x''} \in \mathcal{X}$,

1. $d_{\boldsymbol{M}}(\boldsymbol{x}, \boldsymbol{x'}) \geq 0$ (nonnegativity),
2. $d_{\boldsymbol{M}}(\boldsymbol{x}, \boldsymbol{x}) = 0$ (identity),
3. $d_{\boldsymbol{M}}(\boldsymbol{x}, \boldsymbol{x'}) = d(\boldsymbol{x'}, \boldsymbol{x})$ (symmetry),
4. $d_{\boldsymbol{M}}(\boldsymbol{x}, \boldsymbol{x''}) \leq d(\boldsymbol{x}, \boldsymbol{x'}) + d(\boldsymbol{x'}, \boldsymbol{x''})$ (triangle inequality).

**Interpretation** Note that when $\boldsymbol{M}$ is the identity matrix, we recover the Euclidean distance. Otherwise, one can express $\boldsymbol{M}$ as $\boldsymbol{L}^T\boldsymbol{L}$, where $\boldsymbol{L} \in \mathbb{R}^{k \times d}$ where $k$ is the rank of $\boldsymbol{M}$. We can then rewrite $d_{\boldsymbol{M}}(\boldsymbol{x}, \boldsymbol{x'})$ as follows:

$$\begin{aligned} d_{\boldsymbol{M}}(\boldsymbol{x}, \boldsymbol{x'}) &= \sqrt{(\boldsymbol{x} - \boldsymbol{x'})^T \boldsymbol{M} (\boldsymbol{x} - \boldsymbol{x'})} \\ &= \sqrt{(\boldsymbol{x} - \boldsymbol{x'})^T \boldsymbol{L}^T \boldsymbol{L} (\boldsymbol{x} - \boldsymbol{x'})} \\ &= \sqrt{(\boldsymbol{L}\boldsymbol{x} - \boldsymbol{L}\boldsymbol{x'})^T (\boldsymbol{L}\boldsymbol{x} - \boldsymbol{L}\boldsymbol{x'})}. \end{aligned}$$

Thus, a Mahalanobis distance implicitly corresponds to computing the Euclidean distance after the linear projection of the data defined by the transformation matrix $\boldsymbol{L}$. Note that if $\boldsymbol{M}$ is low-rank, i.e., $\text{rank}(\boldsymbol{M}) = r < d$, then it induces a linear projection of the data into a space of lower dimension $r$. It thus allows a more compact representation of the data and cheaper distance computations, especially when the original feature space is high-dimensional. These nice properties explain why learning Mahalanobis distance has attracted a lot of interest and is a major component of metric learning.

**Challenges** This leads us to two important challenges associated with learning Mahalanobis distances. The first one is to maintain $\boldsymbol{M} \in \mathbb{S}_+^d$ in an efficient way during the optimization process. A simple way to do this is to use the projected gradient method which consists in alternating between a gradient step and a projection step onto the PSD cone by setting the negative eigenvalues to zero.[14] However this is expensive for high-dimensional problems as eigenvalue decomposition scales in $O(d^3)$. The second challenge is to learn a low-rank matrix (which implies a low-dimensional projection space, as noted earlier) instead of a full-rank one. Unfortunately, optimizing $\boldsymbol{M}$ subject to a rank constraint or regularization is NP-hard and thus cannot be carried out efficiently.

13. Note that in practice, to get rid of the square root, the Mahalanobis distance is learned in its more convenient squared form $d^2_{\boldsymbol{M}}(\boldsymbol{x}, \boldsymbol{x'}) = (\boldsymbol{x} - \boldsymbol{x'})^T \boldsymbol{M} (\boldsymbol{x} - \boldsymbol{x'})$.
14. Note that Qian et al. (2013) have proposed some heuristics to avoid doing this projection at each iteration.

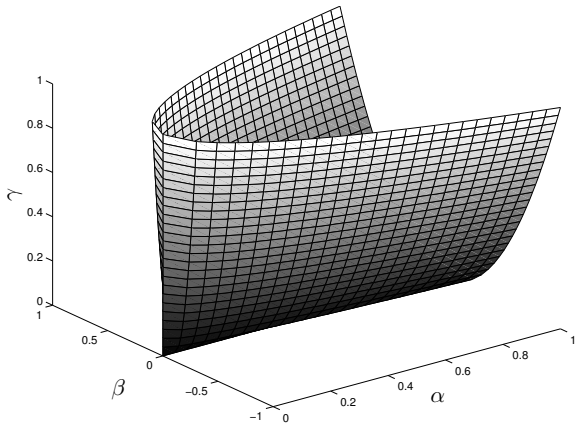


Figure 4: The cone $\mathbb{S}_+^2$ of positive semi-definite 2x2 matrices of the form $\begin{bmatrix} \alpha & \beta \\ \beta & \gamma \end{bmatrix}$.

The rest of this section is a comprehensive review of the supervised Mahalanobis distance learning methods of the literature. We first present two early approaches (Section 3.1). We then discuss methods that are specific to $k$-nearest neighbors (Section 3.2), inspired from information theory (Section 3.3), online learning approaches (Section 3.4), multi-task learning (Section 3.5) and a few more that do not fit any of the previous categories (Section 3.6).

### 3.1 Early Approaches

The approaches in this section deal with the PSD constraint in a rudimentary way.

**MMC (Xing et al.)** The seminal work of Xing et al. (2002) is the first Mahalanobis distance learning method.[15] It relies on a convex formulation with no regularization, which aims at maximizing the sum of distances between dissimilar points while keeping the sum of distances between similar examples small:

$$\begin{aligned} \max_{\boldsymbol{M} \in \mathbb{S}_+^d} \quad & \sum_{(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{D}} d_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_j) \\ \text{s.t.} \quad & \sum_{(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{S}} d^2_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_j) \leq 1. \end{aligned} \tag{1}$$

The algorithm used to solve (1) is a simple projected gradient approach requiring the full eigenvalue decomposition of $\boldsymbol{M}$ at each iteration. This is typically intractable for medium and high-dimensional problems.

**S&J (Schultz & Joachims)** The method proposed by Schultz and Joachims (2003) relies on the parameterization $\boldsymbol{M} = \boldsymbol{A}^T \boldsymbol{W} \boldsymbol{A}$, where $\boldsymbol{A}$ is fixed and known and $\boldsymbol{W}$ diagonal. We get:

$$d^2_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{A}\boldsymbol{x}_i - \boldsymbol{A}\boldsymbol{x}_j)^T \boldsymbol{W} (\boldsymbol{A}\boldsymbol{x}_i - \boldsymbol{A}\boldsymbol{x}_j).$$

15. Source code available at: http://www.cs.cmu.edu/~epxing/papers/

By definition, $\boldsymbol{M}$ is PSD and thus one can optimize over the diagonal matrix $\boldsymbol{W}$ and avoid the need for costly projections on the PSD cone. They propose a formulation based on triplet constraints:

$$\begin{aligned} \min_{\boldsymbol{W}} \quad & \|\boldsymbol{M}\|_{\mathcal{F}}^2 \quad + \quad C \sum_{i,j,k} \xi_{ijk} \\ \text{s.t.} \quad & d_{\boldsymbol{M}}^2(\boldsymbol{x}_i, \boldsymbol{x}_k) - d_{\boldsymbol{M}}^2(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 1 - \xi_{ijk} \quad \forall (\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_k) \in \mathcal{R}, \end{aligned} \tag{2}$$

where $\|\boldsymbol{M}\|_{\mathcal{F}}^2 = \sum_{i,j} M_{ij}^2$ is the squared Frobenius norm of $\boldsymbol{M}$, the $\xi_{ijk}$'s are "slack" variables to allow soft constraints[16] and $C \geq 0$ is the trade-off parameter between regularization and constraint satisfaction. Problem (2) is convex and can be solved efficiently. The main drawback of this approach is that it is less general than full Mahalanobis distance learning: one only learns a weighting $\boldsymbol{W}$ of the features. Furthermore, $\boldsymbol{A}$ must be chosen manually.

### 3.2 Approaches Driven by Nearest Neighbors

The objective functions of the methods presented in this section are related to a nearest neighbor prediction rule.

**NCA (Goldberger et al.)** The idea of Neighbourhood Component Analysis[17] (NCA), introduced by Goldberger et al. (2004), is to optimize the expected leave-one-out error of a stochastic nearest neighbor classifier in the projection space induced by $d_{\boldsymbol{M}}$. They use the decomposition $\boldsymbol{M} = \boldsymbol{L}^T\boldsymbol{L}$ and they define the probability that $\boldsymbol{x}_i$ is the neighbor of $\boldsymbol{x}_j$ by

$$p_{ij} = \frac{\exp(-\|\boldsymbol{L}\boldsymbol{x}_i - \boldsymbol{L}\boldsymbol{x}_j\|_2^2)}{\sum_{l \neq i} \exp(-\|\boldsymbol{L}\boldsymbol{x}_i - \boldsymbol{L}\boldsymbol{x}_l\|_2^2)}, \quad p_{ii} = 0.$$

Then, the probability that $\boldsymbol{x}_i$ is correctly classified is:

$$p_i = \sum_{j : y_j = y_i} p_{ij}.$$

They learn the distance by solving:

$$\max_{L} \sum_i p_i. \tag{3}$$

Note that the matrix $\boldsymbol{L}$ can be chosen to be rectangular, inducing a low-rank $\boldsymbol{M}$. The main limitation of (3) is that it is nonconvex and thus subject to local maxima. Hong et al. (2011) later proposed to learn a mixture of NCA metrics, while Tarlow et al. (2013) generalize NCA to $k$-NN with $k > 1$.

**MCML (Globerson & Roweis)** Shortly after Goldberger et al., Globerson and Roweis (2005) proposed MCML (Maximally Collapsing Metric Learning), an alternative convex formulation based on minimizing a KL divergence between $p_{ij}$ and an ideal distribution,

16. This is a classic trick used for instance in soft-margin SVM (Cortes and Vapnik, 1995). Throughout this survey, we will consistently use the symbol $\xi$ to denote slack variables.
17. Source code available at: http://www.ics.uci.edu/~fowlkes/software/nca/

which can be seen as attempting to collapse each class to a single point.[18] Unlike NCA, the optimization is done with respect to the matrix $\boldsymbol{M}$ and the problem is thus convex. However, like MMC, MCML requires costly projections onto the PSD cone.

**LMNN (Weinberger et al.)** Large Margin Nearest Neighbors[19] (LMNN), introduced by Weinberger et al. (2005; 2008; 2009), is one of the most widely-used Mahalanobis distance learning methods and has been the subject of many extensions (described in later sections). One of the reasons for its popularity is that the constraints are defined in a local way: the $k$ nearest neighbors (the "target neighbors") of any training instance should belong to the correct class while keeping away instances of other classes (the "impostors"). The Euclidean distance is used to determine the target neighbors. Formally, the constraints are defined in the following way:

$$\begin{aligned}\mathcal{S} &= \{(\boldsymbol{x}_i, \boldsymbol{x}_j) : y_i = y_j \text{ and } \boldsymbol{x}_j \text{ belongs to the } k\text{-neighborhood of } \boldsymbol{x}_i\},\\ \mathcal{R} &= \{(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_k) : (\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{S}, y_i \neq y_k\}.\end{aligned}$$

The distance is learned using the following convex program:

$$\begin{aligned}\min_{\boldsymbol{M} \in \mathbb{S}_+^d} \quad & (1-\mu) \sum_{(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{S}} d^2_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_j) \quad + \quad \mu \sum_{i,j,k} \xi_{ijk} \\ \text{s.t.} \quad & d^2_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_k) - d^2_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 1 - \xi_{ijk} \qquad \forall (\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_k) \in \mathcal{R},\end{aligned} \tag{4}$$

where $\mu \in [0, 1]$ controls the "pull/push" trade-off. The authors developed a special-purpose solver—based on subgradient descent and careful book-keeping—that is able to deal with billions of constraints. Alternative ways of solving the problem have been proposed (Torresani and Lee, 2006; Nguyen and Guo, 2008; Park et al., 2011; Der and Saul, 2012). LMNN generally performs very well in practice, although it is sometimes prone to overfitting due to the absence of regularization, especially in high dimension. It is also very sensitive to the ability of the Euclidean distance to select relevant target neighbors. Note that Do et al. (2012) highlighted a relation between LMNN and Support Vector Machines.

### 3.3 Information-Theoretic Approaches

The methods presented in this section frame metric learning as an optimization problem involving an information measure.

**RCA (Bar-Hillel et al.)** Relevant Component Analysis[20] (Shental et al., 2002; Bar-Hillel et al., 2003, 2005) makes use of positive pairs only and is based on subsets of the training examples called "chunklets". These are obtained from the set of positive pairs by applying a transitive closure: for instance, if $(\boldsymbol{x}_1, \boldsymbol{x}_2) \in \mathcal{S}$ and $(\boldsymbol{x}_2, \boldsymbol{x}_3) \in \mathcal{S}$, then $\boldsymbol{x}_1$, $\boldsymbol{x}_2$ and $\boldsymbol{x}_3$ belong to the same chunklet. Points in a chunklet are believed to share the same label. Assuming a total of $n$ points in $k$ chunklets, the algorithm is very efficient since it

18. An implementation is available within the Matlab Toolbox for Dimensionality Reduction: http://homepage.tudelft.nl/19j49/Matlab_Toolbox_for_Dimensionality_Reduction.html
19. Source code available at: http://www.cse.wustl.edu/~kilian/code/code.html
20. Source code available at: http://www.scharp.org/thertz/code.html

simply amounts to computing the following matrix:

$$\hat{\boldsymbol{C}} = \frac{1}{n}\sum_{j=1}^{k}\sum_{i=1}^{n_j}(x_{ji} - \hat{m}_j)(x_{ji} - \hat{m}_j)^T,$$

where chunklet $j$ consists of $\{x_{ji}\}_{i=1}^{n_j}$ and $\hat{m}_j$ is its mean. Thus, RCA essentially reduces the within-chunklet variability in an effort to identify features that are irrelevant to the task. The inverse of $\hat{\boldsymbol{C}}$ is used in a Mahalanobis distance. The authors have shown that (i) it is the optimal solution to an information-theoretic criterion involving a mutual information measure, and (ii) it is also the optimal solution to the optimization problem consisting in minimizing the within-class distances. An obvious limitation of RCA is that it cannot make use of the discriminative information brought by negative pairs, which explains why it is not very competitive in practice. RCA was later extended to handle negative pairs, at the cost of a more expensive algorithm (Hoi et al., 2006; Yeung and Chang, 2006).

**ITML (Davis et al.)** Information-Theoretic Metric Learning[21] (ITML), proposed by Davis et al. (2007), is an important work because it introduces LogDet divergence regularization that will later be used in several other Mahalanobis distance learning methods (e.g., Jain et al., 2008; Qi et al., 2009). This Bregman divergence on positive definite matrices is defined as:

$$D_{ld}(\boldsymbol{M}, \boldsymbol{M}_0) = \mathrm{tr}(\boldsymbol{M}\boldsymbol{M}_0^{-1}) - \log\det(\boldsymbol{M}\boldsymbol{M}_0^{-1}) - d,$$

where $d$ is the dimension of the input space and $\boldsymbol{M}_0$ is some positive definite matrix we want to remain close to. In practice, $\boldsymbol{M}_0$ is often set to $\boldsymbol{I}$ (the identity matrix) and thus the regularization aims at keeping the learned distance close to the Euclidean distance. The key feature of the LogDet divergence is that it is finite if and only if $\boldsymbol{M}$ is positive definite. Therefore, minimizing $D_{ld}(\boldsymbol{M}, \boldsymbol{M}_0)$ provides an automatic and cheap way of preserving the positive semi-definiteness of $\boldsymbol{M}$. ITML is formulated as follows:

$$\begin{array}{lll} \min\limits_{\boldsymbol{M}\in\mathbb{S}_+^d} & D_{ld}(\boldsymbol{M}, \boldsymbol{M}_0) \quad + \quad \gamma\sum\limits_{i,j}\xi_{ij} & \\ \text{s.t.} & d^2_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_j) \leq u + \xi_{ij} & \forall(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{S} \\ & d^2_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq v - \xi_{ij} & \forall(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{D}, \end{array} \tag{5}$$

where $u, v \in \mathbb{R}$ are threshold parameters and $\gamma \geq 0$ the trade-off parameter. ITML thus aims at satisfying the similarity and dissimilarity constraints while staying as close as possible to the Euclidean distance (if $\boldsymbol{M}_0 = \boldsymbol{I}$). More precisely, the information-theoretic interpretation behind minimizing $D_{ld}(\boldsymbol{M}, \boldsymbol{M}_0)$ is that it is equivalent to minimizing the KL divergence between two multivariate Gaussian distributions parameterized by $\boldsymbol{M}$ and $\boldsymbol{M}_0$. The algorithm proposed to solve (5) is efficient, converges to the global minimum and the resulting distance performs well in practice. A limitation of ITML is that $\boldsymbol{M}_0$, that must be picked by hand, can have an important influence on the quality of the learned distance. Note that Kulis et al. (2009) have shown how hashing can be used together with ITML to achieve fast similarity search.

21. Source code available at: http://www.cs.utexas.edu/~pjain/itml/

**SDML (Qi et al.)** With Sparse Distance Metric Learning (SDML), Qi et al. (2009) specifically deal with the case of high-dimensional data together with few training samples, i.e., $n \ll d$. To avoid overfitting, they use a double regularization: the LogDet divergence (using $\boldsymbol{M}_0 = \boldsymbol{I}$ or $\boldsymbol{M}_0 = \boldsymbol{\Omega}^{-1}$ where $\boldsymbol{\Omega}$ is the covariance matrix) and $L_1$-regularization on the off-diagonal elements of $\boldsymbol{M}$. The justification for using this $L_1$-regularization is two-fold: (i) a practical one is that in high-dimensional spaces, the off-diagonal elements of $\boldsymbol{\Omega}^{-1}$ are often very small, and (ii) a theoretical one suggested by a consistency result from a previous work in covariance matrix estimation (Ravikumar et al., 2011) that applies to SDML. They use a fast algorithm based on block-coordinate descent (the optimization is done over each row of $\boldsymbol{M}^{-1}$) and obtain very good performance for the specific case $n \ll d$.

### 3.4 Online Approaches

In online learning (Littlestone, 1988), the algorithm receives training instances one at a time and updates at each step the current hypothesis. Although the performance of online algorithms is typically inferior to batch algorithms, they are very useful to tackle large-scale problems that batch methods fail to address due to time and space complexity issues. Online learning methods often come with regret bounds, stating that the accumulated loss suffered along the way is not much worse than that of the best hypothesis chosen in hindsight.[22]

**POLA (Shalev-Shwartz et al.)** POLA (Shalev-Shwartz et al., 2004), for Pseudo-metric Online Learning Algorithm, is the first online Mahalanobis distance learning approach and learns the matrix $\boldsymbol{M}$ as well as a threshold $b \geq 1$. At each step $t$, POLA receives a pair $(\boldsymbol{x}_i, \boldsymbol{x}_j, y_{ij})$, where $y_{ij} = 1$ if $(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{S}$ and $y_{ij} = -1$ if $(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{D}$, and performs two successive orthogonal projections:

1. Projection of the current solution $(\boldsymbol{M}^{t-1}, b^{t-1})$ onto the set $C_1 = \{(\boldsymbol{M}, b) \in \mathbb{R}^{d^2+1} : [y_{ij}(d^2_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_j) - b) + 1]_+ = 0\}$, which is done efficiently (closed-form solution). The constraint basically requires that the distance between two instances of same (resp. different) labels be below (resp. above) the threshold $b$ with a margin 1. We get an intermediate solution $(\boldsymbol{M}^{t-\frac{1}{2}}, b^{t-\frac{1}{2}})$ that satisfies this constraint while staying as close as possible to the previous solution.

2. Projection of $(\boldsymbol{M}^{t-\frac{1}{2}}, b^{t-\frac{1}{2}})$ onto the set $C_2 = \{(\boldsymbol{M}, b) \in \mathbb{R}^{d^2+1} : \boldsymbol{M} \in \mathbb{S}^d_+, b \geq 1\}$, which is done rather efficiently (in the worst case, one only needs to compute the minimal eigenvalue of $\boldsymbol{M}^{t-\frac{1}{2}}$). This projects the matrix back onto the PSD cone. We thus get a new solution $(\boldsymbol{M}^t, b^t)$ that yields a valid Mahalanobis distance.

A regret bound for the algorithm is provided.

**LEGO (Jain et al.)** LEGO (Logdet Exact Gradient Online), developed by Jain et al. (2008), is an improved version of POLA based on LogDet divergence regularization. It features tighter regret bounds, more efficient updates and better practical performance.

22. A regret bound has the following general form: $\sum_{t=1}^{T} \ell(h_t, z_t) - \sum_{t=1}^{T} \ell(h^*, z_t) \leq O(T)$, where $T$ is the number of steps, $h_t$ is the hypothesis at time $t$ and $h^*$ is the best batch hypothesis.

**RDML (Jin et al.)** RDML (Jin et al., 2009) is similar to POLA in spirit but is more flexible. At each step $t$, instead of forcing the margin constraint to be satisfied, it performs a gradient descent step of the following form (assuming Frobenius regularization):

$$\boldsymbol{M}^t = \pi_{\mathbb{S}_+^d}\left(\boldsymbol{M}^{t-1} - \lambda y_{ij}(\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^T\right),$$

where $\pi_{\mathbb{S}_+^d}(\cdot)$ is the projection to the PSD cone. The parameter $\lambda$ implements a trade-off between satisfying the pairwise constraint and staying close to the previous matrix $\boldsymbol{M}^{t-1}$. Using some linear algebra, the authors show that this update can be performed by solving a convex quadratic program instead of resorting to eigenvalue computation like POLA. RDML is evaluated on several benchmark datasets and is shown to perform comparably to LMNN and ITML.

**MDML (Kunapuli & Shavlik)** MDML (Kunapuli and Shavlik, 2012), for Mirror Descent Metric Learning, is an attempt of proposing a general framework for online Mahalanobis distance learning. It is based on composite mirror descent (Duchi et al., 2010), which allows online optimization of many regularized problems. It can accommodate a large class of loss functions and regularizers for which efficient updates are derived, and the algorithm comes with a regret bound. Their study focuses on regularization with the nuclear norm (also called trace norm) introduced by Fazel et al. (2001) and defined as $\|\boldsymbol{M}\|_* = \sum_i \sigma_i$, where the $\sigma_i$'s are the singular values of $\boldsymbol{M}$.[23] It is known to be the best convex relaxation of the rank of the matrix and thus nuclear norm regularization tends to induce low-rank matrices. In practice, MDML has performance comparable to LMNN and ITML, is fast and sometimes induces low-rank solutions, but surprisingly the algorithm was not evaluated on large-scale datasets.

### 3.5 Multi-Task Metric Learning

This section covers Mahalanobis distance learning for the multi-task setting (Caruana, 1997), where given a set of related tasks, one learns a metric for each in a coupled fashion in order to improve the performance on all tasks.

**mt-LMNN (Parameswaran & Weinberger)** Multi-Task LMNN[24] (Parameswaran and Weinberger, 2010) is a straightforward adaptation of the ideas of Multi-Task SVM (Evgeniou and Pontil, 2004) to metric learning. Given $T$ related tasks, they model the problem as learning a shared Mahalanobis metric $d_{\boldsymbol{M}_0}$ as well as task-specific metrics $d_{\boldsymbol{M}_1}, \ldots, d_{\boldsymbol{M}_t}$ and define the metric for task $t$ as

$$d_t(\boldsymbol{x}, \boldsymbol{x'}) = (\boldsymbol{x} - \boldsymbol{x'})^T(\boldsymbol{M}_0 + \boldsymbol{M}_t)(\boldsymbol{x} - \boldsymbol{x'}).$$

Note that $\boldsymbol{M}_0 + \boldsymbol{M}_t \succeq 0$, hence $d_t$ is a valid pseudo-metric. The LMNN formulation is easily generalized to this multi-task setting so as to learn the metrics jointly, with a specific regularization term defined as follows:

$$\gamma_0\|\boldsymbol{M}_0 - \boldsymbol{I}\|_{\mathcal{F}}^2 + \sum_{t=1}^{T}\gamma_t\|\boldsymbol{M}_t\|_{\mathcal{F}}^2,$$

23. Note that when $\boldsymbol{M} \in \mathbb{S}_+^d$, $\|\boldsymbol{M}\|_* = \mathrm{tr}(\boldsymbol{M}) = \sum_{i=1}^d M_{ii}$, which is much cheaper to compute.
24. Source code available at: http://www.cse.wustl.edu/~kilian/code/code.html

where $\gamma_t$ controls the regularization of $\boldsymbol{M}_t$. When $\gamma_0 \to \infty$, the shared metric $d_{\boldsymbol{M}_0}$ is simply the Euclidean distance, and the formulation reduces to $T$ independent LMNN formulations. On the other hand, when $\gamma_{t>0} \to \infty$, the task-specific matrices are simply zero matrices and the formulation reduces to LMNN on the union of all data. In-between these extreme cases, these parameters can be used to adjust the relative importance of each metric: $\gamma_0$ to set the overall level of shared information, and $\gamma_t$ to set the importance of $\boldsymbol{M}_t$ with respect to the shared metric. The formulation remains convex and can be solved using the same efficient solver as LMNN. In the multi-task setting, mt-LMNN clearly outperforms single-task metric learning methods and other multi-task classification techniques such as mt-SVM.

**MLCS (Yang et al.)** MLCS (Yang et al., 2011) is a different approach to the problem of multi-task metric learning. For each task $t \in \{1, \dots, T\}$, the authors consider learning a Mahalanobis metric

$$d^2_{\boldsymbol{L}_t^T \boldsymbol{L}_t}(\boldsymbol{x}, \boldsymbol{x'}) = (\boldsymbol{x} - \boldsymbol{x'})^T \boldsymbol{L}_t^T \boldsymbol{L}_t (\boldsymbol{x} - \boldsymbol{x'}) = (\boldsymbol{L}_t \boldsymbol{x} - \boldsymbol{L}_t \boldsymbol{x'})^T (\boldsymbol{L}_t \boldsymbol{x} - \boldsymbol{L}_t \boldsymbol{x'})$$

parameterized by the transformation matrix $\boldsymbol{L}_t \in \mathbb{R}^{r \times d}$. They show that $\boldsymbol{L}_t$ can be decomposed into a "subspace" part $\boldsymbol{L}_0^t \in \mathbb{R}^{r \times d}$ and a "low-dimensional metric" part $\boldsymbol{R}_t \in \mathbb{R}^{r \times r}$ such that $\boldsymbol{L}_t = \boldsymbol{R}_t \boldsymbol{L}_0^t$. The main assumption of MLCS is that all tasks share a common subspace, i.e., $\forall t$, $\boldsymbol{L}_0^t = \boldsymbol{L}_0$. This parameterization can be used to extend most of metric learning methods to the multi-task setting, although it breaks the convexity of the formulation and is thus subject to local optima. However, as opposed to mt-LMNN, it can be made low-rank by setting $r < d$ and thus has many less parameters to learn. In their work, MLCS is applied to the version of LMNN solved with respect to the transformation matrix (Torresani and Lee, 2006). The resulting method is evaluated on problems with very scarce training data and study the performance for different values of $r$. It is shown to outperform mt-LMNN, but the setup is a bit unfair to mt-LMNN since it is forced to be low-rank by eigenvalue thresholding.

**GPML (Yang et al.)** The work of Yang et al. (2012) identifies two drawbacks of previous multi-task metric learning approaches: (i) MLCS's assumption of common subspace is sometimes too strict and leads to a nonconvex formulation, and (ii) the Frobenius regularization of mt-LMNN does not preserve geometry. This property is defined as being the ability to propagate side-information: the task-specific metrics should be regularized so as to preserve the relative distance between training pairs. They introduce the following formulation, which extends any metric learning algorithm to the multi-task setting:

$$\min_{\boldsymbol{M}_0, \dots, \boldsymbol{M}_t \in \mathbb{S}_+^d} \quad \sum_{i=1}^{t} \left( \ell(\boldsymbol{M}_t, \mathcal{S}_t, \mathcal{D}_t, \mathcal{R}_t) + \gamma d_\varphi(\boldsymbol{M}_t, \boldsymbol{M}_0) \right) \quad + \quad \gamma_0 d_\varphi(\boldsymbol{A}_0, \boldsymbol{M}_0), \tag{6}$$

where $\ell(\boldsymbol{M}_t, \mathcal{S}_t, \mathcal{D}_t, \mathcal{R}_t)$ is the loss function for the task $t$ based on the training pairs/triplets (depending on the chosen algorithm), $d_\varphi(\boldsymbol{A}, \boldsymbol{B}) = \varphi(\boldsymbol{A}) - \varphi(\boldsymbol{B}) - \operatorname{tr}\left((\nabla \varphi \boldsymbol{B})^T (\boldsymbol{A} - \boldsymbol{B})\right)$ is a Bregman matrix divergence (Dhillon and Tropp, 2007) and $\boldsymbol{A}_0$ is a predefined metric (e.g., the identity matrix $\boldsymbol{I}$). mt-LMNN can essentially be recovered from (6) by setting $\varphi(\boldsymbol{A}) = \|\boldsymbol{A}\|_{\mathcal{F}}^2$ and additional constraints $\boldsymbol{M}_t \succeq \boldsymbol{M}_0$. The authors focus on the von

Neumann divergence:

$$d_{VN}(\boldsymbol{A}, \boldsymbol{B}) = \mathrm{tr}(\boldsymbol{A}\log\boldsymbol{A} - \boldsymbol{A}\log\boldsymbol{B} - \boldsymbol{A} + \boldsymbol{B}),$$

where $\log \boldsymbol{A}$ is the matrix logarithm of $\boldsymbol{A}$. Like the LogDet divergence mentioned earlier in this survey (Section 3.3), the von Neumann divergence is known to be rank-preserving and to provide automatic enforcement of positive-semidefiniteness. The authors further show that minimizing this divergence encourages geometry preservation between the learned metrics. Problem (6) remains convex as long as the original algorithm used for solving each task is convex, and can be solved efficiently using gradient descent methods. In the experiments, the method is adapted to LMNN and outperforms single-task LMNN as well as mt-LMNN, especially when training data is very scarce.

**TML (Zhang & Yeung)** Zhang and Yeung (2010) propose a transfer metric learning (TML) approach.[25] They assume that we are given $S$ independent source tasks with enough labeled data and that a Mahalanobis distance $\boldsymbol{M}_s$ has been learned for each task $s$. The goal is to leverage the information of the source metrics to learn a distance $\boldsymbol{M}_t$ for a target task, for which we only have a scarce amount $n_t$ of labeled data. No assumption is made about the relation between the source tasks and the target task: they may be positively/negatively correlated or uncorrelated. The problem is formulated as follows:

$$\begin{aligned} \min_{\boldsymbol{M}_t \in \mathbb{S}_+^d, \boldsymbol{\Omega} \succeq 0} \quad & \frac{2}{n_t^2} \sum_{i<j} \ell\left(y_i y_j \left[1 - d^2_{\boldsymbol{M}_t}(\boldsymbol{x}_i, \boldsymbol{x}_j)\right]\right) + \frac{\lambda_1}{2}\|\boldsymbol{M}_t\|_{\mathcal{F}}^2 + \frac{\lambda_2}{2}\,\mathrm{tr}(\tilde{\boldsymbol{M}}\boldsymbol{\Omega}^{-1}\tilde{\boldsymbol{M}}^T) \\ \text{s.t.} \quad & \mathrm{tr}(\boldsymbol{\Omega}) = 1, \end{aligned} \tag{7}$$

where $\ell(t) = \max(0, 1-t)$ is the hinge loss, $\tilde{\boldsymbol{M}} = (\mathrm{vec}(\boldsymbol{M}_1), \ldots, \mathrm{vec}(\boldsymbol{M}_s), \mathrm{vec}(\boldsymbol{M}_t))$. The first two terms are classic (loss on all possible pairs and Frobenius regularization) while the third one models the relation between tasks based on a positive definite covariance matrix $\boldsymbol{\Omega}$. Assuming that the source tasks are independent and of equal importance, $\boldsymbol{\Omega}$ can be expressed as

$$\boldsymbol{\Omega} = \begin{pmatrix} \alpha \boldsymbol{I}^{(m-1)\times(m-1)} & \boldsymbol{\omega}_m \\ \boldsymbol{\omega}_m & \omega \end{pmatrix},$$

where $\boldsymbol{\omega}_m$ denotes the task covariances between the target task and the source tasks, and $\omega$ denotes the variance of the target task. Problem (7) is convex and is solved using an alternating procedure that is guaranteed to converge to the global optimum: (i) fixing $\boldsymbol{\Omega}$ and solving for $\boldsymbol{M}_t$, which is done online with an algorithm similar to RDML, and (ii) fixing $\boldsymbol{M}_t$ and solving for $\boldsymbol{\Omega}$, leading to a second-order cone program whose number of variables and constraints is linear in the number of tasks. In practice, TML consistently outperforms metric learning methods without transfer when training data is scarce.

### 3.6 Other Approaches

In this section, we describe a few approaches that are outside the scope of the previous categories. The first two (LPML and SML) fall into the category of sparse metric learning

25. Source code available at: http://www.cse.ust.hk/~dyyeung/

methods. BoostMetric is inspired from the theory of boosting. DML-$p$ revisits the original metric learning formulation of Xing et al. RML deals with the presence of noisy constraints. Finally, MLR learns a metric for solving a ranking task.

**LPML (Rosales & Fung)** The method of Rosales and Fung (2006) aims at learning matrices with entire columns/rows set to zero, thus making $\boldsymbol{M}$ low-rank. For this purpose, they use $L_1$ norm regularization and, restricting their framework to diagonal dominant matrices, they are able to formulate the problem as a linear program that can be solved efficiently. However, $L_1$ norm regularization favors sparsity at the entry level only, not specifically at the row/column level, even though in practice the learned matrix is sometimes low-rank. Furthermore, the approach is less general than Mahalanobis distances due to the restriction to diagonal dominant matrices.

**SML (Ying et al.)** SML[26] (Ying et al., 2009), for Sparse Metric Learning, is a distance learning approach that regularizes $\boldsymbol{M}$ with the mixed $L_{2,1}$ norm defined as

$$\|\boldsymbol{M}\|_{2,1} = \sum_{i=1}^{d} \|\boldsymbol{M}_i\|_2,$$

which tends to zero out entire rows of $\boldsymbol{M}$ (as opposed to the $L_1$ norm used in LPML), and therefore performs feature selection. More precisely, they set $\boldsymbol{M} = \boldsymbol{U}^T \boldsymbol{W} \boldsymbol{U}$, where $\boldsymbol{U} \in \mathbb{O}^d$ (the set of $d \times d$ orthonormal matrices) and $\boldsymbol{W} \in \mathbb{S}_+^d$, and solve the following problem:

$$\begin{aligned} \min_{\boldsymbol{U}\in\mathbb{O}^d, \boldsymbol{W}\in\mathbb{S}_+^d} \quad & \|\boldsymbol{W}\|_{2,1} \quad + \quad \gamma \sum_{i,j,k} \xi_{ijk} \\ \text{s.t.} \quad & d^2_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_k) - d^2_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 1 - \xi_{ijk} \quad \forall (\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_k) \in \mathcal{R}, \end{aligned} \tag{8}$$

where $\gamma \geq 0$ is the trade-off parameter. Unfortunately, $L_{2,1}$ regularized problems are typically difficult to optimize. Problem (8) is reformulated as a min-max problem and solved using smooth optimization (Nesterov, 2005). Overall, the algorithm has a fast convergence rate but each iteration has an $O(d^3)$ complexity. The method performs well in practice while achieving better dimensionality reduction than full-rank methods such as Rosales and Fung (2006). However, it cannot be applied to high-dimensional problems due to the complexity of the algorithm. Note that the same authors proposed a unified framework for sparse metric learning (Huang et al., 2009, 2011).

**BoostMetric (Shen et al.)** BoostMetric[27] (Shen et al., 2009, 2012) adapts to Mahalanobis distance learning the ideas of boosting, where a good hypothesis is obtained through a weighted combination of so-called "weak learners" (see the recent book on this matter by Schapire and Freund, 2012). The method is based on the property that any PSD matrix can be decomposed into a positive linear combination of trace-one rank-one matrices. This kind of matrices is thus used as weak learner and the authors adapt the popular boosting algorithm Adaboost (Freund and Schapire, 1995) to this setting. The resulting algorithm

26. Source code is not available but is indicated as "coming soon" by the authors. Check: http://www.enm.bris.ac.uk/staff/xyy/software.html
27. Source code available at: http://code.google.com/p/boosting/

is quite efficient since it does not require full eigenvalue decomposition but only the computation of the largest eigenvalue. In practice, BoostMetric achieves competitive performance but typically requires a very large number of iterations for high-dimensional datasets. Bi et al. (2011) further improve the scalability of the approach, while Liu and Vemuri (2012) introduce regularization on the weights as well as a term to reduce redundancy among the weak learners.

**DML-$p$ (Ying et al., Cao et al.)** The work of Ying and Li (2012); Cao et al. (2012b) revisit MMC, the original approach of Xing et al. (2002), by investigating the following formulation, called DML-$p$:

$$\begin{aligned} \max_{\boldsymbol{M} \in \mathbb{S}_+^d} \quad & \left( \frac{1}{|\mathcal{D}|} \sum_{(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{D}} [d_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_j)]^{2p} \right)^{1/p} \\ \text{s.t.} \quad & \sum_{(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{S}} d^2_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_j) \leq 1. \end{aligned} \tag{9}$$

Note that by setting $p = 0.5$ we recover MMC. The authors show that (9) is convex for $p \in (-\infty, 1)$ and can be cast as a well-known eigenvalue optimization problem called "minimizing the maximal eigenvalue of a symmetric matrix". They further show that it can be solved efficiently using a first-order algorithm that only requires the computation of the largest eigenvalue at each iteration (instead of the costly full eigen-decomposition used by Xing et al.). Experiments show competitive results and low computational complexity. A general drawback of DML-$p$ is that it is not clear how to accommodate a regularizer (e.g., sparse or low-rank).

**RML (Huang et al.)** Robust Metric Learning (Huang et al., 2010) is a method that can successfully deal with the presence of noisy/incorrect training constraints, a situation that can arise when they are not derived from class labels but from side information such as users' implicit feedback. The approach is based on robust optimization (Ben-Tal et al., 2009): assuming that a proportion $1 - \eta$ of the $m$ training constraints (say triplets) are incorrect, it minimizes some loss function $\ell$ for any $\eta$ fraction of the triplets:

$$\begin{aligned} \min_{\boldsymbol{M} \in \mathbb{S}_+^d, t} \quad & t \quad + \quad \frac{\lambda}{2} \|\boldsymbol{M}\|_{\mathcal{F}} \\ \text{s.t.} \quad & t \geq \sum_{i=1}^{m} q_i \ell \left( d^2_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_i'') - d^2_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_i') \right), \quad \forall \boldsymbol{q} \in \mathcal{Q}(\eta), \end{aligned} \tag{10}$$

where $\ell$ is taken to be the hinge loss and $\mathcal{Q}(\eta)$ is defined as

$$\mathcal{Q}(\eta) = \left\{ \boldsymbol{q} \in \{0,1\}^m : \sum_{i=1}^{m} q_i \leq \eta m \right\}.$$

In other words, Problem (10) minimizes the worst-case violation over all possible sets of correct constraints. $\mathcal{Q}(\eta)$ can be replaced by its convex hull, leading to a semi-definite program with an infinite number of constraints. This can be further simplified into a

convex minimization problem that can be solved either using subgradient descent or smooth optimization (Nesterov, 2005). However, both of these require a projection onto the PSD cone. Experiments on standard datasets show good robustness for up to 30% of incorrect triplets, while the performance of other methods such as LMNN is greatly damaged.

**MLR (McFee & Lankriet)** The idea of MLR (McFee and Lanckriet, 2010), for Metric Learning to Rank, is to learn a metric for a ranking task, where given a query instance, one aims at producing a ranked list of examples where relevant ones are ranked higher than irrelevant ones.[28] Let $\mathcal{P}$ the set of all permutations (i.e., possible rankings) over the training set. Given a Mahalanobis distance $d^2_{\boldsymbol{M}}$ and a query $\boldsymbol{x}$, the predicted ranking $p \in \mathcal{P}$ consists in sorting the instances by ascending $d^2_{\boldsymbol{M}}(\boldsymbol{x}, \cdot)$. The metric learning $\boldsymbol{M}$ is based on Structural SVM (Tsochantaridis et al., 2005):

$$\begin{aligned} \min_{\boldsymbol{M} \in \mathbb{S}^d_+} \quad & \|\boldsymbol{M}\|_* \quad + \quad C \sum_i \xi_i \\ \text{s.t.} \quad & \langle \boldsymbol{M}, \psi(\boldsymbol{x}_i, p_i) - \psi(\boldsymbol{x}_i, p) \rangle_{\mathcal{F}} \geq \Delta(p_i, p) - \xi_i \quad \forall i \in \{1, \dots, n\}, p \in \mathcal{P}, \end{aligned} \tag{11}$$

where $\|\boldsymbol{M}\|_* = \text{tr}(\boldsymbol{M})$ is the nuclear norm, $C \geq 0$ the trade-off parameter, $\langle \boldsymbol{A}, \boldsymbol{B} \rangle_{\mathcal{F}} = \sum_{i,j} A_{ij} B_{ij}$ the Frobenius inner product, $\psi : \mathbb{R} \times \mathcal{P} \to \mathbb{S}^d$ the feature encoding of an input-output pair $(\boldsymbol{x}_i, p)$,[29] and $\Delta(p_i, p) \in [0, 1]$ the "margin" representing the loss of predicting ranking $p$ instead of the true ranking $p_i$. In other words, $\Delta(p_i, p)$ assesses the quality of ranking $p$ with respect to the best ranking $p_i$ and can be evaluated using several measures, such as the Area Under the ROC Curve (AUC), Precision-at-$k$ or Mean Average Precision (MAP). Since the number of constraints is super-exponential in the number of training instances, the authors solve (11) using a 1-slack cutting-plane approach (Joachims et al., 2009) which essentially iteratively optimizes over a small set of active constraints (adding the most violated ones at each step) using subgradient descent. However, the algorithm requires a full eigendecomposition of $\boldsymbol{M}$ at each iteration, thus MLR does not scale well with the dimensionality of the data. In practice, it is competitive with other metric learning algorithms for $k$-NN classification and a structural SVM algorithm for ranking, and can induce low-rank solutions due to the nuclear norm. Lim et al. (2013) propose R-MLR, an extension to MLR to deal with the presence of noisy features[30] using the mixed $L_{2,1}$ norm as in SML (Ying et al., 2009). R-MLR is shown to be able to ignore most of the irrelevant features and outperforms MLR in this situation.

## 4. Other Advances in Metric Learning

So far, we focused on (linear) Mahalanobis metric learning which has inspired a large amount of work during the past ten years. In this section, we cover other advances and trends in metric learning for feature vectors. Most of the section is devoted to (fully and weakly) supervised methods. In Section 4.1, we address linear similarity learning. Section 4.2 deals

28. Source code is available at: `http://www-cse.ucsd.edu/~bmcfee/code/mlr`
29. The feature map $\psi$ is designed such that the ranking $p$ which maximizes $\langle \boldsymbol{M}, \psi(\boldsymbol{x}, p) \rangle_{\mathcal{F}}$ is the one given by ascending $d^2_{\boldsymbol{M}}(\boldsymbol{x}, \cdot)$.
30. Notice that this is different from noisy side information, which was investigated by the method RML (Huang et al., 2010) presented earlier in this section.

with nonlinear metric learning (including the kernelization of linear methods), Section 4.3 with local metric learning and Section 4.4 with metric learning for histogram data. Section 4.5 presents the recently-developed frameworks for deriving generalization guarantees for supervised metric learning. We conclude this section with a review of semi-supervised metric learning (Section 4.6).

### 4.1 Linear Similarity Learning

Although most of the work in linear metric learning has focused on the Mahalanobis distance, other linear measures, in the form of similarity functions, have recently attracted some interest. These approaches are often motivated by the perspective of more scalable algorithms due to the absence of PSD constraint.

**SiLA (Qamar et al.)** SiLA (Qamar et al., 2008) is an approach for learning similarity functions of the following form:

$$K_{\boldsymbol{M}}(\boldsymbol{x}, \boldsymbol{x'}) = \frac{\boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x'}}{N(\boldsymbol{x}, \boldsymbol{x'})},$$

where $\boldsymbol{M} \in \mathbb{R}^{d \times d}$ is not required to be PSD nor symmetric, and $N(\boldsymbol{x}, \boldsymbol{x'})$ is a normalization term which depends on $\boldsymbol{x}$ and $\boldsymbol{x'}$. This similarity function can be seen as a generalization of the cosine similarity, widely used in text and image retrieval (see for instance Baeza-Yates and Ribeiro-Neto, 1999; Sivic and Zisserman, 2009). The authors build on the same idea of "target neighbors" that was introduced in LMNN, but optimize the similarity in an online manner with an algorithm based on voted perceptron. At each step, the algorithm goes through the training set, updating the matrix when an example does not satisfy a criterion of separation. The authors present theoretical results that follow from the voted perceptron theory in the form of regret bounds for the separable and inseparable cases. In subsequent work, Qamar and Gaussier (2012) study the relationship between SiLA and RELIEF, an online feature reweighting algorithm.

**gCosLA (Qamar & Gaussier)** gCosLA (Qamar and Gaussier, 2009) learns generalized cosine similarities of the form

$$K_{\boldsymbol{M}}(\boldsymbol{x}, \boldsymbol{x'}) = \frac{\boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x'}}{\sqrt{\boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x}} \sqrt{\boldsymbol{x'}^T \boldsymbol{M} \boldsymbol{x'}}},$$

where $\boldsymbol{M} \in \mathbb{S}_+^d$. It corresponds to a cosine similarity in the projection space implied by $\boldsymbol{M}$. The algorithm itself, an online procedure, is very similar to that of POLA (presented in Section 3.4). Indeed, they essentially use the same loss function and also have a two-step approach: a projection onto the set of arbitrary matrices that achieve zero loss on the current example pair, followed by a projection back onto the PSD cone. The first projection is different from POLA (since the generalized cosine has a normalization factor that depends on $\boldsymbol{M}$) but the authors manage to derive a closed-form solution. The second projection is based on a full eigenvalue decomposition of $\boldsymbol{M}$, making the approach costly as dimensionality grows. A regret bound for the algorithm is provided and it is shown experimentally that gCosLA converges in fewer iterations than SiLA and is generally more accurate. Its performance is competitive with LMNN and ITML. Note that Nguyen and Bai (2010) optimize the same form of similarity based on a nonconvex formulation.

**OASIS (Chechik et al.)** OASIS[31] (Chechik et al., 2009, 2010) learns a bilinear similarity with a focus on large-scale problems. The bilinear similarity has been used for instance in image retrieval (Deng et al., 2011) and has the following simple form:

$$K_{\boldsymbol{M}}(\boldsymbol{x}, \boldsymbol{x'}) = \boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x'},$$

where $\boldsymbol{M} \in \mathbb{R}^{d \times d}$ is not required to be PSD nor symmetric. In other words, it is related to the (generalized) cosine similarity but does not include normalization nor PSD constraint. Note that when $\boldsymbol{M}$ is the identity matrix, $K_{\boldsymbol{M}}$ amounts to an unnormalized cosine similarity. The bilinear similarity has two advantages. First, it is efficiently computable for sparse inputs: if $\boldsymbol{x}$ and $\boldsymbol{x'}$ have $k_1$ and $k_2$ nonzero features, $K_{\boldsymbol{M}}(\boldsymbol{x}, \boldsymbol{x'})$ can be computed in $O(k_1 k_2)$ time. Second, unlike the Mahalanobis distance, it can define a similarity measure between instances of different dimension (for example, a document and a query) if a rectangular matrix $\boldsymbol{M}$ is used. Since $\boldsymbol{M} \in \mathbb{R}^{d \times d}$ is not required to be PSD, Chechik et al. are able to optimize $K_{\boldsymbol{M}}$ in an online manner using a simple and efficient algorithm, which belongs to the family of Passive-Aggressive algorithms (Crammer et al., 2006). The initialization is $\boldsymbol{M} = \boldsymbol{I}$, then at each step $t$, the algorithm draws a triplet $(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_k) \in \mathcal{R}$ and solves the following convex problem:

$$\begin{aligned} \boldsymbol{M}^t = \quad & \underset{\boldsymbol{M}, \xi}{\arg\min} \quad \frac{1}{2}\|\boldsymbol{M} - \boldsymbol{M}^{t-1}\|_{\mathcal{F}}^2 + C\xi \\ & \text{s.t.} \quad 1 - d_{\boldsymbol{M}}^2(\boldsymbol{x}_i, \boldsymbol{x}_j) + d_{\boldsymbol{M}}^2(\boldsymbol{x}_i, \boldsymbol{x}_k) \leq \xi \\ & \qquad\quad \xi \geq 0, \end{aligned} \tag{12}$$

where $C \geq 0$ is the trade-off parameter between minimizing the loss and staying close from the matrix obtained at the previous step. Clearly, if $1 - d_{\boldsymbol{M}}^2(\boldsymbol{x}_i, \boldsymbol{x}_j) + d_{\boldsymbol{M}}^2(\boldsymbol{x}_i, \boldsymbol{x}_k) \leq 0$, then $\boldsymbol{M}^t = \boldsymbol{M}^{t-1}$ is the solution of (12). Otherwise, the solution is obtained from a simple closed-form update. In practice, OASIS achieves competitive results on medium-scale problems and unlike most other methods, is scalable to problems with millions of training instances. However, it cannot incorporate complex regularizers. Note that the same authors derived two more algorithms for learning bilinear similarities as applications of more general frameworks. The first one is based on online learning in the manifold of low-rank matrices (Shalit et al., 2010, 2012) and the second on adaptive regularization of weight matrices (Crammer and Chechik, 2012).

**SLLC (Bellet et al.)** Similarity Learning for Linear Classification (Bellet et al., 2012b) takes an original angle by focusing on metric learning for linear classification. As opposed to pair and triplet-based constraints used in other approaches, the metric is optimized to be $(\epsilon, \gamma, \tau)$-good (Balcan et al., 2008a), a property based on an average over some points which has a deep connection with the performance of a sparse linear classifier built from such a similarity. SLLC learns a bilinear similarity $K_{\boldsymbol{M}}$ and is formulated as an efficient unconstrained quadratic program:

$$\min_{\boldsymbol{M} \in \mathbb{R}^{d \times d}} \quad \frac{1}{n} \sum_{i=1}^{n} \ell\left(1 - y_i \frac{1}{\gamma |\mathcal{R}|} \sum_{\boldsymbol{x}_j \in \mathcal{R}} y_j K_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_j)\right) \quad + \quad \beta \|\boldsymbol{M}\|_{\mathcal{F}}^2, \tag{13}$$

31. Source code available at: http://ai.stanford.edu/~gal/Research/OASIS/

where $\mathcal{R}$ is a set of reference points randomly selected from the training sample, $\gamma$ is the margin parameter, $\ell$ is the hinge loss and $\beta$ the regularization parameter. Problem (13) essentially learns $K_{\boldsymbol{M}}$ such that training examples are more similar on average to reference points of the same class than to reference points of the opposite class by a margin $\gamma$. In practice, SLLC is competitive with traditional metric learning methods, with the additional advantage of inducing extremely sparse classifiers. A drawback of the approach is that linear classifiers (unlike $k$-NN) cannot naturally deal with the multi-class setting, and thus one-vs-all or one-vs-one strategies must be used.

**RSL (Cheng)** As OASIS and SLLC, Cheng (2013) also proposes to learn a bilinear similarity, but focuses on the setting of pair matching (predicting whether two pairs are similar). Pairs are of the form $(\boldsymbol{x}, \boldsymbol{x'})$, where $\boldsymbol{x} \in \mathbb{R}^d$ and $\boldsymbol{x'} \in \mathbb{R}^{d'}$ potentially have different dimensionality, thus one has to learn a rectangular matrix $\boldsymbol{M} \in \mathbb{R}^{d \times d'}$. This is a relevant setting for matching instances from different domains, such as images with different resolutions, or queries and documents. The matrix $\boldsymbol{M}$ is set to have fixed rank $r \ll \min(d, d')$. RSL (Riemannian Similarity Learning) is formulated as follows:

$$\begin{aligned} \max_{\boldsymbol{M} \in \mathbb{R}^{d \times d'}} \quad & \sum_{(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{S} \cup \mathcal{D}} \ell(\boldsymbol{x}_i, \boldsymbol{x}_j, y_{ij}) \quad + \quad \|\boldsymbol{M}\|_{\mathcal{F}} \\ \text{s.t.} \quad & \operatorname{rank}(\boldsymbol{M}) = r, \end{aligned} \tag{14}$$

where $\ell$ is some differentiable loss function (such as the log loss or the squared hinge loss). The optimization is carried out efficiently using recent advances in optimization over Riemannian manifolds (Absil et al., 2008) and based on the low-rank factorization of $\boldsymbol{M}$. At each iteration, the procedure finds a descent direction in the tangent space of the current solution, and a retractation step to project the obtained matrix back to the low-rank manifold. It outputs a local minimum of (14). Experiments are conducted on pair-matching problems where RSL achieves state-of-the-art results using a small rank matrix.

### 4.2 Nonlinear Methods

As we have seen, work in supervised metric learning has focused on linear metrics because they are more convenient to optimize (in particular, it is easier to derive convex formulations with the guarantee of finding the global optimum) and less prone to overfitting. In some cases, however, there is nonlinear structure in the data that linear metrics are unable to capture. The kernelization of linear methods can be seen as a satisfactory solution to this problem. This strategy is explained in Section 4.2.1. The few approaches consisting in directly learning nonlinear forms of metrics are addressed in Section 4.2.2.

#### 4.2.1 Kernelization of Linear Methods

The idea of kernelization is to learn a linear metric in the nonlinear feature space induced by a kernel function and thereby combine the best of both worlds, in the spirit of what is done in SVM. Some metric learning approaches have been shown to be kernelizable (see for instance Schultz and Joachims, 2003; Shalev-Shwartz et al., 2004; Hoi et al., 2006; Torresani and Lee, 2006; Davis et al., 2007) using specific arguments, but in general kernelizing a particular metric algorithm is not trivial: a new formulation of the problem has to be derived, where

interface to the data is limited to inner products, and sometimes a different implementation is necessary. Moreover, when kernelization is possible, one must learn a $n \times n$ matrix. As the number of training examples $n$ gets large, the problem becomes intractable.

Recently though, several authors (Chatpatanasiri et al., 2010; Zhang et al., 2010) have proposed general kernelization methods based on Kernel Principal Component Analysis (Schölkopf et al., 1998), a nonlinear extension of PCA (Pearson, 1901). In short, KPCA implicitly projects the data into the nonlinear (potentially infinite-dimensional) feature space induced by a kernel and performs dimensionality reduction in that space. The (unchanged) metric learning algorithm can then be used to learn a metric in that nonlinear space—this is referred to as the "KPCA trick". Chatpatanasiri et al. (2010) showed that the KPCA trick is theoretically sound for unconstrained metric learning algorithms (they prove representer theorems). Another trick (similar in spirit in the sense that it involves some nonlinear preprocessing of the feature space) is based on kernel density estimation and allows one to deal with both numerical and categorical attributes (He et al., 2013). General kernelization results can also be obtained from the equivalence between Mahalanobis distance learning in kernel space and linear transformation kernel learning (Jain et al., 2010, 2012), but are restricted to spectral regularizers. Lastly, Wang et al. (2011) address the problem of choosing an appropriate kernel function by proposing a multiple kernel framework for metric learning.

Note that kernelizing a metric learning algorithm may drastically improve the quality of the learned metric on highly nonlinear problems, but may also favor overfitting (because pair or triplet-based constraints become much easier to satisfy in a nonlinear, high-dimensional kernel space) and thereby lead to poor generalization performance.

#### 4.2.2 Learning Nonlinear Forms of Metrics

A few approaches have tackled the direct optimization of nonlinear forms of metrics. These approaches are subject to local optima and more inclined to overfit the data, but have the potential to significantly outperform linear methods on some problems.

**LSMD (Chopra et al.)** Chopra et al. (2005) pioneered the nonlinear metric learning literature. They learn a nonlinear projection $G_W(\boldsymbol{x})$ parameterized by a vector $W$ such that the $L_1$ distance in the low-dimensional target space $\|G_W(\boldsymbol{x}) - G_W(\boldsymbol{x'})\|_1$ is small for positive pairs and large for negative pairs. No assumption is made about the nature of $G_W$: the parameter $W$ corresponds to the weights in a convolutional neural network and can thus be an arbitrarily complex nonlinear mapping. These weights are learned through back-propagation and stochastic gradient descent so as to minimize a loss function designed to make the distance for positive pairs smaller than the distance of negative pairs by a given margin. Due to the use of neural networks, the approach suffers from local optimality and needs careful tuning of the many hyperparameters, requiring a significant amount of validation data in order to avoid overfitting. This leads to a high computational complexity. Nevertheless, the authors demonstrate the usefulness of LSMD on face verification tasks.

**NNCA (Salakhutdinov & Hinton)** Nonlinear NCA (Salakhutdinov and Hinton, 2007) is another distance learning approach based on deep learning. NNCA first learns a nonlinear, low-dimensional representation of the data using a deep belief network (stacked Restricted Boltzmann Machines) that is pretrained layer-by-layer in an unsupervised way. In a second

step, the parameters of the last layer are fine-tuned by optimizing the NCA objective (Section 3.2). Additional unlabeled data can be used as a regularizer by minimizing their reconstruction error. Although it suffers from the same limitations as LSMD due to its deep structure, NNCA is shown to perform well when enough data is available. For instance, on a digit recognition dataset, NNCA based on a 30-dimensional nonlinear representation significantly outperforms $k$-NN in the original pixel space as well as NCA based on a linear space of same dimension.

**SVML (Xu et al.)** Xu et al. (2012) observe that learning a Mahalanobis distance with an existing algorithm and plugging it into a RBF kernel does not significantly improve SVM classification performance. They instead propose Support Vector Metric Learning (SVML), an algorithm that alternates between (i) learning the SVM model with respect to the current Mahalanobis distance and (ii) learning a Mahalanobis distance that minimizes a surrogate of the validation error of the current SVM model. Since the latter step is nonconvex in any event (due to the nonconvex loss function), the authors optimize the distance based on the decomposition $\boldsymbol{L}^T\boldsymbol{L}$, thus there is no PSD constraint and the approach can be made low-rank. Frobenius regularization on $\boldsymbol{L}$ may be used to avoid overfitting. The optimization procedure is done using a gradient descent approach and is rather efficient although subject to local minima. Nevertheless, SVML significantly improves standard SVM results.

**GB-LMNN (Kedem et al.)** Kedem et al. (2012) propose Gradient-Boosted LMNN, a nonlinear method consisting in generalizing the Euclidean distance with a nonlinear transformation $\phi$ as follows:

$$d_\phi(\boldsymbol{x}, \boldsymbol{x'}) = \|\phi(\boldsymbol{x}) - \phi(\boldsymbol{x'})\|_2.$$

This nonlinear mapping takes the form of an additive function $\phi = \phi_0 + \alpha\sum_{t=1}^{T} h_t$, where $h_1, \ldots, h_T$ are gradient boosted regression trees (Friedman, 2001) of limited depth $p$ and $\phi_0$ corresponds to the mapping learned by linear LMNN. They once again use the same objective function as LMNN and are able to do the optimization efficiently, building on gradient boosting. On an intuitive level, the tree selected by gradient descent at each iteration divides the space into $2^p$ regions, and instances falling in the same region are translated by the same vector—thus examples in different regions are translated in different directions. Dimensionality reduction can be achieved by learning trees with $r$-dimensional output. In practice, GB-LMNN seems quite robust to overfitting and performs well, often achieving comparable or better performance than LMNN and ITML.

**HDML (Norouzi et al.)** Hamming Distance Metric Learning (Norouzi et al., 2012a) proposes to learn mappings from real-valued vectors to binary codes on which the Hamming distance performs well.[32] Recall that the Hamming distance $d_H$ between two binary codes of same length is simply the number of bits on which they disagree. A great advantage of working with binary codes is their small storage cost and the fact that exact neighbor search can be done in sublinear time (Norouzi et al., 2012b). The goal here is to optimize a mapping $b(\boldsymbol{x})$ that projects a $d$-dimensional real-valued input $\boldsymbol{x}$ to a $q$-dimensional binary code. The mapping takes the general form:

$$b(\boldsymbol{x}; \boldsymbol{w}) = \text{sign}\left(f(\boldsymbol{x}; \boldsymbol{w})\right),$$

32. Source code available at: https://github.com/norouzi/hdml

where $f : \mathbb{R}^d \to \mathbb{R}^q$ can be any function differentiable in $\boldsymbol{w}$, $\text{sign}(\cdot)$ is the element-wise sign function and $\boldsymbol{w}$ is a real-valued vector representing the parameters to be learned. For instance, $f$ can be a nonlinear transform obtained with a multilayer neural network. Given a relative constraint $(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_k) \in \mathcal{R}$, denote by $\boldsymbol{h}_i$, $\boldsymbol{h}_j$ and $\boldsymbol{h}_k$ their corresponding binary codes given by $b$. The loss is then given by

$$\ell(\boldsymbol{h}_i, \boldsymbol{h}_j, \boldsymbol{h}_k) = \left[1 - d_H\left(\boldsymbol{h}_i, \boldsymbol{h}_k\right) + d_H\left(\boldsymbol{h}_i, \boldsymbol{h}_j\right)\right]_+ .$$

In the other words, the loss is zero when the Hamming distance between $\boldsymbol{h}_i$ and $\boldsymbol{h}_j$ is a at least one bit smaller than the distance between $\boldsymbol{h}_i$ and $\boldsymbol{h}_k$. HDML is formalized as a loss minimization problem with $L_2$ norm regularization on $\boldsymbol{w}$. This objective function is non-convex and discontinuous, but the authors propose to optimize a continuous upper bound on the loss which can be computed in $O(q^2)$ time, which is efficient as long as the code length $q$ remains small. In practice, the objective is optimized using a stochastic gradient descent approach. Experiments show that relatively short codes obtained by nonlinear mapping are sufficient to achieve few constraint violations, and that a $k$-NN classifier based on these codes can achieve competitive performance with state-of-the-art classifiers. Neyshabur et al. (2013) later showed that using asymmetric codes can lead to shorter encodings while maintaining similar performance.

### 4.3 Local Metric Learning

The methods studied so far learn a global (linear or nonlinear) metric. However, if the data is heterogeneous, a single metric may not well capture the complexity of the task and it might be beneficial to use multiple local metrics that vary across the space (e.g., one for each class or for each instance).[33] This can often be seen as approximating the geodesic distance defined by a metric tensor (see Ramanan and Baker, 2011, for a review on this matter). It is typically crucial that the local metrics be learned simultaneously in order to make them meaningfully comparable and also to alleviate overfitting. Local metric learning has been shown to significantly outperform global methods on some problems, but typically comes at the expense of higher time and memory requirements. Furthermore, they usually do not give rise to a consistent global metric, although some recent work partially addresses this issue (Zhan et al., 2009; Hauberg et al., 2012).

**M$^2$-LMNN (Weinberger & Saul)** Multiple Metrics LMNN[34] (Weinberger and Saul, 2008, 2009) learns several Mahalanobis distances in different parts of the space. As a preprocessing step, training data is partitioned in $C$ clusters. These can be obtained either in a supervised way (using class labels) or without supervision (e.g., using $K$-Means). Then, $C$ metrics (one for each cluster) are learned in a coupled fashion in the form of a generalization of the LMNN's objective, where the distance to a target neighbor or an impostor $\boldsymbol{x}$ is measured under the local metric associated with the cluster to which $\boldsymbol{x}$ belongs. In practice, M$^2$-LMNN can yield significant improvements over standard LMNN (especially with supervised clustering), but this comes at the expense of a higher computational cost,

33. The work of Frome et al. (2007) is one of the first to propose to learn multiple local metrics. However, their approach is specific to computer vision so we chose not to review it here.
34. Source code available at: http://www.cse.wustl.edu/~kilian/code/code.html

and important overfitting (since each local metric can be overly specific to its region) unless a large validation set is used (Wang et al., 2012c).

**GLML (Noh et al.)** The work of Noh et al. (2010), Generative Local Metric Learning, aims at leveraging the power of generative models (known to outperform purely discriminative models when the training set is small) in the context of metric learning. They focus on nearest neighbor classification and express the expected error of a 1-NN classifier as the sum of two terms: the asymptotic probability of misclassification and a metric-dependent term representing the bias due to finite sampling. They show that this bias can be minimized locally by learning a Mahalanobis distance $d_{\boldsymbol{M_i}}$ at each training point $\boldsymbol{x}_i$. This is done by solving, for each training instance, an independent semidefinite program that has an analytical solution. Each matrix $\boldsymbol{M}_i$ is further regularized towards a diagonal matrix in order to alleviate overfitting. Since each local metric is computed independently, GLML can be very scalable. Its performance is competitive on some datasets (where the assumption of Gaussian distribution to model the distribution of data is reasonable) but can perform very poorly on more complex problems (Wang et al., 2012c). Note that GLML does not straightforwardly extend to the $k$-NN setting for $k > 1$. Shi et al. (2011) use GLML metrics as base kernels to learn a global kernel in a discriminative manner.

**Bk-means (Wu et al.)** Wu et al. (2009, 2012) propose to learn Bregman distances (or Bregman divergences), a family of metrics that do not necessarily satisfy the triangle inequality or symmetry (Bregman, 1967). Given the strictly convex and twice differentiable function $\varphi : \mathbb{R}^d \to \mathbb{R}$, the Bregman distance is defined as:

$$d_\varphi(\boldsymbol{x}, \boldsymbol{x'}) = \varphi(\boldsymbol{x}) - \varphi(\boldsymbol{x'}) - (\boldsymbol{x} - \boldsymbol{x'})^T \nabla\varphi(\boldsymbol{x'}).$$

It generalizes many widely-used measures: the Mahalanobis distance is recovered by setting $\varphi(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{M}\boldsymbol{x}$, the KL divergence (Kullback and Leibler, 1951) by choosing $\varphi(\boldsymbol{p}) = \sum_{i=1}^d p_i \log p_i$ (here, $\boldsymbol{p}$ is a discrete probability distribution), etc. Wu et al. consider the following symmetrized version:

$$\begin{aligned} d_\varphi(\boldsymbol{x}, \boldsymbol{x'}) &= \left(\nabla\varphi(\boldsymbol{x}) - \nabla\varphi(\boldsymbol{x'})\right)^T (\boldsymbol{x} - \boldsymbol{x'}) \\ &= (\boldsymbol{x} - \boldsymbol{x'})^T \nabla^2\varphi(\tilde{\boldsymbol{x}})(\boldsymbol{x} - \boldsymbol{x'}), \end{aligned}$$

where $\tilde{\boldsymbol{x}}$ is a point on the line segment between $\boldsymbol{x}$ and $\boldsymbol{x'}$. Therefore, $d_\varphi$ amounts to a Mahalanobis distance parameterized by the Hessian matrix of $\varphi$ which depends on the location of $\boldsymbol{x}$ and $\boldsymbol{x'}$. In this respect, learning $\varphi$ can be seen as learning an infinite number of local Mahalanobis distances. They take a nonparametric approach by assuming $\phi$ to belong to a Reproducing Kernel Hilbert Space $\mathcal{H}_K$ associated to a kernel function $K(\boldsymbol{x}, \boldsymbol{x'}) = h(\boldsymbol{x}^T\boldsymbol{x'})$ where $h(z)$ is a strictly convex function (set to $\exp(z)$ in the experiments). This allows the derivation of a representer theorem. Setting $\varphi(\boldsymbol{x}) = \sum_{i=1}^n \alpha_i h(\boldsymbol{x}_i^T\boldsymbol{x})$ leads to the following formulation based on classic positive/negative pairs:

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^n_+, b} \quad \frac{1}{2}\boldsymbol{\alpha}^T\boldsymbol{K}\boldsymbol{\alpha} \quad + \quad C \sum_{(\boldsymbol{x}_i, \boldsymbol{x}_j)\in\mathcal{S}\cup\mathcal{D}} \ell\left(y_{ij}\left[d_\varphi(\boldsymbol{x}_i, \boldsymbol{x}_j) - b\right]\right), \tag{15}$$

where $\boldsymbol{K}$ is the Gram matrix, $\ell(t) = \max(0, 1 - t)$ is the hinge loss and $C$ is the trade-off parameter. Problem (15) is solved by a simple subgradient descent approach where each

iteration has a linear complexity. Note that (15) only has $n+1$ variables instead of $d^2$ in most metric learning formulations, leading to very scalable learning. The downside is that computing the learned distance requires $n$ kernel evaluations, which can be expensive for large datasets. The method is evaluated on clustering problems and exhibits good performance, matching or improving that of other metric learning approaches.

**PLML (Wang et al.)** Wang et al. (2012c) propose PLML,[35] a Parametric Local Metric Learning method where a Mahalanobis metric $d^2_{\boldsymbol{M_i}}$ is learned for each training instance $\boldsymbol{x}_i$:

$$d^2_{\boldsymbol{M}_i}(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i - \boldsymbol{x}_j)^T \boldsymbol{M}_i (\boldsymbol{x}_i - \boldsymbol{x}_j).$$

$\boldsymbol{M}_i$ is parameterized to be a weighted linear combination of metric bases $\boldsymbol{M}_{b_1}, \ldots, \boldsymbol{M}_{b_2}$, where $\boldsymbol{M}_{b_j} \succeq 0$ is associated with an anchor point $\boldsymbol{u}_j$.[36] In other words, $\boldsymbol{M}_i$ is defined as:

$$\boldsymbol{M}_i = \sum_{j=1}^{m} W_{ib_j} \boldsymbol{M}_{b_j}, \quad W_{i,b_j} \geq 0, \quad \sum_{j=1}^{m} W_{ib_j} = 1,$$

where the nonnegativity of the weights ensures that the combination is PSD. The weight learning procedure is a trade-off between three terms: (i) each point $\boldsymbol{x}$ should be close to its linear approximation $\sum_{j=1}^{m} W_{ib_j} \boldsymbol{u}_j$, (ii) the weighting scheme should be local (i.e., $W_{ib_j}$ should be large if $\boldsymbol{x}_i$ and $\boldsymbol{u}_i$ are similar), and (iii) the weights should vary smoothly over the data manifold (i.e., similar training instances should be assigned similar weights).[37] Given the weights, the basis metrics $\boldsymbol{M}_{b_1}, \ldots, \boldsymbol{M}_{b_m}$ are then learned in a large-margin fashion using positive and negative training pairs and Frobenius regularization. In terms of scalability, the weight learning procedure is fairly efficient. However, the metric bases learning procedure requires at each step an eigen-decomposition that scales in $O(d^3)$, making the approach intractable for high-dimensional problems. In practice, PLML performs very well on the evaluated datasets, and is quite robust to overfitting due to its global manifold regularization. However, like LMNN, PLML is sensitive to the relevance of the Euclidean distance to assess the similarity between (anchor) points. Note that PLML has many hyper-parameters but in the experiments the authors use default values for most of them. Huang et al. (2013) propose to regularize the anchor metrics to be low-rank and use alternating optimization to solve the problem.

**RFD (Xiong et al.)** The originality of the Random Forest Distance (Xiong et al., 2012) is to see the metric learning problem as a pair classification problem.[38] Each pair of examples $(\boldsymbol{x}, \boldsymbol{x'})$ is mapped to the following feature space:

$$\phi(\boldsymbol{x}, \boldsymbol{x'}) = \begin{bmatrix} |\boldsymbol{x} - \boldsymbol{x'}| \\ \frac{1}{2}(\boldsymbol{x} + \boldsymbol{x'}) \end{bmatrix} \in \mathbb{R}^{2d}.$$

35. Source code available at: http://cui.unige.ch/~wangjun/papers/PLML.zip
36. In practice, these anchor points are defined as the means of clusters constructed by the $K$-Means algorithm.
37. The weights of a test instance can be learned by optimizing the same trade-off given the weights of the training instances, and simply set to the weights of the nearest training instance.
38. Source code available at: http://www.cse.buffalo.edu/~cxiong/RFD_Package.zip

The first part of $\phi(\boldsymbol{x}, \boldsymbol{x'})$ encodes the relative position of the examples and the second part their absolute position, as opposed to the implicit mapping of the Mahalanobis distance which only encodes relative information. The metric is based on a random forest $F$, i.e.,

$$d_{RFD}(\boldsymbol{x}, \boldsymbol{x'}) = F(\phi(\boldsymbol{x}, \boldsymbol{x'})) = \frac{1}{T}\sum_{t=1}^{T} f_t(\phi(\boldsymbol{x}, \boldsymbol{x'})),$$

where $f_t(\cdot) \in \{0, 1\}$ is the output of decision tree $t$. RFD is thus highly nonlinear and is able to implicitly adapt the metric throughout the space: when a decision tree in $F$ selects a node split based on a value of the absolute position part, then the entire sub-tree is specific to that region of $\mathbb{R}^{2d}$. As compared to other local metric learning methods, training is very efficient: each tree takes $O(n \log n)$ time to generate and trees can be built in parallel. A drawback is that the evaluation of the learned metric requires to compute the output of the $T$ trees. The experiments highlight the importance of encoding absolute information, and show that RFD outperforms some global and local metric learning methods on several datasets and appears to be quite fast.

### 4.4 Metric Learning for Histogram Data

Histograms are feature vectors that lie on the probability simplex $\mathcal{S}^d$. This representation is very common in areas dealing with complex objects, such as natural language processing, computer vision or bioinformatics: each instance is represented as a bag of features, i.e., a vector containing the frequency of each feature in the object. Bags-of(-visual)-words (Salton et al., 1975; Li and Perona, 2005) are a common example of such data. We present here three metric learning methods designed specifically for histograms.

**$\chi^2$-LMNN (Kedem et al.)** Kedem et al. (2012) propose $\chi^2$-LMNN, which is based on a simple yet prominent histogram metric, the $\chi^2$ distance (Hafner et al., 1995), defined as

$$\chi^2(\boldsymbol{x}, \boldsymbol{x'}) = \frac{1}{2}\sum_{i=1}^{d} \frac{(x^i - x'^i)^2}{x^i + x'^i}, \tag{16}$$

where $x^i$ denotes the $i^{th}$ feature of $\boldsymbol{x}$.[39] Note that $\chi^2$ is a (nonlinear) proper distance. They propose to generalize this distance with a linear transformation, introducing the following pseudo-distance:

$$\chi^2_{\boldsymbol{L}}(\boldsymbol{x}, \boldsymbol{x'}) = \chi^2(\boldsymbol{L}\boldsymbol{x}, \boldsymbol{L}\boldsymbol{x'}),$$

where $\boldsymbol{L} \in \mathbb{R}^{r \times d}$, with the constraint that $\boldsymbol{L}$ maps any $\boldsymbol{x}$ onto $\mathcal{S}^d$ (the authors show that this can be enforced using a simple trick). The objective function is the same as LMNN[40] and is optimized using a standard subgradient descent procedure. Although subject to local optima, experiments show great improvements on histogram data compared to standard histogram metrics and Mahalanobis distance learning methods, and promising results for dimensionality reduction (when $r < d$).

39. The sum in (16) must be restricted to entries that are nonzero in either $\boldsymbol{x}$ or $\boldsymbol{x'}$ to avoid division by zero.
40. To be precise, it requires an additional parameter. In standard LMNN, due to the linearity of the Mahalanobis distance, solutions obtained with different values of the margin only differ up to a scaling factor—the margin is thus set to 1. Here, $\chi^2$ is nonlinear and therefore this value must be tuned.

**GML (Cuturi & Avis)** While $\chi^2$-LMNN optimizes a simple bin-to-bin histogram distance, Cuturi and Avis (2011) propose to consider the more powerful cross-bin Earth Mover's Distance (EMD) introduced by Rubner et al. (2000), which can be seen as the distance between a source histogram $\boldsymbol{x}$ and a destination histogram $\boldsymbol{x'}$. On an intuitive level, $\boldsymbol{x}$ is viewed as piles of earth at several locations (bins) and $\boldsymbol{x'}$ as several holes, where the value of each feature represents the amount of earth and the capacity of the hole respectively. The EMD is then equal to the minimum amount of effort needed to move all the earth from $\boldsymbol{x}$ to $\boldsymbol{x'}$. The costs of moving one unit of earth from bin $i$ of $\boldsymbol{x}$ to bin $j$ of $\boldsymbol{x'}$ is encoded in the so-called ground distance matrix $\boldsymbol{D} \in \mathbb{R}^{d\times d}$.[41] The computation of EMD amounts to finding the optimal flow matrix $\boldsymbol{F}$, where $f_{ij}$ corresponds to the amount of earth moved from bin $i$ of $\boldsymbol{x}$ to bin $j$ of $\boldsymbol{x'}$. Given the ground distance matrix $\boldsymbol{D}$, $\text{EMD}_{\boldsymbol{D}}(\boldsymbol{x}, \boldsymbol{x'})$ is linear and can be formulated as a linear program:

$$\text{EMD}_{\boldsymbol{D}}(\boldsymbol{x}, \boldsymbol{x'}) = \min_{\boldsymbol{f}\in\mathbb{C}(\boldsymbol{x},\boldsymbol{x'})} \boldsymbol{d}^T\boldsymbol{f},$$

where $\boldsymbol{f}$ and $\boldsymbol{d}$ are respectively the flow and the ground matrices rewritten as vectors for notational simplicity, and $\mathbb{C}(\boldsymbol{x}, \boldsymbol{x'})$ is the convex set of feasible flows (which can be represented as linear constraints). Ground Metric Learning (GML) aims at learning $\boldsymbol{D}$ based on training triplets $(\boldsymbol{x}_i, \boldsymbol{x}_j, w_{ij})$ where $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are two histograms and $w_{ij} \in \mathbb{R}$ is a weight quantifying the similarity between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. The optimized criterion essentially aims at minimizing the sum of $w_{ij}\text{EMD}_{\boldsymbol{D}}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ — which is a nonlinear function in $\boldsymbol{D}$ — by casting the problem as a difference of two convex functions. A local minima is found efficiently by a subgradient descent approach. Experiments on image datasets show that GML outperforms standard histogram distances as well as Mahalanobis distance methods.

**EMDL (Wang & Guibas)** Building on GML and successful Mahalanobis distance learning approaches such as LMNN, Wang and Guibas (2012) aim at learning the EMD ground matrix in the more flexible setting where the algorithm is provided with a set of relative constraints $\mathcal{R}$ that must be satisfied with a large margin. The problem is formulated as

$$\begin{aligned} \min_{\boldsymbol{D}\in\mathbb{D}} \quad & \|\boldsymbol{D}\|_{\mathcal{F}}^2 \quad + \quad C\sum_{i,j,k}\xi_{ijk} \\ \text{s.t.} \quad & \text{EMD}_{\boldsymbol{D}}(\boldsymbol{x}_i, \boldsymbol{x}_k) - \text{EMD}_{\boldsymbol{D}}(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 1 - \xi_{ijk} \quad \forall(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_k) \in \mathcal{R}, \end{aligned} \tag{17}$$

where $\mathbb{D} = \left\{\boldsymbol{D} \in \mathbb{R}^{d\times d} : \forall i, j \in \{1, \ldots, d\}, d_{ij} \geq 0, d_{ii} = 0\right\}$ and $C \geq 0$ is the trade-off parameter.[42] The authors also propose a pair-based formulation. Problem (17) is bi-convex and is solved using an alternating procedure: first fix the ground metric and solve for the flow matrices (this amounts to a set of standard EMD problems), then solve for the ground matrix given the flows (this is a quadratic program). The algorithm stops when the changes in the ground matrix are sufficiently small. The procedure is subject to local optima (because (17) is not jointly convex) and is not guaranteed to converge: there is a need for a trade-off parameter $\alpha$ between stable but conservative updates (i.e., staying close

41. For EMD to be proper distance, $\boldsymbol{D}$ must satisfy the following $\forall i, j, k \in \{1, \ldots, d\}$: (i) $d_{ij} \geq 0$, (ii) $d_{ii} = 0$, (iii) $d_{ij} = d_{ji}$ and (iv) $d_{ij} \leq d_{ik} + d_{kj}$.
42. Note that unlike in GML, $\boldsymbol{D} \in \mathbb{D}$ may not be a valid distance matrix. In this case, $\text{EMD}_{\boldsymbol{D}}$ is not a proper distance.

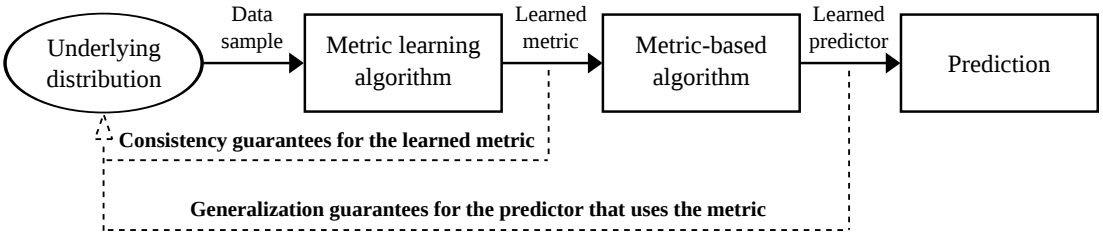


Figure 5: The two-fold problem of generalization in metric learning. We may be interested in the generalization ability of the learned metric itself: can we say anything about its consistency on unseen data drawn from the same distribution? Furthermore, we may also be interested in the generalization ability of the predictor using that metric: can we relate its performance on unseen data to the quality of the learned metric?

to the previous ground matrix) and aggressive but less stable updates. Experiments on face verification datasets confirm that EMDL improves upon standard histogram distances and Mahalanobis distance learning methods.

### 4.5 Generalization Guarantees for Metric Learning

The derivation of guarantees on the generalization performance of the learned model is a wide topic in statistical learning theory (Vapnik and Chervonenkis, 1971; Valiant, 1984). Assuming that data points are drawn i.i.d. from some (unknown but fixed) distribution $P$, one essentially aims at bounding the deviation of the *true risk* of the learned model (its performance on unseen data) from its *empirical risk* (its performance on the training sample).[43]

In the specific context of metric learning, we claim that the question of generalization can be seen as two-fold (Bellet, 2012), as illustrated by Figure 5:

- First, one may consider the *consistency of the learned metric*, i.e., trying to bound the deviation between the empirical performance of the metric on the training sample and its generalization performance on unseen data.
- Second, the learned metric is used to improve the performance of some prediction model (e.g., $k$-NN or a linear classifier). It would thus be meaningful to express the *generalization performance of this predictor* in terms of that of the learned metric.

As in the classic supervised learning setting (where training data consist of individual labeled instances), generalization guarantees may be derived for supervised metric learning (where training data consist of pairs or triplets). Indeed, most of supervised metric learning methods can be seen as minimizing a (regularized) loss function $\ell$ based on the training pairs/triplets. However, the i.i.d. assumption is violated in the metric learning scenario since the training pairs/triplets are constructed from the training sample. For this reason,

43. This deviation is typically a function of the number of training examples and some notion of complexity of the model.

establishing generalization guarantees for the learned metric is challenging and only recently has this question been investigated from a theoretical standpoint.

**Metric consistency bounds for batch methods** Given a training sample $\mathcal{T} = \{\boldsymbol{z}_i = (\boldsymbol{x}_i, y_i)\}_{i=1}^n$ drawn i.i.d. from an unknown distribution $\mu$, let us consider fully supervised Mahalanobis metric learning of the following general form:

$$\min_{\boldsymbol{M} \in \mathbb{S}_+^d} \quad \frac{1}{n^2} \sum_{\boldsymbol{z}_i, \boldsymbol{z}_j \in \mathcal{T}} \ell(d^2_{\boldsymbol{M}}, \boldsymbol{z}_i, \boldsymbol{z}_j) \quad + \quad \lambda R(\boldsymbol{M}),$$

where $R(\boldsymbol{M})$ is the regularizer, $\lambda$ the regularization parameter and the loss function $\ell$ is of the form $\ell(d^2_{\boldsymbol{M}}, \boldsymbol{z}_i, \boldsymbol{z}_j) = g(y_i y_j [c - d^2_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_j)])$ with $c > 0$ a decision threshold variable and $g$ convex and Lipschitz continuous. This includes popular loss functions such as the hinge loss. Several recent work have proposed to study the convergence of the empirical risk (as measured by $\ell$ on pairs from $\mathcal{T}$) to the true risk over the unknown probability distribution $\mu$. The framework proposed by Bian & Tao (2011; 2012) is quite rigid since it relies on strong assumptions on the distribution of the examples and cannot accommodate any regularization (a constraint to bound $\boldsymbol{M}$ is used instead). Jin et al. (2009) use a notion of uniform stability (Bousquet and Elisseeff, 2002) adapted to the case of metric learning (where training data is made of pairs to derive generalization bounds that are limited to Frobenius norm regularization. Bellet and Habrard (2012) demonstrate how to adapt the more flexible notion of algorithmic robustness (Xu and Mannor, 2012) to the metric learning setting to derive (loose) generalization bounds for any matrix norm (including sparsity-inducing ones) as regularizer. They also show that a weak notion of robustness is necessary and sufficient for metric learning algorithms to generalize well. Lastly, Cao et al. (2012a) use a notion of Rademacher complexity (Bartlett and Mendelson, 2002) dependent on the regularizer to derive bounds for several matrix norms. All these results can easily adapted to non-Mahalanobis linear metric learning formulations.

**Regret bound conversion for online methods** Wang et al. (2012d, 2013b) deal with the online learning setting. They show that existing proof techniques to convert regret bounds into generalization bounds (see for instance Cesa-Bianchi and Gentile, 2008) only hold for univariate loss functions, but derive an alternative framework that can deal with pairwise losses. At each round, the online algorithm receives a new instance and is assumed to pair it with all previously-seen data points. As this is expensive or even infeasible in practice, Kar et al. (2013) propose to use a buffer containing only a bounded number of the most recent instances. They are also able to obtain tighter bounds based on a notion of Rademacher complexity, essentially adapting and extending the work of Cao et al. (2012a). These results suggest that one can obtain generalization bounds for most/all online metric learning algorithms with bounded regret (such as those presented in Section 3.4).

**Link between learned metric and classification performance** The second question of generalization (i.e., at the classifier level) remains an open problem for the most part. To the best of our knowledge, it has only been addressed in the context of metric learning for linear classification. Bellet et al. (2011, 2012a,b) rely upon the theory of learning with $(\epsilon, \gamma, \tau)$-good similarity function (Balcan et al., 2008a), which makes the link between properties of a similarity function and the generalization of a linear classifier built from this

similarity. Bellet et al. propose to use $(\epsilon, \gamma, \tau)$-goodness as an objective function for metric learning, and show that in this case it is possible to derive generalization guarantees not only for the learned similarity but also for the linear classifier. Guo and Ying (2014) extend the results of Bellet et al. to several matrix norms using a Rademacher complexity analysis, based on techniques from Cao et al. (2012a).

### 4.6 Semi-Supervised Metric Learning Methods

In this section, we present two categories of metric learning methods that are designed to deal with semi-supervised learning tasks. The first one corresponds to the standard semi-supervised setting, where the learner makes use of unlabeled pairs in addition to positive and negative constraints. The second one concerns approaches which learn metrics to address semi-supervised domain adaptation problems where the learner has access to labeled data drawn according to a source distribution and unlabeled data generated from a different (but related) target distribution.

#### 4.6.1 Standard Semi-Supervised Setting

The following metric learning methods leverage the information brought by the set of *unlabeled pairs*, i.e., pairs of training examples that do not belong to the sets of positive and negative pairs:

$$\mathcal{U} = \{(\boldsymbol{x}_i, \boldsymbol{x}_j) : i \neq j, (\boldsymbol{x}_i, \boldsymbol{x}_j) \notin \mathcal{S} \cup \mathcal{D}\}.$$

An early approach by Bilenko et al. (2004) combined semi-supervised clustering with metric learning. In the following, we review general metric learning formulations that incorporate information from the set of unlabeled pairs $\mathcal{U}$.

**LRML (Hoi et al.)** Hoi et al. (2008, 2010) propose to follow the principles of manifold regularization for semi-supervised learning (Belkin and Niyogi, 2004) by resorting to a weight matrix $\boldsymbol{W}$ that encodes the similarity between pairs of points.[44] Hoi et al. construct $\boldsymbol{W}$ using the Euclidean distance as follows:

$$W_{ij} = \begin{cases} 1 & \text{if } \boldsymbol{x}_i \in \mathcal{N}(\boldsymbol{x}_j) \text{ or } \boldsymbol{x}_j \in \mathcal{N}(\boldsymbol{x}_i) \\ 0 & \text{otherwise} \end{cases}$$

where $\mathcal{N}(\boldsymbol{x}_j)$ denotes the nearest neighbor list of $\boldsymbol{x}_j$. Using $\boldsymbol{W}$, they use the following regularization known as the graph Laplacian regularizer:

$$\frac{1}{2}\sum_{i,j=1}^{n} d^2_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_j) W_{ij} = \text{tr}(\boldsymbol{X}\boldsymbol{L}\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{M}),$$

where $\boldsymbol{X}$ is the data matrix and $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$ is the graph Laplacian matrix with $\boldsymbol{D}$ a diagonal matrix such that $D_{ii} = \sum_j W_{ij}$. Intuitively, this regularization favors an "affinity-preserving" metric: the distance between points that are similar according to $\boldsymbol{W}$ should remain small according to the learned metric. Experiments show that LRML (Laplacian Regularized Metric Learning) significantly outperforms supervised methods when the side

44. Source code available at: http://www.ee.columbia.edu/~wliu/

information is scarce. An obvious drawback is that computing $\boldsymbol{W}$ is intractable for large-scale datasets. This work has inspired a number of extensions and improvements: Liu et al. (2010) introduce a refined way of constructing $\boldsymbol{W}$ while Baghshah and Shouraki (2009), Zhong et al. (2011) and Wang et al. (2013a) use a different (but similar in spirit) manifold regularizer.

**M-DML (Zha et al.)** The idea of Zha et al. (2009) is to augment Laplacian regularization with metrics $\boldsymbol{M}_1, \dots, \boldsymbol{M}_K$ learned from auxiliary datasets. Formally, for each available auxiliary metric, a weight matrix $\boldsymbol{W}_k$ is constructed following Hoi et al. (2008, 2010) but using metric $\boldsymbol{M}_k$ instead of the Euclidean distance. These are then combined to obtain the following regularizer:

$$\sum_{k=1}^{K} \alpha_k \operatorname{tr}(\boldsymbol{X}\boldsymbol{L}_k\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{M}),$$

where $\boldsymbol{L}_k$ is the Laplacian associated with weight matrix $\boldsymbol{W}_k$ and $\alpha_k$ is the weight reflecting the utility of auxiliary metric $\boldsymbol{M}_k$. As such weights are difficult to set in practice, Zha et al. propose to learn them together with the metric $\boldsymbol{M}$ by alternating optimization (which only converges to a local minimum). Experiments on a face recognition task show that metrics learned from auxiliary datasets can be successfully used to improve performance over LRML.

**SERAPH (Niu et al.)** Niu et al. (2012) tackle semi-supervised metric learning from an information-theoretic perspective by optimizing a probability of labeling a given pair parameterized by a Mahalanobis distance:[45]

$$p^{\boldsymbol{M}}(y|\boldsymbol{x}, \boldsymbol{x'}) = \frac{1}{1 + \exp\left(y(d^2_{\boldsymbol{M}}(\boldsymbol{x}, \boldsymbol{x'}) - \eta)\right)}.$$

$\boldsymbol{M}$ is optimized to maximize the entropy of $p^{\boldsymbol{M}}$ on the labeled pairs $\mathcal{S} \cup \mathcal{D}$ and minimize it on unlabeled pairs $\mathcal{U}$, following the entropy regularization principle (Grandvalet and Bengio, 2004). Intuitively, the regularization enforces low uncertainty of unobserved weak labels. They also encourage a low-rank projection by using the trace norm. The resulting nonconvex optimization problem is solved using an EM-like iterative procedure where the M-step involves a projection on the PSD cone. The proposed method outperforms supervised metric learning methods when the amount of supervision is very small, but was only evaluated against one semi-supervised method (Baghshah and Shouraki, 2009) known to be subject to overfitting.

### 4.6.2 Metric Learning for Domain Adaptation

In the domain adaptation (DA) setting (Mansour et al., 2009; Quiñonero-Candela, 2009; Ben-David et al., 2010), the labeled training data and the test data come from different (but somehow related) distributions (referred to as the source and target distributions respectively). This situation occurs very often in real-world applications—famous examples include speech recognition, spam detection and object recognition—and is also relevant for metric learning. Although domain adaptation is sometimes achieved by using a small

45. Source code available at: http://sugiyama-www.cs.titech.ac.jp/~gang/software.html

sample of labeled target data (Saenko et al., 2010; Kulis et al., 2011), we review here the more challenging case where only unlabeled target data is available.

**CDML (Cao et al.)** CDML (Cao et al., 2011), for Consistent Distance Metric Learning, deals with the setting of covariate shift, which assumes that source and target data distributions $p_S(\boldsymbol{x})$ and $p_T(\boldsymbol{x})$ are different but the conditional distribution of the labels given the features, $p(y|\boldsymbol{x})$, remains the same. In the context of metric learning, the assumption is made at the pair level, i.e., $p(y_{ij}|\boldsymbol{x}_i, \boldsymbol{x}_j)$ is stable across domains. Cao et al. show that if some metric learning algorithm minimizing some training loss $\sum_{(\boldsymbol{x}_i,\boldsymbol{x}_j)\in\mathcal{S}\cup\mathcal{D}} \ell(d^2_{\boldsymbol{M}}, \boldsymbol{x}_i, \boldsymbol{x}_j)$ is asymptotically consistent without covariate shift, then the following algorithm is consistent under covariate shift:

$$\min_{\boldsymbol{M}\in\mathbb{S}^d_+} \sum_{(\boldsymbol{x}_i,\boldsymbol{x}_j)\in\mathcal{S}\cup\mathcal{D}} w_{ij}\ell(d^2_{\boldsymbol{M}}, \boldsymbol{x}_i, \boldsymbol{x}_j), \quad \text{where } w_{ij} = \frac{p_T(\boldsymbol{x}_i)p_T(\boldsymbol{x}_j)}{p_S(\boldsymbol{x}_i)p_S(\boldsymbol{x}_j)}. \tag{18}$$

Problem (18) can be seen as cost-sensitive metric learning, where the cost of each pair is given by the importance weight $w_{ij}$. Therefore, adapting a metric learning algorithm to covariate shift boils down to computing the importance weights, which can be done reliably using unlabeled data (Tsuboi et al., 2008). The authors experiment with ITML and show that their adapted version outperforms the regular one in situations of (real or simulated) covariate shift.

**DAML (Geng et al.)** DAML (Geng et al., 2011), for Domain Adaptation Metric Learning, tackles the general domain adaptation setting. In this case, a classic strategy in DA is to use a term that brings the source and target distribution closer. Following this line of work, Geng et al. regularize the metric using the empirical Maximum Mean Discrepancy (MMD, Gretton et al., 2006), a nonparametric way of measuring the difference in distribution between the source sample $S$ and the target sample $T$:

$$MMD(S,T) = \left\| \frac{1}{|S|}\sum_{i=1}^{|S|}\varphi(\boldsymbol{x}_i) - \frac{1}{|T|}\sum_{i=1}^{|T|}\varphi(\boldsymbol{x}'_i) \right\|^2_{\mathcal{H}},$$

where $\varphi(\boldsymbol{x})$ is a nonlinear feature mapping function that maps $\boldsymbol{x}$ to the Reproducing Kernel Hilbert Space $\mathcal{H}$. The MMD can be computed efficiently using the kernel trick and can thus be used as a (convex) regularizer in kernelized metric learning algorithms (see Section 4.2.1). DAML is thus a trade-off between satisfying the constraints on the labeled source data and finding a projection that minimizes the discrepancy between the source and target distribution. Experiments on face recognition and image annotation tasks in the DA setting highlight the effectiveness of DAML compared to classic metric learning methods.

## 5. Metric Learning for Structured Data

In many domains, data naturally come structured, as opposed to the "flat" feature vector representation we have focused on so far. Indeed, instances can come in the form of strings, such as words, text documents or DNA sequences; trees like XML documents, secondary structure of RNA or parse trees; and graphs, such as networks, 3D objects or molecules. In

| Page | Name | Year | Source Code | Data Type | Method | Script | Optimum | Negative Pairs |
|---|---|---|---|---|---|---|---|---|
| 38 | R&Y | 1998 | Yes | String | Generative+EM | All | Local | No |
| 38 | O&S | 2006 | Yes | String | Discriminative+EM | All | Local | No |
| 39 | Saigo | 2006 | Yes | String | Gradient Descent | All | Local | No |
| 39 | GESL | 2011 | Yes | All | Gradient Descent | Levenshtein | Global | Yes |
| 40 | Bernard | 2006 | Yes | Tree | Both+EM | All | Local | No |
| 40 | Boyer | 2007 | Yes | Tree | Generative+EM | All | Local | No |
| 40 | Dalvi | 2009 | No | Tree | Discriminative+EM | All | Local | No |
| 40 | Emms | 2012 | No | Tree | Discriminative+EM | Optimal | Local | No |
| 40 | N&B | 2007 | No | Graph | Generative+EM | All | Local | No |

Table 3: Main features of metric learning methods for structured data. Note that all methods make use of positive pairs.

the context of structured data, metrics are especially appealing because they can be used as a proxy to access data without having to manipulate these complex objects. Indeed, given an appropriate structured metric, one can use any metric-based algorithm as if the data consisted of feature vectors. Many of these metrics actually rely on representing structured objects as feature vectors, such as some string kernels (see Lodhi et al., 2002, and variants) or bags-of-(visual)-words (Salton et al., 1975; Li and Perona, 2005). In this case, metric learning can simply be performed on the feature vector representation, but this strategy can imply a significant loss of structural information. On the other hand, there exist metrics that operate directly on the structured objects and can thus capture more structural distortions. However, learning such metrics is challenging because most of structured metrics are combinatorial by nature, which explains why it has received less attention than metric learning from feature vectors. In this section, we focus on the edit distance, which basically measures (in terms of number of operations) the cost of turning an object into another. Edit distance has attracted most of the interest in the context of metric learning for structured data because (i) it is defined for a variety of objects: sequences (Levenshtein, 1966), trees (Bille, 2005) and graphs (Gao et al., 2010), (ii) it is naturally amenable to learning due to its parameterization by a cost matrix.

We review string edit distance learning in Section 5.1, while methods for trees and graphs are covered in Section 5.2. The features of each approach are summarized in Table 3.

### 5.1 String Edit Distance Learning

In this section, we first introduce some notations as well as the string edit distance. We then review the relevant metric learning methods.

#### 5.1.1 Notations and Definitions

**Definition 1 (Alphabet and string)** *An alphabet $\Sigma$ is a finite nonempty set of symbols. A string $x$ is a finite sequence of symbols from $\Sigma$. The empty string/symbol is denoted by $\$$ and $\Sigma^*$ is the set of all finite strings (including $\$$) that can be generated from $\Sigma$. Finally, the length of a string $x$ is denoted by $|x|$.*

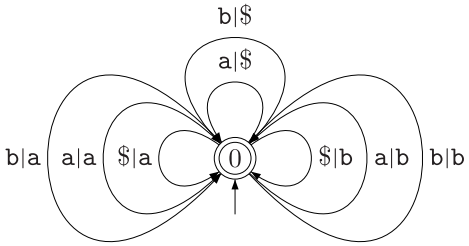


Figure 6: A memoryless stochastic transducer that models the edit probability of any pair of strings built from $\Sigma = \{\mathtt{a}, \mathtt{b}\}$. Edit probabilities assigned to each transition are not shown here for the sake of readability.

**Definition 2 (String edit distance)** *Let $\boldsymbol{C}$ be a nonnegative $(|\Sigma|+1)\times(|\Sigma|+1)$ matrix giving the cost of the following elementary edit operations: insertion, deletion and substitution of a symbol, where symbols are taken from $\Sigma\cup\{\$\}$. Given two strings $\mathtt{x}, \mathtt{x}' \in \Sigma^*$, an edit script is a sequence of operations that turns $\mathtt{x}$ into $\mathtt{x}'$. The string edit distance (Levenshtein, 1966) between $\mathtt{x}$ and $\mathtt{x}'$ is defined as the cost of the cheapest edit script and can be computed in $O(|\mathtt{x}|\cdot|\mathtt{x}'|)$ time by dynamic programming.*

Similar metrics include the Needleman-Wunsch score (Needleman and Wunsch, 1970) and the Smith-Waterman score (Smith and Waterman, 1981). These alignment-based measures use the same substitution operations as the edit distance, but a linear gap penalty function instead of insertion/deletion costs.

The standard edit distance, often called Levenshtein edit distance, is based on a unit cost for all operations. However, this might not reflect the reality of the considered task: for example, in typographical error correction, the probability that a user hits the Q key instead of W on a QWERTY keyboard is much higher than the probability that he hits Q instead of Y. For some applications, such as protein alignment or handwritten digit recognition, hand-tuned cost matrices may be available (Dayhoff et al., 1978; Henikoff and Henikoff, 1992; Micó and Oncina, 1998). Otherwise, there is a need for automatically learning the cost matrix $\boldsymbol{C}$ for the task at hand.

### 5.1.2 Stochastic String Edit Distance Learning

Optimizing the edit distance is challenging because the optimal sequence of operations depends on the edit costs themselves, and therefore updating the costs may change the optimal edit script. Most general-purpose approaches get round this problem by considering a stochastic variant of the edit distance, where the cost matrix defines a probability distribution over the edit operations. One can then define an edit similarity as the posterior probability $p_e(\mathtt{x}'|\mathtt{x})$ that an input string $\mathtt{x}$ is turned into an output string $\mathtt{x}'$. This corresponds to summing over all possible edit scripts that turn $\mathtt{x}$ into $\mathtt{x}'$ instead of only considering the optimal script. Such a stochastic edit process can be represented as a probabilistic model, such as a stochastic transducer (Figure 6), and one can estimate the parameters of the model (i.e., the cost matrix) that maximize the expected log-likelihood of positive pairs. This is done via an EM-like iterative procedure (Dempster et al., 1977).

Note that unlike the standard edit distance, the obtained edit similarity does not usually satisfy the properties of a distance (in fact, it is often not symmetric and rarely satisfies the triangular inequality).

**Ristad and Yianilos** The first method for learning a string edit metric, in the form of a generative model, was proposed by Ristad and Yianilos (1998).[46] They use a memoryless stochastic transducer which models the joint probability of a pair $p_e(\mathsf{x}, \mathsf{x}')$ from which $p_e(\mathsf{x}'|\mathsf{x})$ can be estimated. Parameter estimation is performed with an EM procedure. The Expectation step takes the form of a probabilistic version of the dynamic programing algorithm of the standard edit distance. The M-step aims at maximizing the likelihood of the training pairs of strings so as to define a joint distribution over the edit operations:

$$\sum_{(u,v)\in(\Sigma\cup\{\$\})^2\setminus\{\$,\$\}} \boldsymbol{C}_{uv} + c(\#) = 1, \quad \text{with } c(\#) > 0 \text{ and } \boldsymbol{C}_{uv} \geq 0,$$

where # is a termination symbol and $c(\#)$ the associated cost (probability).

Note that Bilenko and Mooney (2003) extended this approach to the Needleman-Wunsch score with affine gap penalty and applied it to duplicate detection. To deal with the tendency of Maximum Likelihood estimators to overfit when the number of parameters is large (in this case, when the alphabet size is large), Takasu (2009) proposes a Bayesian parameter estimation of pair-HMM providing a way to smooth the estimation.

**Oncina and Sebban** The work of Oncina and Sebban (2006) describes three levels of bias induced by the use of generative models: (i) dependence between edit operations, (ii) dependence between the costs and the prior distribution of strings $p_e(\mathsf{x})$, and (iii) the fact that to obtain the posterior probability one must divide by the empirical estimate of $p_e(\mathsf{x})$. These biases are highlighted by empirical experiments conducted with the method of Ristad and Yianilos (1998). To address these limitations, they propose the use of a conditional transducer as a discriminative model that directly models the posterior probability $p(\mathsf{x}'|\mathsf{x})$ that an input string $\mathsf{x}$ is turned into an output string $\mathsf{x}'$ using edit operations.[46] Parameter estimation is also done with EM where the maximization step differs from that of Ristad and Yianilos (1998) as shown below:

$$\forall u \in \Sigma, \sum_{v\in\Sigma\cup\{\$\}} \boldsymbol{C}_{v|u} + \sum_{v\in\Sigma} \boldsymbol{C}_{v|\$} = 1, \quad \text{with } \sum_{v\in\Sigma} \boldsymbol{C}_{v|\$} + c(\#) = 1.$$

In order to allow the use of negative pairs, McCallum et al. (2005) consider another discriminative model, conditional random fields, that can deal with positive and negative pairs in specific states, still using EM for parameter estimation.

### 5.1.3 String Edit Distance Learning by Gradient Descent

The use of EM has two main drawbacks: (i) it may converge to a local optimum, and (ii) parameter estimation and distance calculations must be done at each iteration, which can be very costly if the size of the alphabet and/or the length of the strings are large.

46. An implementation is available within the SEDiL platform (Boyer et al., 2008): http://labh-curien.univ-st-etienne.fr/SEDiL/

The following methods get round these drawbacks by formulating the learning problem in the form of an optimization problem that can be efficiently solved by a gradient descent procedure.

**Saigo et al.** Saigo et al. (2006) manage to avoid the need for an iterative procedure like EM in the context of detecting remote homology in protein sequences.[47] They learn the parameters of the Smith-Waterman score which is plugged in their local alignment kernel $k_{LA}$ where all the possible local alignments $\pi$ for changing $\mathtt{x}$ into $\mathtt{x}'$ are taken into account (Saigo et al., 2004):

$$k_{LA}(x, x') = \sum_{\pi} e^{t \cdot s(x, x', \pi)}. \tag{19}$$

In the above formula, $t$ is a parameter and $s(x, x', \pi)$ is the corresponding score of $\pi$ and defined as follows:

$$s(x, x', \pi) = \sum_{u,v \in \Sigma} n_{u,v}(x, x', \pi) \cdot \boldsymbol{C}_{uv} - n_{g_d}(x, x', \pi) \cdot g_d - n_{g_e}(x, x', \pi) \cdot g_e, \tag{20}$$

where $n_{u,v}(x, x', \pi)$ is the number of times that symbol $u$ is aligned with $v$ while $g_d$ and $g_e$, along with their corresponding number of occurrences $n_{g_d}(x, x', \pi)$ and $n_{g_e}(x, x', \pi)$, are two parameters dealing respectively with the opening and extension of gaps.

Unlike the Smith-Waterman score, $k_{LA}$ is differentiable and can be optimized by a gradient descent procedure. The objective function that they optimize is meant to favor the discrimination between positive and negative examples, but this is done by only using positive pairs of distant homologs. The approach has two additional drawbacks: (i) the objective function is nonconvex and thus subject to local minima, and (ii) in general, $k_{LA}$ does not fulfill the properties of a kernel.

**GESL (Bellet et al.)** Bellet et al. (2011, 2012a) propose a convex programming approach to learn edit similarity functions from both positive and negative pairs without requiring a costly iterative procedure.[48] They use the following simplified edit function:

$$e_{\boldsymbol{C}}(\mathtt{x}, \mathtt{x}') = \sum_{(u,v) \in (\Sigma \cup \{\$\})^2 \setminus \{\$,\$\}} \boldsymbol{C}_{uv} \cdot \#_{uv}(\mathtt{x}, \mathtt{x}'),$$

where $\#_{uv}(\mathtt{x}, \mathtt{x}')$ is the number of times the operation $u \to v$ appears in the Levenshtein script. Therefore, $e_{\boldsymbol{C}}$ can be optimized directly since the sequence of operations is fixed (it does not depend on the costs). The authors optimize the nonlinear similarity $K_{\boldsymbol{C}}(\mathtt{x}, \mathtt{x}') = 2\exp(-e_{\boldsymbol{C}}(\mathtt{x}, \mathtt{x}')) - 1$, derived from $e_{\boldsymbol{C}}$. Note that $K_{\boldsymbol{C}}$ is not required to be PSD nor symmetric. GESL (Good Edit Similarity Learning) is expressed as follows:

$$\begin{aligned} \min_{\boldsymbol{C}, B_1, B_2} \quad & \frac{1}{n^2} \sum_{z_i, z_j} \ell(\boldsymbol{C}, z_i, z_j) \quad + \quad \beta \|\boldsymbol{C}\|_{\mathcal{F}}^2 \\ \text{s.t.} \quad & B_1 \geq -\log(\frac{1}{2}), \quad 0 \leq B_2 \leq -\log(\frac{1}{2}), \quad B_1 - B_2 = \eta_\gamma, \end{aligned}$$

47. Source code available at: http://sunflower.kuicr.kyoto-u.ac.jp/~hiroto/project/optaa.html
48. Source code available at: http://www-bcf.usc.edu/~bellet/

where $\beta \geq 0$ is a regularization parameter, $\eta_\gamma \geq 0$ a parameter corresponding to a desired "margin" and

$$\ell(\boldsymbol{C}, z_i, z_j) = \begin{cases} [B1 - e_{\boldsymbol{C}}(\mathsf{x_i}, \mathsf{x_j})]_+ \text{ if } y_i \neq y_j \\ [e_{\boldsymbol{C}}(\mathsf{x_i}, \mathsf{x_j}) - B2]_+ \text{ if } y_i = y_j. \end{cases}$$

GESL essentially learns the edit cost matrix $\boldsymbol{C}$ so as to optimize the $(\epsilon, \gamma, \tau)$-goodness (Balcan et al., 2008a) of the similarity $K_{\boldsymbol{C}}(\mathsf{x}, \mathsf{x}')$ and thereby enjoys generalization guarantees both for the learned similarity and for the resulting linear classifier (see Section 4.5). A potential drawback of GESL is that it optimized a simplified variant of the edit distance, although this does not seem to be an issue in practice. Note that GESL can be straightforwardly adapted to learn tree or graph edit similarities (Bellet et al., 2012a).

### 5.2 Tree and Graph Edit Distance Learning

In this section, we briefly review the main approaches in tree/graph edit distance learning. We do not delve into the details of these approaches as they are essentially adaptations of stochastic string edit distance learning presented in Section 5.1.2.

**Bernard et al.** Extending the work of Ristad and Yianilos (1998) and Oncina and Sebban (2006) on string edit similarity learning, Bernard et al. (2006, 2008) propose both a generative and a discriminative model for learning tree edit costs.[46] They rely on the tree edit distance by Selkow (1977)—which is cheaper to compute than that of Zhang and Shasha (1989)—and adapt the updates of EM to this case.

**Boyer et al.** The work of Boyer et al. (2007) tackles the more complex variant of the tree edit distance (Zhang and Shasha, 1989), which allows the insertion and deletion of single nodes instead of entire subtrees only.[46] Parameter estimation in the generative model is also based on EM.

**Dalvi et al.** The work of Dalvi et al. (2009) points out a limitation of the approach of Bernard et al. (2006, 2008): they model a distribution over tree edit scripts rather than over the trees themselves, and unlike the case of strings, there is no bijection between the edit scripts and the trees. Recovering the correct conditional probability with respect to trees requires a careful and costly procedure. They propose a more complex conditional transducer that models the conditional probability over trees and use again EM for parameter estimation.

**Emms** The work of Emms (2012) points out a theoretical limitation of the approach of Boyer et al. (2007): the authors use a factorization that turns out to be incorrect in some cases. Emms shows that a correct factorization exists when only considering the edit script of highest probability instead of all possible scripts, and derives the corresponding EM updates. An obvious drawback is that the output of the model is not the probability $p(\mathsf{x}'|\mathsf{x})$. Moreover, the approach is prone to overfitting and requires smoothing and other heuristics (such as a final step of zeroing-out the diagonal of the cost matrix).

**Neuhaus & Bunke** In their paper, Neuhaus and Bunke (2007) learn a (more general) graph edit similarity, where each edit operation is modeled by a Gaussian mixture density. Parameter estimation is done using an EM-like algorithm. Unfortunately, the approach is

intractable: the complexity of the EM procedure is exponential in the number of nodes (and so is the computation of the distance).

## 6. Conclusion and Discussion

In this survey, we provided a comprehensive review of the main methods and trends in metric learning. We here briefly summarize and draw promising lines for future research.

### 6.1 Summary

**Numerical data** While metric learning for feature vectors was still in its early life at the time of the first survey (Yang and Jin, 2006), it has now reached a good maturity level. Indeed, recent methods are able to deal with a large spectrum of settings in a scalable way. In particular, online approaches have played a significant role towards better scalability, complex tasks can be tackled through nonlinear or local metric learning, methods have been derived for difficult settings such as ranking, multi-task learning or domain adaptation, and the question of generalization in metric learning has been the focus of recent papers.

**Structured data** On the other hand, much less work has gone into metric learning for structured data and advances made for numerical data have not yet propagated to structured data. Indeed, most approaches remain based on EM-like algorithms which make them intractable for large datasets and instance size, and hard to analyze due to local optima. Nevertheless, recent advances such as GESL (Bellet et al., 2011) have shown that drawing inspiration from successful feature vector formulations (even if it requires simplifying the metric) can be highly beneficial in terms of scalability and flexibility. This is promising direction and probably a good omen for the development of this research area.

### 6.2 What next?

In light of this survey, we can identify the limitations of the current literature and speculate on where the future of metric learning is going.

**Scalability with both $n$ and $d$** There has been satisfying solutions to perform metric learning on large datasets ("Big Data") through online learning or stochastic optimization. The question of scalability with the dimensionality is more involved, since most methods learn $O(d^2)$ parameters, which is intractable for real-world applications involving thousands of features, unless dimensionality reduction is applied beforehand. Kernelized methods have $O(n^2)$ parameters instead, but this is infeasible when $n$ is also large. Therefore, the challenge of achieving high scalability with both $n$ and $d$ has yet to be overcome. Recent approaches have tackled the problem by optimizing over the manifold of low-rank matrices (Shalit et al., 2012; Cheng, 2013) or defining the metric based on a combination of simple classifiers (Kedem et al., 2012; Xiong et al., 2012). These approaches have a good potential for future research.

**More theoretical understanding** Although several recent papers have looked at the generalization of metric learning, analyzing the link between the consistency of the learned metric and its performance in a given algorithm (classifier, clustering procedure, etc) remains an important open problem. So far, only results for linear classification have been

obtained (Bellet et al., 2012b; Guo and Ying, 2014), while learned metrics are also heavily used for $k$-NN classification, clustering or information retrieval, for which no theoretical result is known.

**Unsupervised metric learning** A natural question to ask is whether one can learn a metric in a purely unsupervised way. So far, this has only been done as a byproduct of dimensionality reduction algorithms. Other relevant criteria should be investigated, for instance learning a metric that is robust to noise or invariant to some transformations of interest, in the spirit of denoising autoencoders (Vincent et al., 2008; Chen et al., 2012). Some results in this direction have been obtained for image transformations (Kumar et al., 2007). A related problem is to characterize what it means for a metric to be good for clustering. There has been preliminary work on this question (Balcan et al., 2008b; Lajugie et al., 2014), which deserves more attention.

**Leveraging the structure** The simple example of metric learning designed specifically for histogram data (Kedem et al., 2012) has shown that taking the structure of the data into account when learning the metric can lead to significant improvements in performance. As data is becoming more and more structured (e.g., social networks), using this structure to bias the choice of metric is likely to receive increasing interest in the near future.

**Adapting the metric to changing data** An important issue is to develop methods robust to changes in the data. In this line of work, metric learning in the presence of noisy data as well as for transfer learning and domain adaptation have recently received some interest. However, these efforts are still insufficient for dealing with lifelong learning applications, where the learner experiences concept drift and must detect and adapt the metric to different changes.

**Learning richer metrics** Existing metric learning algorithms ignore the fact that the notion of similarity is often multimodal: there exist several ways in which two instances may be similar (perhaps based on different features), and different degrees of similarity (versus the simple binary similar/dissimilar view). Being able to model these shades as well as to interpret why things are similar would bring the learned metrics closer to our own notions of similarity.

## Acknowledgments

We would like to acknowledge support from the ANR LAMPADA 09-EMER-007-02 project.

## References

M. Ehsan Abbasnejad, Dhanesh Ramachandram, and Mandava Rajeswari. A survey of the state of the art in learning the kernels. *Knowledge and Information Systems (KAIS)*, 31 (2):193–221, 2012.

Pierre-Antoine Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2008.

Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, 1999.

Mahdieh S. Baghshah and Saeed B. Shouraki. Semi-Supervised Metric Learning Using Pairwise Constraints. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1217–1222, 2009.

Maria-Florina Balcan, Avrim Blum, and Nathan Srebro. Improved Guarantees for Learning via Similarity Functions. In *Proceedings of the 21st Annual Conference on Learning Theory (COLT)*, pages 287–298, 2008a.

Maria-Florina Balcan, Avrim Blum, and Santosh Vempala. A Discriminative Framework for Clustering via Similarity Functions. In *ACM Symposium on Theory of Computing (STOC)*, pages 671–680, 2008b.

Aharon Bar-Hillel, Tomer Hertz, Noam Shental, and Daphna Weinshall. Learning Distance Functions using Equivalence Relations. In *Proceedings of the 20th International Conference on Machine Learning (ICML)*, pages 11–18, 2003.

Aharon Bar-Hillel, Tomer Hertz, Noam Shental, and Daphna Weinshall. Learning a Mahalanobis Metric from Equivalence Constraints. *Journal of Machine Learning Research (JMLR)*, 6:937–965, 2005.

Peter L. Bartlett and Shahar Mendelson. Rademacher and Gaussian Complexities: Risk Bounds and Structural Results. *Journal of Machine Learning Research (JMLR)*, 3:463–482, 2002.

Jonathan Baxter and Peter L. Bartlett. The Canonical Distortion Measure in Feature Space and 1-NN Classification. In *Advances in Neural Information Processing Systems (NIPS) 10*, 1997.

Mikhail Belkin and Partha Niyogi. Semi-Supervised Learning on Riemannian Manifolds. *Machine Learning Journal (MLJ)*, 56(1–3):209–239, 2004.

Aurélien Bellet. *Supervised Metric Learning with Generalization Guarantees*. PhD thesis, University of Saint-Etienne, 2012.

Aurélien Bellet and Amaury Habrard. Robustness and Generalization for Metric Learning. Technical report, University of Saint-Etienne, September 2012. arXiv:1209.1086.

Aurélien Bellet, Amaury Habrard, and Marc Sebban. Learning Good Edit Similarities with Generalization Guarantees. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD)*, pages 188–203, 2011.

Aurélien Bellet, Amaury Habrard, and Marc Sebban. Good edit similarity learning by loss minimization. *Machine Learning Journal (MLJ)*, 89(1):5–35, 2012a.

Aurélien Bellet, Amaury Habrard, and Marc Sebban. Similarity Learning for Provably Accurate Sparse Linear Classification. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, pages 1871–1878, 2012b.

Xianye Ben, Weixiao Meng, Rui Yan, and Kejun Wang. An improved biometrics technique based on metric learning approach. *Neurocomputing*, 97:44–51, 2012.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine Learning Journal (MLJ)*, 79(1-2):151–175, 2010.

Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. *Robust Optimization*. Princeton University Press, 2009.

Marc Bernard, Amaury Habrard, and Marc Sebban. Learning Stochastic Tree Edit Distance. In *Proceedings of the 17th European Conference on Machine Learning (ECML)*, pages 42–53, 2006.

Marc Bernard, Laurent Boyer, Amaury Habrard, and Marc Sebban. Learning probabilistic models of tree edit distance. *Pattern Recognition (PR)*, 41(8):2611–2629, 2008.

Jinbo Bi, Dijia Wu, Le Lu, Meizhu Liu, Yimo Tao, and Matthias Wolf. AdaBoost on low-rank PSD matrices for metric learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2617–2624, 2011.

Wei Bian. Constrained Empirical Risk Minimization Framework for Distance Metric Learning. *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, 23(8): 1194–1205, 2012.

Wei Bian and Dacheng Tao. Learning a Distance Metric by Empirical Loss Minimization. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1186–1191, 2011.

Mikhail Bilenko and Raymond J. Mooney. Adaptive Duplicate Detection Using Learnable String Similarity Measures. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 39–48, 2003.

Mikhail Bilenko, Sugato Basu, and Raymond J. Mooney. Integrating Constraints and Metric Learning in Semi-Supervised Clustering. In *Proceedings of the 21st International Conference on Machine Learning (ICML)*, pages 81–88, 2004.

Philip Bille. A survey on tree edit distance and related problems. *Theoretical Computer Science (TCS)*, 337(1-3):217–239, 2005.

Olivier Bousquet and André Elisseeff. Stability and Generalization. *Journal of Machine Learning Research (JMLR)*, 2:499–526, 2002.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

Laurent Boyer, Amaury Habrard, and Marc Sebban. Learning Metrics between Tree Structured Data: Application to Image Recognition. In *Proceedings of the 18th European Conference on Machine Learning (ECML)*, pages 54–66, 2007.

Laurent Boyer, Yann Esposito, Amaury Habrard, José Oncina, and Marc Sebban. SEDiL: Software for Edit Distance Learning. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD)*, pages 672–677, 2008. URL http://labh-curien.univ-st-etienne.fr/SEDiL/.

Lev M. Bregman. The relaxation method of finding the common points of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7(3):200–217, 1967.

Bin Cao, Xiaochuan Ni, Jian-Tao Sun, Gang Wang, and Qiang Yang. Distance Metric Learning under Covariate Shift. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1204–1210, 2011.

Qiong Cao, Zheng-Chu Guo, and Yiming Ying. Generalization Bounds for Metric and Similarity Learning. Technical report, University of Exeter, July 2012a. arXiv:1207.5437.

Qiong Cao, Yiming Ying, and Peng Li. Distance Metric Learning Revisited. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD)*, pages 283–298, 2012b.

Rich Caruana. Multitask Learning. *Machine Learning Journal (MLJ)*, 28(1):41–75, 1997.

Nicoló Cesa-Bianchi and Claudio Gentile. Improved Risk Tail Bounds for On-Line Algorithms. *IEEE Transactions on Information Theory (TIT)*, 54(1):386–390, 2008.

Ratthachat Chatpatanasiri, Teesid Korsrilabutr, Pasakorn Tangchanachaianan, and Boonserm Kijsirikul. A new kernelization framework for Mahalanobis distance learning algorithms. *Neurocomputing*, 73:1570–1579, 2010.

Gal Chechik, Uri Shalit, Varun Sharma, and Samy Bengio. An Online Algorithm for Large Scale Image Similarity Learning. In *Advances in Neural Information Processing Systems (NIPS) 22*, pages 306–314, 2009.

Gal Chechik, Varun Sharma, Uri Shalit, and Samy Bengio. Large Scale Online Learning of Image Similarity Through Ranking. *Journal of Machine Learning Research (JMLR)*, 11: 1109–1135, 2010.

Minmin Chen, Zhixiang Eddie Xu, Kilian Q. Weinberger, and Fei Sha. Marginalized Denoising Autoencoders for Domain Adaptation. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, 2012.

Li Cheng. Riemannian Similarity Learning. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, 2013.

Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a Similarity Metric Discriminatively, with Application to Face Verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 539–546, 2005.

Corinna Cortes and Vladimir Vapnik. Support-Vector Networks. *Machine Learning Journal (MLJ)*, 20(3):273–297, 1995.

Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory (TIT)*, 13(1):21–27, 1967.

Koby Crammer and Gal Chechik. Adaptive Regularization for Weight Matrices. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, 2012.

Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. Online Passive-Aggressive Algorithms. *Journal of Machine Learning Research (JMLR)*, 7:551–585, 2006.

Marco Cuturi and David Avis. Ground Metric Learning. Technical report, Kyoto University, 2011. 1110.2306.

Nilesh N. Dalvi, Philip Bohannon, and Fei Sha. Robust web extraction: an approach based on a probabilistic tree-edit model. In *Proceedings of the ACM SIGMOD International Conference on Management of data (COMAD)*, pages 335–348, 2009.

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 (ICML)*, pages 209–216, 2007.

Margaret O. Dayhoff, Robert M. Schwartz, and Bruce C. Orcutt. A model of evolutionary change in proteins. *Atlas of protein sequence and structure*, 5(3):345–351, 1978.

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

Jia Deng, Alexander C. Berg, and Li Fei-Fei. Hierarchical semantic indexing for large scale image retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 785–792, 2011.

Matthew Der and Lawrence K. Saul. Latent Coincidence Analysis: A Hidden Variable Model for Distance Metric Learning. In *Advances in Neural Information Processing Systems (NIPS) 25*, pages 3239–3247, 2012.

Inderjit S. Dhillon and Joel A. Tropp. Matrix Nearness Problems with Bregman Divergences. *SIAM Journal on Matrix Analysis and Applications*, 29(4):1120–1146, 2007.

Huyen Do, Alexandros Kalousis, Jun Wang, and Adam Woznica. A metric learning perspective of SVM: on the relation of LMNN and SVM. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 308–317, 2012.

John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Ambuj Tewari. Composite Objective Mirror Descent. In *Proceedings of the 23rd Annual Conference on Learning Theory (COLT)*, pages 14–26, 2010.

Martin Emms. On Stochastic Tree Distances and Their Training via Expectation-Maximisation. In *Proceedings of the 1st International Conference on Pattern Recognition Applications and Methods (ICPRAM)*, pages 144–153, 2012.

Theodoros Evgeniou and Massimiliano Pontil. Regularized multi-task learning. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 109–117, 2004.

Maryam Fazel, Haitham Hindi, and Stephen P. Boyd. A Rank Minimization Heuristic with Application to Minimum Order System Approximation. In *Proceedings of the American Control Conference*, pages 4734–4739, 2001.

Imola K. Fodor. A Survey of Dimension Reduction Techniques. Technical report, Lawrence Livermore National Laboratory, 2002. UCRL-ID- 148494.

Yoav Freund and Robert E. Schapire. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. In *Proceedings of the 2nd European Conference on Computational Learning Theory (EuroCOLT)*, pages 23–37, 1995.

Jerome H. Friedman. Flexible Metric Nearest Neighbor Classification. Technical report, Department of Statistics, Stanford University, 1994.

Jerome H. Friedman. Greedy Function Approximation: A Gradient Boosting Machine. *Annals of Statistics (AOS)*, 29(5):1189–1232, 2001.

Andrea Frome, Yoram Singer, Fei Sha, and Jitendra Malik. Learning Globally-Consistent Local Distance Functions for Shape-Based Image Retrieval and Classification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1–8, 2007.

Keinosuke Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, 1990.

Xinbo Gao, Bing Xiao, Dacheng Tao, and Xuelong Li. A survey of graph edit distance. *Pattern Analysis and Applications (PAA)*, 13(1):113–129, 2010.

Bo Geng, Dacheng Tao, and Chao Xu. DAML: Domain Adaptation Metric Learning. *IEEE Transactions on Image Processing (TIP)*, 20(10):2980–2989, 2011.

Amir Globerson and Sam T. Roweis. Metric Learning by Collapsing Classes. In *Advances in Neural Information Processing Systems (NIPS) 18*, pages 451–458, 2005.

Jacob Goldberger, Sam Roweis, Geoff Hinton, and Ruslan Salakhutdinov. Neighbourhood Components Analysis. In *Advances in Neural Information Processing Systems (NIPS) 17*, pages 513–520, 2004.

Mehmet Gönen and Ethem Alpaydin. Multiple Kernel Learning Algorithms. *Journal of Machine Learning Research (JMLR)*, 12:2211–2268, 2011.

Yves Grandvalet and Yoshua Bengio. Semi-supervised Learning by Entropy Minimization. In *Advances in Neural Information Processing Systems (NIPS) 17*, pages 29–536, 2004.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander J. Smola. A Kernel Method for the Two-Sample-Problem. In *Advances in Neural Information Processing Systems (NIPS) 19*, pages 513–520, 2006.

Matthieu Guillaumin, Thomas Mensink, Jakob J. Verbeek, and Cordelia Schmid. TagProp: Discriminative metric learning in nearest neighbor models for image auto-annotation. In *Proceddings of the IEEE International Conference on Computer Vision (ICCV)*, pages 309–316, 2009a.

Matthieu Guillaumin, Jakob J. Verbeek, and Cordelia Schmid. Is that you? Metric learning approaches for face identification. In *Proceddings of the IEEE International Conference on Computer Vision (ICCV)*, pages 498–505, 2009b.

Zheng-Chu Guo and Yiming Ying. Guaranteed Classification via Regularized Similarity Learning. *Neural Computation*, 26(3):497–522, 2014.

James L. Hafner, Harpreet S. Sawhney, William Equitz, Myron Flickner, and Wayne Niblack. Efficient Color Histogram Indexing for Quadratic Form Distance Functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 17(7):729–736, 1995.

Trevor Hastie and Robert Tibshirani. Discriminant Adaptive Nearest Neighbor Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 18(6): 607–616, 1996.

Søren Hauberg, Oren Freifeld, and Michael J. Black. A Geometric take on Metric Learning. In *Advances in Neural Information Processing Systems (NIPS) 25*, pages 2033–2041, 2012.

Yujie He, Wenlin Chen, and Yixin Chen. Kernel Density Metric Learning. Technical report, Washington University in St. Louis, 2013.

Steven Henikoff and Jorja G. Henikoff. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences of the United States of America*, 89(22):10915–10919, 1992.

Steven C. Hoi, Wei Liu, Michael R. Lyu, and Wei-Ying Ma. Learning Distance Metrics with Contextual Constraints for Image Retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2072–2078, 2006.

Steven C. Hoi, Wei Liu, and Shih-Fu Chang. Semi-supervised distance metric learning for Collaborative Image Retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

Steven C. Hoi, Wei Liu, and Shih-Fu Chang. Semi-supervised distance metric learning for collaborative image retrieval and clustering. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, 6(3), 2010.

Yi Hong, Quannan Li, Jiayan Jiang, and Zhuowen Tu. Learning a mixture of sparse distance metrics for classification and dimensionality reduction. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 906–913, 2011.

Kaizhu Huang, Yiming Ying, and Colin Campbell. GSML: A Unified Framework for Sparse Metric Learning. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, pages 189–198, 2009.

Kaizhu Huang, Rong Jin, Zenglin Xu, and Cheng-Lin Liu. Robust Metric Learning by Smooth Optimization. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 244–251, 2010.

Kaizhu Huang, Yiming Ying, and Colin Campbell. Generalized sparse metric learning with relative comparisons. *Knowledge and Information Systems (KAIS)*, 28(1):25–45, 2011.

Yinjie Huang, Cong Li, Michael Georgiopoulos, and Georgios C. Anagnostopoulos. Reduced-Rank Local Distance Metric Learning. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD)*, pages 224–239, 2013.

Prateek Jain, Brian Kulis, Inderjit S. Dhillon, and Kristen Grauman. Online Metric Learning and Fast Similarity Search. In *Advances in Neural Information Processing Systems (NIPS) 21*, pages 761–768, 2008.

Prateek Jain, Brian Kulis, and Inderjit S. Dhillon. Inductive Regularized Learning of Kernel Functions. In *Advances in Neural Information Processing Systems (NIPS) 23*, pages 946–954, 2010.

Prateek Jain, Brian Kulis, Jason V. Davis, and Inderjit S. Dhillon. Metric and Kernel Learning Using a Linear Transformation. *Journal of Machine Learning Research (JMLR)*, 13:519–547, 2012.

Nan Jiang, Wenyu Liu, and Ying Wu. Order determination and sparsity-regularized metric learning adaptive visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1956–1963, 2012.

Rong Jin, Shijun Wang, and Yang Zhou. Regularized Distance Metric Learning: Theory and Algorithm. In *Advances in Neural Information Processing Systems (NIPS) 22*, pages 862–870, 2009.

Thorsten Joachims, Thomas Finley, and Chun-Nam J. Yu. Cutting-plane training of structural SVMs. *Machine Learning Journal (MLJ)*, 77(1):27–59, 2009.

Purushottam Kar, Bharath Sriperumbudur Prateek Jain, and Harish Karnick. On the Generalization Ability of Online Learning Algorithms for Pairwise Loss Functions. In *Proceedings of the 30th International Conference on Machine Learning*, 2013.

Tsuyoshi Kato and Nozomi Nagano. Metric learning for enzyme active-site search. *Bioinformatics*, 26(21):2698–2704, 2010.

Dor Kedem, Stephen Tyree, Kilian Weinberger, Fei Sha, and Gert Lanckriet. Non-linear Metric Learning. In *Advances in Neural Information Processing Systems (NIPS) 25*, pages 2582–2590, 2012.

Brian Kulis. Metric Learning: A Survey. *Foundations and Trends in Machine Learning (FTML)*, 5(4):287–364, 2012.

Brian Kulis, Prateek Jain, and Kristen Grauman. Fast Similarity Search for Learned Metrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 31(12): 2143–2157, 2009.

Brian Kulis, Kate Saenko, and Trevor Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1785–1792, 2011.

Solomon Kullback and Richard Leibler. On Information and Sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86, 1951.

M. Pawan Kumar, Philip H. S. Torr, and Andrew Zisserman. An Invariant Large Margin Nearest Neighbour Classifier. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1–8, 2007.

Gautam Kunapuli and Jude Shavlik. Mirror Descent for Metric Learning: A Unified Approach. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Database (ECML/PKDD)*, pages 859–874, 2012.

Rémi Lajugie, Sylvain Arlot, and Francis Bach. Large-Margin Metric Learning for Constrained Partitioning Problems. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, 2014.

Marc T. Law, Carlos S. Gutierrez, Nicolas Thome, and Stéphane Gançarski. Structural and visual similarity learning for Web page archiving. In *Proceedings of the 10th International Workshop on Content-Based Multimedia Indexing (CBMI)*, pages 1–6, 2012.

Guy Lebanon. Metric Learning for Text Documents. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 28(4):497–508, 2006.

Jung-Eun Lee, Rong Jin, and Anil K. Jain. Rank-based distance metric learning: An application to image retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

Vladimir I. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics-Doklandy*, 6:707–710, 1966.

Fei-Fei Li and Pietro Perona. A Bayesian Hierarchical Model for Learning Natural Scene Categories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 524–531, 2005.

Xi Li, Chunhua Shen, Qinfeng Shi, Anthony Dick, and Anton van den Hengel. Non-sparse Linear Representations for Visual Tracking with Online Reservoir Metric Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1760–1767, 2012.

Zhaohui Liang, Gang Zhang, Li Jiang, and Wenbin Fu. Learning a Consistent PRO-Outcomes Metric through KCCA for an Efficacy Assessing Model of Acupuncture. *Journal of Chinese Medicine Research and Development (JCMRD)*, 1(3):79–88, 2012.

Daryl K. Lim, Brian McFee, and Gert Lanckriet. Robust Structural Metric Learning. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, 2013.

Nick Littlestone. Learning Quickly When Irrelevant Attributes Abound: A New Linear-Threshold Algorithm. *Machine Learning Journal (MLJ)*, 2(4):285–318, 1988.

Meizhu Liu and Baba C. Vemuri. A Robust and Efficient Doubly Regularized Metric Learning Approach. In *Proceedings of the 12th European Conference on Computer Vision (ECCV)*, pages 646–659, 2012.

Wei Liu, Shiqian Ma, Dacheng Tao, Jianzhuang Liu, and Peng Liu. Semi-Supervised Sparse Metric Learning using Alternating Linearization Optimization. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1139–1148, 2010.

Stuart P. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory (TIT)*, 28:129–137, 1982.

Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Cristianini, and Chris Watkins. Text Classification using String Kernels. *Journal of Machine Learning Research (JMLR)*, 2:419–444, 2002.

Jiwen Lu, Junlin Hu, Xiuzhuang Zhou, Yuanyuan Shang, Yap-Peng Tan, and Gang Wang. Neighborhood repulsed metric learning for kinship verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2594–2601, 2012.

Prasanta Chandra Mahalanobis. On the generalised distance in statistics. *Proceedings of the National Institute of Sciences of India*, 2(1):49–55, 1936.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain Adaptation: Learning Bounds and Algorithms. In *Proceedings of the 22nd Annual Conference on Learning Theory (COLT)*, 2009.

Andrew McCallum, Kedar Bellare, and Fernando Pereira. A Conditional Random Field for Discriminatively-trained Finite-state String Edit Distance. In *Proceedings of the 21st Conference in Uncertainty in Artificial Intelligence (UAI)*, pages 388–395, 2005.

Brian McFee and Gert R. G. Lanckriet. Metric Learning to Rank. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, pages 775–782, 2010.

Brian McFee, Luke Barrington, and Gert R. G. Lanckriet. Learning Content Similarity for Music Recommendation. *IEEE Transactions on Audio, Speech & Language Processing (TASLP)*, 20(8):2207–2218, 2012.

Thomas Mensink, Jakob J. Verbeek, Florent Perronnin, and Gabriela Csurka. Metric Learning for Large Scale Image Classification: Generalizing to New Classes at Near-Zero Cost. In *Proceedings of the 12th European Conference on Computer Vision (ECCV)*, pages 488–501, 2012.

Luisa Micó and Jose Oncina. Comparison of fast nearest neighbour classifiers for handwritten character recognition. *Pattern Recognition Letters (PRL)*, 19:351–356, 1998.

Saul B. Needleman and Christian D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology (JMB)*, 48(3):443–453, 1970.

Yurii Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103:127–152, 2005.

Michel Neuhaus and Horst Bunke. Automatic learning of cost functions for graph edit distance. *Journal of Information Science (JIS)*, 177(1):239–247, 2007.

Behnam Neyshabur, Nati Srebro, Ruslan Salakhutdinov, Yury Makarychev, and Payman Yadollahpour. The Power of Asymmetry in Binary Hashing. In *Advances in Neural Information Processing Systems (NIPS) 26*, pages 2823–2831, 2013.

Hieu V. Nguyen and Li Bai. Cosine Similarity Metric Learning for Face Verification. In *Proceedings of the 10th Asian Conference on Computer Vision (ACCV)*, pages 709–720, 2010.

Nam Nguyen and Yunsong Guo. Metric Learning: A Support Vector Approach. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD)*, pages 125–136, 2008.

Gang Niu, Bo Dai, Makoto Yamada, and Masashi Sugiyama. Information-theoretic Semi-supervised Metric Learning via Entropy Regularization. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, 2012.

Yung-Kyun Noh, Byoung-Tak Zhang, and Daniel D. Lee. Generative Local Metric Learning for Nearest Neighbor Classification. In *Advances in Neural Information Processing Systems (NIPS) 23*, pages 1822–1830, 2010.

Mohammad Norouzi, David J. Fleet, and Ruslan Salakhutdinov. Hamming Distance Metric Learning. In *Advances in Neural Information Processing Systems (NIPS) 25*, pages 1070–1078, 2012a.

Mohammad Norouzi, Ali Punjani, and David J. Fleet. Fast Search in Hamming Space with Multi-Index Hashing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012b.

Jose Oncina and Marc Sebban. Learning Stochastic Edit Distance: application in handwritten character recognition. *Pattern Recognition (PR)*, 39(9):1575–1587, 2006.

Shibin Parameswaran and Kilian Q. Weinberger. Large Margin Multi-Task Metric Learning. In *Advances in Neural Information Processing Systems (NIPS) 23*, pages 1867–1875, 2010.

Kyoungup Park, Chunhua Shen, Zhihui Hao, and Junae Kim. Efficiently Learning a Distance Metric for Large Margin Nearest Neighbor Classification. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence*, 2011.

Karl Pearson. On Lines and Planes of Closest Fit to Points in Space. *Philosophical Magazine*, 2(6):559–572, 1901.

Ali M. Qamar and Eric Gaussier. Online and Batch Learning of Generalized Cosine Similarities. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, pages 926–931, 2009.

Ali M. Qamar and Eric Gaussier. RELIEF Algorithm and Similarity Learning for k-NN. *International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM)*, 4:445–458, 2012.

Ali M. Qamar, Eric Gaussier, Jean-Pierre Chevallet, and Joo-Hwee Lim. Similarity Learning for Nearest Neighbor Classification. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, pages 983–988, 2008.

Guo-Jun Qi, Jinhui Tang, Zheng-Jun Zha, Tat-Seng Chua, and Hong-Jiang Zhang. An Efficient Sparse Metric Learning in High-Dimensional Space via l1-Penalized Log-Determinant Regularization. In *Proceedings of the 26th International Conference on Machine Learning (ICML)*, 2009.

Qi Qian, Rong Jin, Jinfeng Yi, Lijun Zhang, and Shenghuo Zhu. Efficient Distance Metric Learning by Adaptive Sampling and Mini-Batch Stochastic Gradient Descent (SGD). arXiv:1304.1192, April 2013.

Joaquin Quiñonero-Candela. *Dataset Shift in Machine Learning*. MIT Press, 2009.

Deva Ramanan and Simon Baker. Local Distance Functions: A Taxonomy, New Algorithms, and an Evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 33(4):794–806, 2011.

Pradeep Ravikumar, Martin J. Wainwright, Garvesh Raskutti, and Bin Yu. High-dimensional covariance estimation by minimizing $\ell_1$-penalized log-determinant divergence. *Electronic Journal of Statistics*, 5:935–980, 2011.

Eric S. Ristad and Peter N. Yianilos. Learning String-Edit Distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 20(5):522–532, 1998.

Romer Rosales and Glenn Fung. Learning Sparse Metrics via Linear Programming. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 367–373, 2006.

Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. The Earth Mover's Distance as a Metric for Image Retrieval. *International Journal of Computer Vision (IJCV)*, 40(2): 99–121, 2000.

Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting Visual Category Models to New Domains. In *Proceedings of the 11th European Conference on Computer Vision (ECCV)*, pages 213–226, 2010.

Hiroto Saigo, Jean-Philippe Vert, Nobuhisa Ueda, and Tatsuya Akutsu. Protein homology detection using string alignment kernels. *Bioinformatics*, 20(11):1682–1689, 2004.

Hiroto Saigo, Jean-Philippe Vert, and Tatsuya Akutsu. Optimizing amino acid substitution matrices with a local alignment kernel. *Bioinformatics*, 7(246):1–12, 2006.

Ruslan Salakhutdinov and Geoffrey E. Hinton. Learning a Nonlinear Embedding by Preserving Class Neighbourhood Structure. In *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 412–419, 2007.

Gerard Salton, Andrew Wong, and C. S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

Robert E. Schapire and Yoav Freund. *Boosting: Foundations and Algorithms*. MIT Press, 2012.

Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation (NECO)*, 10(1):1299–1319, 1998.

Matthew Schultz and Thorsten Joachims. Learning a Distance Metric from Relative Comparisons. In *Advances in Neural Information Processing Systems (NIPS) 16*, 2003.

Stanley M. Selkow. The tree-to-tree editing problem. *Information Processing Letters*, 6(6): 184–186, 1977.

Shai Shalev-Shwartz, Yoram Singer, and Andrew Y. Ng. Online and batch learning of pseudo-metrics. In *Proceedings of the 21st International Conference on Machine Learning (ICML)*, 2004.

Uri Shalit, Daphna Weinshall, and Gal Chechik. Online Learning in The Manifold of Low-Rank Matrices. In *Advances in Neural Information Processing Systems (NIPS) 23*, pages 2128–2136, 2010.

Uri Shalit, Daphna Weinshall, and Gal Chechik. Online Learning in the Embedded Manifold of Low-rank Matrices. *Journal of Machine Learning Research (JMLR)*, 13:429–458, 2012.

Blake Shaw, Bert C. Huang, and Tony Jebara. Learning a Distance Metric from a Network. In *Advances in Neural Information Processing Systems (NIPS) 24*, pages 1899–1907, 2011.

Chunhua Shen, Junae Kim, Lei Wang, and Anton van den Hengel. Positive Semidefinite Metric Learning with Boosting. In *Advances in Neural Information Processing Systems (NIPS) 22*, pages 1651–1660, 2009.

Chunhua Shen, Junae Kim, Lei Wang, and Anton van den Hengel. Positive Semidefinite Metric Learning Using Boosting-like Algorithms. *Journal of Machine Learning Research (JMLR)*, 13:1007–1036, 2012.

Noam Shental, Tomer Hertz, Daphna Weinshall, and Misha Pavel. Adjustment Learning and Relevant Component Analysis. In *Proceedings of the 7th European Conference on Computer Vision (ECCV)*, pages 776–792, 2002.

Yuan Shi, Yung-Kyun Noh, Fei Sha, and Daniel D. Lee. Learning Discriminative Metrics via Generative Models and Kernel Learning. arXiv:1109.3940, September 2011.

Robert D. Short and Keinosuke Fukunaga. The optimal distance measure for nearest neighbor classification. *IEEE Transactions on Information Theory (TIT)*, 27(5):622–626, 1981.

Josef Sivic and Andrew Zisserman. Efficient visual search of videos cast as text retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 31:591–606, 2009.

Temple F. Smith and Michael S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology (JMB)*, 147(1):195–197, 1981.

Atsuhiro Takasu. Bayesian Similarity Model Estimation for Approximate Recognized Text Search. In *Proceedings of the 10th International Conference on Document Analysis and Recognition (ICDAR)*, pages 611–615, 2009.

Daniel Tarlow, Kevin Swersky, Ilya Sutskever, Laurent Charlin, and Rich Zemel. Stochastic k-Neighborhood Selection for Supervised and Unsupervised Learning. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, 2013.

Matthew E. Taylor, Brian Kulis, and Fei Sha. Metric learning for reinforcement learning agents. In *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 777–784, 2011.

Lorenzo Torresani and Kuang-Chih Lee. Large Margin Component Analysis. In *Advances in Neural Information Processing Systems (NIPS) 19*, pages 1385–1392, 2006.

Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. Large Margin Methods for Structured and Interdependent Output Variables. *Journal of Machine Learning Research*, 6:1453–1484, 2005.

Yuta Tsuboi, Hisashi Kashima, Shohei Hido, Steffen Bickel, and Masashi Sugiyama. Direct Density Ratio Estimation for Large-scale Covariate Shift Adaptation. In *Proceedings of the SIAM International Conference on Data Mining (SDM)*, pages 443–454, 2008.

Leslie G. Valiant. A theory of the learnable. *Communications of the ACM*, 27:1134–1142, 1984.

Laurens J.P. van der Maaten, Eric O. Postma, and H. Jaap van den Herik. Dimensionality Reduction: A Comparative Review. Technical report, Tilburg University, 2009. TiCC-TR 2009-005.

Vladimir N. Vapnik and Alexey Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications (TPA)*, 16(2):264–280, 1971.

Nakul Verma, Dhruv Mahajan, Sundararajan Sellamanickam, and Vinod Nair. Learning Hierarchical Similarity Metrics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2280–2287, 2012.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning (ICML)*, pages 1096–1103, 2008.

Fan Wang and Leonidas J. Guibas. Supervised Earth Mover's Distance Learning and Its Computer Vision Applications. In *Proceedings of the 12th European Conference on Computer Vision (ECCV)*, pages 442–455, 2012.

Jingyan Wang, Xin Gao, Quanquan Wang, and Yongping Li. ProDis-ContSHC: learning protein dissimilarity measures and hierarchical context coherently for protein-protein comparison in protein database retrieval. *BMC Bioinformatics*, 13(S-7):S2, 2012a.

Jun Wang, Huyen T. Do, Adam Woznica, and Alexandros Kalousis. Metric Learning with Multiple Kernels. In *Advances in Neural Information Processing Systems (NIPS) 24*, pages 1170–1178, 2011.

Jun Wang, Adam Woznica, and Alexandros Kalousis. Learning Neighborhoods for Metric Learning. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD)*, pages 223–236, 2012b.

Jun Wang, Adam Woznica, and Alexandros Kalousis. Parametric Local Metric Learning for Nearest Neighbor Classification. In *Advances in Neural Information Processing Systems (NIPS) 25*, pages 1610–1618, 2012c.

Qianying Wang, Pong C. Yuen, and Guocan Feng. Semi-supervised metric learning via topology preserving multiple semi-supervised assumptions. *Pattern Recognition (PR)*, 2013a.

Yuyang Wang, Roni Khardon, Dmitry Pechyony, and Rosie Jones. Generalization Bounds for Online Learning Algorithms with Pairwise Loss Functions. In *Proceedings of the 25th Annual Conference on Learning Theory (COLT)*, pages 13.1–13.22, 2012d.

Yuyang Wang, Roni Khardon, Dmitry Pechyony, and Rosie Jones. Online Learning with Pairwise Loss Functions. Technical report, Tufts University, January 2013b. arXiv:1301.5332.

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 (ICML)*, pages 1160–1167, 2008.

Kilian Q. Weinberger and Lawrence K. Saul. Distance Metric Learning for Large Margin Nearest Neighbor Classification. *Journal of Machine Learning Research (JMLR)*, 10: 207–244, 2009.

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 (NIPS) 18*, pages 1473–1480, 2005.

Lei Wu, Rong Jin, Steven C.-H. Hoi, Jianke Zhu, and Nenghai Yu. Learning Bregman Distance Functions and Its Application for Semi-Supervised Clustering. In *Advances in Neural Information Processing Systems (NIPS) 22*, pages 2089–2097, 2009.

Lei Wu, Steven C.-H. Hoi, Rong Jin, Jianke Zhu, and Nenghai Yu. Learning Bregman Distance Functions for Semi-Supervised Clustering. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 24(3):478–491, 2012.

Eric P. Xing, Andrew Y. Ng, Michael I. Jordan, and Stuart J. Russell. Distance Metric Learning with Application to Clustering with Side-Information. In *Advances in Neural Information Processing Systems (NIPS) 15*, pages 505–512, 2002.

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, 2012.

Huilin Xiong and Xue-Wen Chen. Kernel-based distance metric learning for microarray data classification. *BMC Bioinformatics*, 7:299, 2006.

Huan Xu and Shie Mannor. Robustness and Generalization. *Machine Learning Journal (MLJ)*, 86(3):391–423, 2012.

Zhixiang Xu, Kilian Q. Weinberger, and Olivier Chapelle. Distance Metric Learning for Kernel Machines. arXiv:1208.3422, 2012.

Liu Yang and Rong Jin. Distance Metric Learning: A Comprehensive Survey. Technical report, Department of Computer Science and Engineering, Michigan State University, 2006.

Peipei Yang, Kaizhu Huang, and Cheng-Lin Liu. Multi-Task Low-Rank Metric Learning Based on Common Subspace. In *Proceedings of the 18th International Conference on Neural Information Processing (ICONIP)*, pages 151–159, 2011.

Peipei Yang, Kaizhu Huang, and Cheng-Lin Liu. Geometry Preserving Multi-task Metric Learning. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD)*, pages 648–664, 2012.

Dit-Yan Yeung and Hong Chang. Extending the relevant component analysis algorithm for metric learning using both positive and negative equivalence constraints. *Pattern Recognition (PR)*, 39(5):1007–1010, 2006.

Yiming Ying and Peng Li. Distance Metric Learning with Eigenvalue Optimization. *Journal of Machine Learning Research (JMLR)*, 13:1–26, 2012.

Yiming Ying, Kaizhu Huang, and Colin Campbell. Sparse Metric Learning via Smooth Optimization. In *Advances in Neural Information Processing Systems (NIPS) 22*, pages 2214–2222, 2009.

Jun Yu, Meng Wang, and Dacheng Tao. Semisupervised Multiview Distance Metric Learning for Cartoon Synthesis. *IEEE Transactions on Image Processing (TIP)*, 21(11):4636–4648, 2012.

Zheng-Jun Zha, Tao Mei, Meng Wang, Zengfu Wang, and Xian-Sheng Hua. Robust Distance Metric Learning with Auxiliary Knowledge. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1327–1332, 2009.

De-Chuan Zhan, Ming Li, Yu-Feng Li, and Zhi-Hua Zhou. Learning instance specific distances using metric propagation. In *Proceedings of the 26th International Conference on Machine Learning*, 2009.

Changshui Zhang, Feiping Nie, and Shiming Xiang. A general kernelization framework for learning algorithms based on kernel PCA. *Neurocomputing*, 73(4–6):959–967, 2010.

Kaizhong Zhang and Dennis Shasha. Simple fast algorithms for the editing distance between trees and related problems. *SIAM Journal of Computing (SICOMP)*, 18(6):1245–1262, 1989.

Yu Zhang and Dit-Yan Yeung. Transfer metric learning by learning task relationships. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1199–1208, 2010.

Guoqiang Zhong, Kaizhu Huang, and Cheng-Lin Liu. Low Rank Metric Learning with Manifold Regularization. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, pages 1266–1271, 2011.