Data Min Knowl Disc (2015) 29:534–564
DOI 10.1007/s10618-014-0356-z

Survey on distance metric learning and dimensionality reduction in data mining

Fei Wang · Jimeng Sun

Received: 6 November 2012 / Accepted: 24 May 2014 / Published online: 27 June 2014

Abstract Distance metric learning is a fundamental problem in data mining and knowledge discovery. Many representative data mining algorithms, such as k-nearest neighbor classifier, hierarchical clustering and spectral clustering, heavily rely on the underlying distance metric for correctly measuring relations among input data. In recent years, many studies have demonstrated, either theoretically or empirically, that learning a good distance metric can greatly improve the performance of classification, clustering and retrieval tasks. In this survey, we overview existing distance metric learning approaches according to a common framework. Specifically, depending on the available supervision information during the distance metric learning process, we categorize each distance metric learning algorithm as *supervised, unsupervised* or *semi-supervised*. We compare those different types of metric learning methods, point out their strength and limitations. Finally, we summarize open challenges in distance metric learning and propose future directions for distance metric learning.

Keywords Distance metric learning · Dimensionality reduction · Data mining

Responsible editor: Ian Davidson.

F. Wang (✉)
Healthcare Analytics Research Group,
IBM T. J. Watson Research Center, Yorktown Heights, NY, USA
e-mail: fwang@us.ibm.com; feiwang03@gmail.com

J. Sun
School of Computational Science and Engineering, College of Computing,
Georgia Institute of Technology, Atlanta, GA, USA
e-mail: jsun@cc.gatech.edu

1 Introduction

Distance metric learning is a fundamental problem in many applications. In information retrieval applications, the underlying distances between the query and documents determine the retrieval ranks. In clinical decision support applications, physicians are interested in finding out similar patients given a query patient (Sun et al. 2010; Wang et al. 2011a, b). Learning a proper distance metric is also at the core of many popular data mining and knowledge discover algorithms, such as *k-Nearest Neighbor* (kNN) classifier (Duda et al. 2001), hierarchical agglomerative clustering (HAC) (Jain and Dubes 1988), and spectral clustering (SC) (Luxburg 2007). In recent years, many studies have demonstrated Kulis (2010), Werman et al. (2010), and Yang and Jin (2006), either theoretically or empirically, that learning a good distance metric can greatly improve the performance of classification (Weinberger and Saul 2009), clustering (Domeniconi et al. 2007) and retrieval (He et al. 2006) tasks.

In this survey, we will give an overview of the existing distance metric learning approaches and point out their strength and limitations, as well as present challenges and future research directions. We categorize distance learning algorithms as *supervised*, *unsupervised* or *semi-supervised* according to the availability of supervision information during the distance metric learning process. If complete supervision information (e.g., labels) for a data set is required, this distance metric learning approach is called supervised metric learning; If no supervision information is used to construct a distance metric, the approach is called unsupervised metric learning. Finally, if both labeled and unlabeled data are used to learn a distance metric, this approach is called a semi-supervised metric learning method.

In the rest of this survey, we will first introduce the definition of distance metric in Sect. 2. Then we will overview supervised, unsupervised and semi-supervised distance metric learning algorithms in detail in Sect. 3. After that, we present advanced topics in distance metric learning in Sect. 4. Finally, we conclude the survey with challenges and open problems in distance metric learning in Sect. 5. Table 1 summarizes the notations and symbols that will be used throughout the paper.

Difference with Existing Surveys There has been quite a few existing surveys and tutorials on distance metric learning. For example, Yang and Jin (2006) wrote one of the early metric learning survey summarizing the metric learning approaches until 2006. Werman et al. (2010) gave a tutorial on distance functions and metric learning in computer vision applications. Kulis (2010) gave a comprehensive tutorial on distance metric learning and later published a monograph Kulis (2012) on this topic. Different from those existing tutorials and surveys, this paper summarizes the metric learning approaches from a different perspective—dimensionality reduction. We point out that most of the existing metric learning approaches can be viewed as a standard Euclidean distance in some embedding space. Thus distance metric learning and dimensionality reduction can be analyzed from a unified point of view. We surveyed a set of representative distance metric learning and dimensionality reduction methods. According to the availability of supervision information, we categorize each approach as unsupervised, supervised or semi-supervised. In the last part of this survey, we also introduce some advanced topics including online learning, active learning and transfer learning.

Table 1 The meanings of various symbols that will be used throughout the paper

Symbols	Meaning
n	Number of data
d	Data dimensionality
$\mathbf{x}_i$	The i-th data vector
$\mathbf{X}$	Data matrix
$\mathbf{M}$	Precision matrix of the generalized Mahalanobis distance
$\mathbf{w}_i$	The i-th projection vector
$\mathbf{W}$	Projection matrix
$\mathcal{N}_i$	The neighborhood of $\mathbf{x}_i$
$\phi(\cdot)$	Nonlinear mapping used in kernel methods
$\mathbf{K}$	Kernel matrix
$\mathbf{L}$	Laplacian matrix

2 The definition of distance metric learning

Before describing different types of distance metric learning algorithms, we first define necessary notations and concepts on distance metric learning.

Throughout the paper, we use $\mathcal{X}$ to represent a set of data points. If $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathcal{X}$ are data vectors with the same dimensionality, we call $\mathcal{D} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ a **Distance Metric** if it satisfies the following four properties[1]:

- Nonnegativity: $\mathcal{D}(\mathbf{x}, \mathbf{y}) \geqslant 0$
- Coincidence: $\mathcal{D}(\mathbf{x}, \mathbf{y}) = 0$ if and only if $\mathbf{x} = \mathbf{y}$
- Symmetry: $\mathcal{D}(\mathbf{x}, \mathbf{y}) = \mathcal{D}(\mathbf{y}, \mathbf{x})$
- Subadditivity: $\mathcal{D}(\mathbf{x}, \mathbf{y}) + \mathcal{D}(\mathbf{y}, \mathbf{z}) \geqslant \mathcal{D}(\mathbf{x}, \mathbf{z})$

If we relax the coincidence condition to *if* $\mathbf{x} = \mathbf{y} \Rightarrow \mathcal{D}(\mathbf{x}, \mathbf{y}) = 0$, then $\mathcal{D}$ is called a **Pseudo Metric**. There are many well-known distance metrics. Here we list several examples:

- *Euclidean distance*, which measures the distance between $\mathbf{x}$ and $\mathbf{y}$ by

$$\mathcal{D}(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})^\top (\mathbf{x} - \mathbf{y})} \tag{1}$$

- *Cosine distance*, which measures the distance between $\mathbf{x}$ and $\mathbf{y}$ by

$$\mathcal{D}(\mathbf{x}, \mathbf{y}) = \sqrt{1 - \frac{\mathbf{x}^\top \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|}} \tag{2}$$

  where $\| \cdot \|$ is the vector norm operator. The Cosine distance is often used to measure the distance between pairwise documents. Note that Cosine distance will

[1] http://en.wikipedia.org/wiki/Metric_(mathematics).

be equivalent to Euclidean distance if we normalize every data vector to have a unit norm. Note that Cosine distance is not well defined for zero vectors, thus it is not a strict distance metric.

- χ^2 *distance*, which measures the distance between $\mathbf{x}$ and $\mathbf{y}$ by

$$\mathcal{D}(\mathbf{x}, \mathbf{y}) = \sqrt{\frac{1}{2}\sum_{i=1}^{d}\frac{(x_i - y_i)^2}{x_i + y_i}} \tag{3}$$

  where d is the dimensionality of $\mathbf{x}$ and $\mathbf{y}$. The χ^2 distance is usually used to measure the distance between two discrete distributions (histograms) Pele and Werman (2010). Note that χ^2 distance is not well defined for two zero vectors, thus it is not a strict distance metric.

- *Mahalanobis distance*[2], which measures the distance between $\mathbf{x}$ and $\mathbf{y}$ by

$$\mathcal{D}(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})^\top \mathbf{S}(\mathbf{x} - \mathbf{y})} \tag{4}$$

  where $\mathbf{S}$ is the inverse of the data covariance matrix (also referred to as the precision matrix)[3].

- *Generalized Mahalanobis distance*, which measures the distance between $\mathbf{x}$ and $\mathbf{y}$ by

$$\mathcal{D}(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})^\top \mathbf{M}(\mathbf{x} - \mathbf{y})} \tag{5}$$

  where $\mathbf{M}$ is some arbitrary *Symmetric Positive Semi-Definite* (SPSD) matrix. We can decompose $\mathbf{M}$ as $\mathbf{M} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\top$ with eigenvalue decomposition, where $\mathbf{U}$ is a matrix collecting all eigenvectors of $\mathbf{M}$, and $\boldsymbol{\Lambda}$ is a diagonal matrix with all eigenvalues of $\mathbf{M}$ on its diagonal line. Let $\mathbf{W} = \mathbf{U}\boldsymbol{\Lambda}^{1/2}$, then we have

$$\begin{aligned}\mathcal{D}(\mathbf{x}, \mathbf{y}) &= \sqrt{(\mathbf{x} - \mathbf{y})^\top \mathbf{W}\mathbf{W}^\top(\mathbf{x} - \mathbf{y})} = \sqrt{(\mathbf{W}^\top(\mathbf{x} - \mathbf{y}))^\top(\mathbf{W}^\top(\mathbf{x} - \mathbf{y}))}\\ &= \sqrt{(\widetilde{\mathbf{x}} - \widetilde{\mathbf{y}})^\top(\widetilde{\mathbf{x}} - \widetilde{\mathbf{y}})}\end{aligned} \tag{6}$$

where $\widetilde{\mathbf{x}} = \mathbf{W}^\top\mathbf{x}$. From the expressions of the above distances we can see that the Euclidean, Cosine, χ^2 and Mahalanobis distances can directly be computed given the data, i.e., no learning procedure is needed. We call these distances *fixed* distances. The goal of this survey is to summarize distance metric *learning* techniques, where we will mainly focus on learning generalized Mahalanobis distance, which wants to learn the best precision matrix from the data such that some optimality criterion is met.

In order to formally define distance metric learning, we present another projection viewpoint of learning distance metrics. By examining the expression of Eq. (6), we can observe that the generalized Mahalanobis distance is equivalent to a Euclidean distance of the data in the projected space transformed by matrix $\mathbf{W}$. Therefore, learning an

[2] http://en.wikipedia.org/wiki/Mahalanobis_distance

[3] http://en.wikipedia.org/wiki/Covariance_matrix

optimal precision matrix $\mathbf{M}$ is equivalent to learn a projection matrix $\mathbf{W}$. In this survey, we define distance metric learning as follows:

Distance Metric Learning *The problem of learning a distance function $\mathcal{D}$ for a pair of data points $\mathbf{x}$ and $\mathbf{y}$ is to learn a mapping function f, such that $f(\mathbf{x})$ and $f(\mathbf{y})$ will be in the Euclidean space and $\mathcal{D}(\mathbf{x}, \mathbf{y}) = \|f(\mathbf{x}) - f(\mathbf{y})\|$, where $\|\cdot\|$ is the ℓ_2 norm.*

With this definition, we can also categorize a distance metric learning algorithm is **linear** or **nonlinear** based on whether the projection is linear or nonlinear.

3 Distance learning algorithms

This section surveys the state-of-the-art distance metric learning algorithms. We categorize these algorithms as unsupervised, supervised or semi-supervised, according to the supervision information they use during the learning process. Throughout the paper, we will use $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n] \in \mathbb{R}^{d \times n}$ to represent the data matrix. $\mathbf{x}_i \in \mathbb{R}^d$ is the i-th data vector. Note that in the following presentations we still use $\mathcal{D}$ to denote the distance metric we want to learn.

Although there are a large number of distance metric learning algorithms, almost all of them are optimizing an objective of the following form under some constraints:

$$\mathcal{J}(\mathcal{D}) = \lambda_1 \mathcal{L}(\mathcal{D}) + \lambda_2 \mathcal{U}(\mathcal{D}) \tag{7}$$

where $\mathcal{L}(\mathcal{D})$ is a *supervised* term involving supervision information, $\mathcal{U}(\mathcal{D})$ is an *unsupervised* term only deals with data characteristics. $\lambda_1 \geqslant 0, \lambda_2 \geqslant 0$ are tradeoff parameters. With this general framework in mind, next we introduce the details of those algorithms and explain how they fit into this general formulation.

3.1 Unsupervised methods

Unsupervised distance metric learning methods do not require any supervision, i.e., they learn an optimal distance metric purely from the data matrix $\mathbf{X}$, such that some geometric or discriminative optimality is achieved. Connecting to Eq. (7), unsupervised methods optimize some $\mathcal{J}(\mathcal{D})$ with $\lambda_1 = 0$. According to the properties of the existing unsupervised approaches, we categorize them as in Table 2, where the definition of linear vs. nonlinear has been clarified in Sect. 2. **Local** versus **global** means that whether an algorithm is derived by optimizing some criteria constructed on the entire data or a local region of the data.

3.1.1 Principal component analysis (PCA)

PCA Jolliffe (2002) is a method aiming at extracting the projection directions from the data on which the maximum variance can be achieved. Let $\bar{\mathbf{X}}$ be the *centralized*

Table 2 Unsupervised distance metric learning algorithms (all the abbreviations can be found in the main text)

	Local	Global
Linear	LPP He and Niyogi (2004)	PCA Jolliffe (2002), UMMP Wang et al. (2011c)
Nonlinear	LE Belkin and Niyogi (2001), LLE Roweis and Saul (2000), Isomap Tenenbaum et al. (2000), SNE Hinton and Roweis (2002), KLPP He and Niyogi (2004)	KPCA Schölkopf and Smola (2002), KUMMP Wang et al. (2011c)

data matrix (i.e., subtracting the mean from $\mathbf{X}$), then the first principle component $\mathbf{w}_1$ can be obtained by

$$\begin{aligned} \mathbf{w}_1 &= argmax_{\|\mathbf{w}\|=1} Var(\mathbf{w}^\top \bar{\mathbf{X}}) \\ &= argmax_{\|\mathbf{w}\|=1} \frac{1}{n-1} \mathbf{w}^\top \bar{\mathbf{X}} \bar{\mathbf{X}}^\top \mathbf{w} \end{aligned} \tag{8}$$

Given the first $k-1$ principal components, the k-th principal component can be found by subtracting the effect of the first $k-1$ principal components from $\bar{\mathbf{X}}$:

$$\widehat{\mathbf{X}}_{k-1} = \bar{\mathbf{X}} - \sum_{i=1}^{k-1} \mathbf{w}_i \mathbf{w}_i^\top \widehat{\mathbf{X}}_i \tag{9}$$

By using $\widehat{\mathbf{X}}_{k-1}$ as the new data set, we can find the k-th principal component as

$$\mathbf{w}_k = argmax_{\|\mathbf{w}\|=1} \mathbf{w}^\top \widehat{\mathbf{X}}_{k-1} \widehat{\mathbf{X}}_{k-1}^\top \mathbf{w} \tag{10}$$

Overall, we want to find matrix $\mathbf{W}$ with the follow property:

$$\begin{aligned} \max_{\mathbf{W}} \;& tr\left(\mathbf{W}^\top \bar{\mathbf{X}} \bar{\mathbf{X}}^\top \mathbf{W}\right) \\ s.t. \;& \mathbf{W}^\top \mathbf{W} = \mathbf{I} \end{aligned} \tag{11}$$

We can see that the projection directions of PCA are obtained by successively seeking for the direction that maximizes the entire data variance. Because $tr\left(\mathbf{W}^\top \bar{\mathbf{X}} \bar{\mathbf{X}}^\top \mathbf{W}\right)$ measures the sum of variances on all the projection direction. The solution to $\mathbf{W}$ can be obtained through eigen-decomposition of $\bar{\mathbf{X}}\bar{\mathbf{X}}^\top$[4]. PCA is a global and linear technique. The learned distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ is the Euclidean distance between $\mathbf{W}^\top \bar{\mathbf{x}}_i$ and $\mathbf{W}^\top \bar{\mathbf{x}}_j$. Also PCA can be easily extended to out-of-sample data as the linear projection $\mathbf{W}$ is learned explicitly.

[4] http://en.wikipedia.org/wiki/Eigendecomposition_of_a_matrix.

3.1.2 Nonlinear PCA

One limitation of PCA is that it is a linear method. To enhance its capability of handling nonlinearly distributed data, we can use a *Kernel Trick* Schölkopf and Smola (2002), which is a common technique used in machine learning and data mining that aims at transforming the nonlinear problem in the original data space into a linear problem in some mapped feature space. Suppose $\phi : \mathbb{R}^d \to \mathcal{F}$ is such mapping. Generally $\mathcal{F}$ is a *Reproducing Kernel Hilbert Space* (RKHS) , thus we can rewrite the mapping as $\phi : \mathbf{x} \to k(\cdot, \mathbf{x})$, where $k(\cdot, \mathbf{x})$ is a function in RKHS satisfying

- $\langle k(\cdot, \mathbf{x}), f\rangle = f(\mathbf{x})$
- $\langle k(\cdot, \mathbf{x}), k(\cdot, \mathbf{y})\rangle = k(\mathbf{x}, \mathbf{y})$

where f is also a function in the same RKHS. With the *Representer Theorem* Schölkopf and Smola (2002), we can write any function $f \in \mathcal{F}$ as

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_i, \mathbf{x}) \tag{12}$$

Applying the kernel trick on those traditional methods means that we perform those analysis in the kernel space instead of the original space. For example, *Kernel Principle Component Analysis* (KPCA) performs PCA in the RKHS that those data points are mapped into. Specifically, in RKHS, the data covariance matrix becomes

$$\mathbf{C} = \frac{1}{n-1}\sum_{i=1}^{n}\left(\phi(\mathbf{x}_i) - \bar{\phi}(\mathbf{x})\right)\left(\phi(\mathbf{x}_i) - \bar{\phi}(\mathbf{x})\right)^\top = \frac{1}{n-1}\bar{\boldsymbol{\Phi}}\bar{\boldsymbol{\Phi}}^\top \tag{13}$$

where $\bar{\Phi}(\mathbf{x}) = \sum_{i=1}^{n}\phi(\mathbf{x}_i)/n$ is the mean in the feature space, and

$$\bar{\boldsymbol{\Phi}} = \left[\phi(\mathbf{x}_1) - \bar{\phi}(\mathbf{x}), \phi(\mathbf{x}_2) - \bar{\phi}(\mathbf{x}), \cdots, \phi(\mathbf{x}_n) - \bar{\phi}(\mathbf{x})\right] \tag{14}$$

The goal of KPCA is to perform eigenvalue decomposition on $\mathbf{C}$ and get the eigenvectors as

$$\mathbf{C}\mathbf{v} = \frac{1}{n-1}\bar{\boldsymbol{\Phi}}\bar{\boldsymbol{\Phi}}^\top\mathbf{v} = \lambda\mathbf{v} \tag{15}$$

With the representer theorem, we have $\mathbf{v} = \bar{\boldsymbol{\Phi}}\boldsymbol{\alpha}$. Multiplying $\bar{\boldsymbol{\Phi}}$ on both sides of Eq. (15) and applying the representer theorem, we can get

$$\frac{1}{n-1}\bar{\boldsymbol{\Phi}}^\top\bar{\boldsymbol{\Phi}}\bar{\boldsymbol{\Phi}}^\top\bar{\boldsymbol{\Phi}}\boldsymbol{\alpha} = \lambda\bar{\boldsymbol{\Phi}}^\top\bar{\boldsymbol{\Phi}}\boldsymbol{\alpha} \Rightarrow \frac{1}{n-1}\bar{\mathbf{K}}^2\boldsymbol{\alpha} = \lambda\bar{\mathbf{K}}\boldsymbol{\alpha} \Rightarrow \frac{1}{n-1}\bar{\mathbf{K}}\boldsymbol{\alpha} = \lambda\boldsymbol{\alpha} \tag{16}$$

where $\bar{\mathbf{K}}$ is the centralized kernel matrix with its (i, j)-th entry

$$\bar{\mathbf{K}}_{ij} = \langle\phi(\mathbf{x}_i) - \bar{\phi}(\mathbf{x}), \phi(\mathbf{x}_j) - \bar{\phi}(\mathbf{x})\rangle = \langle\bar{\phi}(\mathbf{x}_i), \bar{\phi}(\mathbf{x}_j)\rangle \tag{17}$$

We can get the projection of $\bar{\phi}(\mathbf{x}_i)$ on $\mathbf{v}$ by

$$\bar{\phi}(\mathbf{x}_i)^\top \mathbf{v} = \sum_{u=1}^{n} \alpha_u \langle \bar{\phi}(\mathbf{x}_i), \bar{\phi}(\mathbf{x}_u) \rangle = \bar{\mathbf{K}}_{i \cdot} \boldsymbol{\alpha} \tag{18}$$

After we get the projections, the pairwise distance metric can be evaluated by the Euclidean distance on those projected coordinates.

Therefore, KPCA can be viewed as performing PCA in the feature space $\mathcal{F}$, so it is a nonlinear, global method. The learned distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ is the Euclidean distance between $\bar{\phi}(\mathbf{x}_i)^\top \mathbf{v}$ and $\bar{\phi}(\mathbf{x}_j)^\top \mathbf{v}$. The computational technique involved in KPCA is also eigenvalue decomposition, and KPCA can be easily extended to out-of-sample data with Eq. (18).

3.1.3 Unsupervised maximum margin projections (UMMP)

Both PCA and KPCA aim to find a projection space under which the total data variance is maximized. Another criterion that is popular in unsupervised learning is that the scatterness of the data clusters should be maximized. Along this direction, *Unsupervised Maximum Margin Projections* (UMMP) Wang et al. (2011c) is one representative method that aims to maximize the aggregated pairwise cluster margins. UMMP can be viewed as an unsupervised extension of the *Support Vector Machine* (SVM) algorithm Vapnik (1995). Suppose data points $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n]$ come from two classes, the label of $\mathbf{x}_i$ is $l_i \in \{-1, +1\}$. Then the goal of SVM is to find the maximum-margin hyperplane that divides the points with $l_i = 1$ from those with $l_i = -1$ (thus it is a supervised method). Any hyperplane can be written as a point $\mathbf{x}$ satisfying

$$\mathbf{w}^\top \mathbf{x} - b = 0 \tag{19}$$

where $b/\|\mathbf{w}\|$ corresponds to the distance of the hyperplane from the origin. SVM aims to choose the $\mathbf{w}$ and b to maximize the distance between the parallel hyperplanes that are as far apart as possible while still separating the data, which is usually referred to as the margin of the two classes. These hyperplanes can be described by the equations

$$\mathbf{w}^\top \mathbf{x} - b = 1 \quad \text{or} \quad \mathbf{w}^\top \mathbf{x} - b = -1 \tag{20}$$

The distance between the two parallel hyperplane is $2/\|\mathbf{w}\|$. Then if the data from two classes are clearly separated, then the goal of SVM is to solve the following optimization problem to find the hyperplane that maximizes the margin between two classes

$$\begin{aligned} &\min_{\mathbf{w},b} \ \frac{1}{2}\|\mathbf{w}\|^2 \\ &\quad s.t. \ \ l_i(\mathbf{w}^\top \mathbf{x}_i - b) \geqslant 1 \ \ (\forall i = 1, 2, \cdots, n) \end{aligned} \tag{21}$$

However in reality the two classes may not be perfectly separable i.e., there might be some overlapping between them. Then we need *soft margin* SVM, which aims at solving

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \ \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i \qquad (22)$$
$$s.t. \ \ l_i(\mathbf{w}^\top \mathbf{x}_i - b) \geqslant 1 - \xi_i \ \ (\forall i = 1, 2, \cdots, n)$$

where $\boldsymbol{\xi} = \{\xi_i\} \geqslant 0$ are slack variables used to penalize the margin on the overlapping region. *Unsupervised Maximum Margin Projections* (UMMP) extends SVM by seeking for k projection hyperplanes $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_k]$ via solving an optimization problem. At the same time, since the label is not given, UMMP also tries to learn an optimal set of label such that data points are optimally separated based on the learned class labels. The optimization problem is formulated as

$$\begin{aligned} \min_{\mathbf{l}\in\{-1,+1\}^n} \ \min_{\mathbf{W},\mathbf{b},\boldsymbol{\xi}_r \geq 0} \ & \frac{1}{2}\sum_{r=1}^{d}\|\mathbf{w}_r\|^2 + \frac{C}{n}\sum_{r=1}^{d}\sum_{i=1}^{n}\xi_{ri} \qquad (23) \\ s.t. \ & \forall i = 1, \ldots, n, \quad r = 1, \ldots, d \\ & l_i\left(\left(\mathbf{w}^r\right)^T \mathbf{x}_i + b\right) \geq 1 - \xi_{ri}, \\ & \mathbf{W}^T\mathbf{W} = \mathbf{I} \end{aligned}$$

where $\boldsymbol{\xi}_r = \{\xi_{ri}\}_{i=1}^{n}$ $(r = 1, 2, \cdots, k)$ with $\xi_{ri} > 0$ are the slack variables for penalizing soft margins and $C > 0$ is a tradeoff parameter. $\sum_{i=1}^{n}\xi_{ri}$ is divided by n to better capture how C scales with the dataset size. Empirically, C can be tuned in a fix range without worrying about the dataset size n. Intuitively what UMMP does is to find k orthogonal projection hyperplanes such that on each projection direction the two data clusters are separated as well as possible (in terms of maximizing the soft margin). After some relaxations, Wang et al. (2011c) proposed a cutting-plane based approach for solving the problem.

As UMMP aims to separate the whole clusters, it is a global and linear approach. One can use the kernel trick to extend UMMP for handling data that are not linearly separable. The learned distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ is the Euclidean distance between $\mathbf{W}^\top\mathbf{x}_i$ and $\mathbf{W}^\top\mathbf{x}_j$. Similar to PCA, UMMP also learns the explicit mapping function $\mathbf{W}$, thus it can be easily extended to out-of-sample data. The computational technique involved in UMMP is quadratic programming. One can refer to Wang et al. (2011c) for more details on Kernel UMMP.

3.1.4 Locality preserving projections (LPP)

PCA and UMMP find the projection directions on which the data are optimally spread. More specifically, PCA seeks for projection directions on which the variance of the entire data is maximized, while UMMP looks for the directions on which the data clusters are best separated. Another popular criterion for learning the projection directions is preserving the geometry of the data in the original space after projection. One representative example is *locality preserving projection (LPP)* He and Niyogi (2004).

LPP aims at finding a projection matrix $\mathbf{W}$, which preserves the localities of the data in the original space. Here the localities of a data set is captured by the pairwise data similarity $\{\omega_{ij}\}_{i,j=1}^{n}$ within a neighborhood. This neighborhood is usually calculated using a Gaussian function as

$$\omega_{ij} = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right) \text{ (if } \mathbf{x}_i \in \mathcal{N}_j \text{ or } \mathbf{x}_j \in \mathcal{N}_i\text{)} \tag{24}$$

Here $\mathcal{N}_i$ or $\mathcal{N}_j$ represents the neighborhood around $\mathbf{x}_i$ or $\mathbf{x}_j$. The implication is that we only need to preserve distances within local neighborhoods.

The goal of LPP is to solve the following optimization problem

$$\begin{aligned} \min_{\mathbf{W}} & \sum_{ij:\mathbf{x}_i \in \mathcal{N}_j \text{ or } \mathbf{x}_j \in \mathcal{N}_i} \|\mathbf{W}^\top \mathbf{x}_i - \mathbf{W}^\top \mathbf{x}_j\|^2 \omega_{ij} = tr\left(\mathbf{W}^\top \mathbf{XLX}^\top \mathbf{W}\right) \\ s.t. & \ \ \mathbf{W}^\top \mathbf{XDX}^\top \mathbf{W} = \mathbf{I} \end{aligned} \tag{25}$$

where $\mathbf{D} \in \mathbb{R}^{n \times n}$ is the diagonal matrix with $\mathbf{D}_{ii} = \sum_j \omega_{ij}$. $\mathbf{L} = \mathbf{D} - \boldsymbol{\Omega}$ is the Laplacian matrix with $\boldsymbol{\Omega}_{ij} = \omega_{ij}$ if $\mathbf{x}_i \in \mathcal{N}_j$ or $\mathbf{x}_j \in \mathcal{N}_i$, and $\boldsymbol{\Omega}_{ij} = 0$ otherwise. The optimal solution to problem (25) can be obtained by doing generalized eigenvalue decomposition on ($\mathbf{XLX}$, $\mathbf{XDX}$) as

$$\mathbf{XLX}^\top \mathbf{w} = \lambda \mathbf{XDX}^\top \mathbf{w} \tag{26}$$

and the optimal projection directions can be obtained by concatenating the k eigenvectors whose corresponding eigenvalues are the smallest ones. Therefore LPP is a linear and local method, because it only wants to preserve the local data geometries. The learned distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ is the Euclidean distance between $\mathbf{W}^\top \mathbf{x}_i$ and $\mathbf{W}^\top \mathbf{x}_j$. The computational technique involved in LPP is eigenvalue decomposition, and it can easily be extended to out-of-sample data with $\mathbf{W}$.

We can also make use of the kernel trick to make LPP nonlinear, in which case we need to solve the generalized eigenvalue decomposition problem

$$\boldsymbol{\Phi} \mathbf{L} \boldsymbol{\Phi}^\top \mathbf{v} = \lambda \boldsymbol{\Phi} \mathbf{D} \boldsymbol{\Phi}^\top \mathbf{v} \Rightarrow \boldsymbol{\Phi} \mathbf{L} \boldsymbol{\Phi}^\top \boldsymbol{\Phi} \boldsymbol{\alpha} = \lambda \boldsymbol{\Phi} \mathbf{D} \boldsymbol{\Phi}^\top \boldsymbol{\Phi} \boldsymbol{\alpha} \tag{27}$$

where $\boldsymbol{\Phi}$ is the data matrix in feature space after kernel mapping, and with the representer theorem, we have $\mathbf{v} = \boldsymbol{\Phi} \boldsymbol{\alpha}$. Therefore

$$\mathbf{KLK}\boldsymbol{\alpha} = \lambda \mathbf{KDK}\alpha \Rightarrow \mathbf{Ly} = \lambda \mathbf{Dy} \tag{28}$$

where $\mathbf{y} = \mathbf{K}\alpha$ represents the data set embeddings after KLPP. Thus KLPP is a nonlinear and local method.

3.1.5 Laplacian embedding (LE)

Actually, before LPP appears, there is another technique called LE Belkin and Niyogi (2001) also focusing on seeking for embeddings that preserve the data localities. Here locality is defined in the same way as in Eq. (24). Suppose we want to embed the data into a one-dimensional space with embedding coordinates $\mathbf{y} = [y_1, y_2, \cdots, y_n]^\top$, then the goal of LE is to get the $\mathbf{y}$ by solving the following optimization problem.

$$\begin{aligned} \min_{\mathbf{y}} \ & \sum_{i=1}^{n} (y_i - y_j)^2 \omega_{ij} = \mathbf{y}^\top \mathbf{L} \mathbf{y} \\ s.t. \ & \mathbf{y}^\top \mathbf{D} \mathbf{y} = 1 \end{aligned} \tag{29}$$

where $\mathbf{L}$ is the Laplacian matrix similar to LPP. Since LE directly learns the data embedding coordinates without any explicit mappings, LE is a nonlinear and local method. The learned distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ is the Euclidean distance between $\mathbf{y}_i$ and $\mathbf{y}_j$. The computational technique involved in LE is eigenvalue decomposition. It is not that straightforward for LE to extend to out-of-sample data, as it learns the embedded data coordinates directly without obtaining any explicit mappings.

3.1.6 Locally linear embedding (LLE) [Roweis and Saul (2000)]

LLE (Roweis and Saul 2000) is another method aiming at obtaining locality preserving embeddings. The difference between LLE and LE is the way they capture the data localities. LLE is based on the *Linear Neighborhood* assumption, which assumes that each data point $\mathbf{x}_i$ $(i = 1, 2, \cdots, n)$ can be *linearly* reconstructed from its neighborhood $\mathcal{N}_i$ $(i = 1, 2, \cdots, n)$, i.e.,

$$\begin{aligned} \min_{\omega_{ij}} \ & \sum_i \left\| \mathbf{x}_i - \sum_{\mathbf{x}_j \in \mathcal{N}_i} \omega_{ij} \mathbf{x}_j \right\|^2 \\ s.t. \ & \sum_j \omega_{ij} = 1 \ (\forall i = 1, 2, \cdots, n) \end{aligned} \tag{30}$$

In the second step, LLE aims to recover the low dimensional embeddings that preserve these local relationships by solving the following optimization problem.

$$\min_{\{\mathbf{y}_i\}_{i=1}^{n}} \ \sum_i \left\| \mathbf{y}_i - \sum_{\mathbf{x}_j \in \mathcal{N}_i} \omega_{ij} \mathbf{y}_j \right\|^2 \tag{31}$$

$$s.t. \ \sum_{i=1}^{n} \mathbf{y}_i = 0, \quad \sum_{i=1}^{n} \mathbf{y}_i \mathbf{y}_i = n\mathbf{I} \tag{32}$$

LLE is also a local and nonlinear approach. Like LE, the learned distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ is the Euclidean distance between $\mathbf{y}_i$ and $\mathbf{y}_j$. The computational techniques involved in LLE include both quadratic programming and eigenvalue decomposition. It is not that straightforward for LLE to extend to out-of-sample data, as it learns the embedded data coordinates directly without obtaining any explicit mappings.

3.1.7 Isometric feature mapping (Isomap)

Isomap (Tenenbaum et al. 2000) is another representative low-dimensional embedding method, where *geodesic distances* on a weighted graph are incorporated with the classical scaling [metric *Multidimensional Scaling* (*MDS*) Cox and Cox (2000)]. The major difference between Isomap and LE, LLE is the way they capture pairwise data similarity. In Isomap, rather than similarity, the pairwise data distances (which can be viewed as *dissimilarity*) are first calculated, then classic MDS is used to obtain the embedding coordinates of the data points such that the pairwise data distances are preserved as well as possible.

Here the geodesic distance between pairwise data points is measured in the following way: first a connected neighborhood graph is constructed on the data set, the graph can either be weighted or unweighted; then the pairwise geodesic distance is the shortest path on such graph. Such calculation can be viewed as the discrete approximation of the real pairwise geodesic distance on data manifold. Therefore Isomap is a global and nonlinear approach. The learned distance is measured by the Euclidean distance on the projected low-dimensional space. The computational techniques involved in Isomap is eigenvalue decomposition. Similar to LE and LLE, it is not that straightforward for Isomap to extend to out-of-sample data, as it learns the embedded data coordinates directly without obtaining any explicit mappings.

3.1.8 Stochastic neighbor embedding (SNE)

Till now all the unsupervised approaches aim to optimize some geometry based criteria, which can be either separation based or locality preserving based. There are also methods obtaining those data embeddings by optimizing an information theoretic criterion. SNE (Hinton and Roweis 2002) is such an algorithm.

Let the pairwise data dissimilarities be

$$d_{ij} = \frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2} \tag{33}$$

where σ is a empirically determined scaling factor. The probability that $\mathbf{x}_i$ picks $\mathbf{x}_j$ as its neighbor is

$$p_{ij} = \frac{\exp\left(-d_{ij}^2\right)}{\sum_{k \neq i} \exp\left(-d_{ik}^2\right)} \tag{34}$$

Similarly, in the embedded space, the probability that $\mathbf{x}_i$ picks $\mathbf{x}_j$ as its neighbor is

$$q_{ij} = \frac{\exp\left(- \|\mathbf{y}_i - \mathbf{y}_j\|^2\right)}{\sum_{k \neq i} \exp\left(- \|\mathbf{y}_i - \mathbf{y}_k\|^2\right)} \tag{35}$$

SNE aims at minimizing

$$\mathcal{J} = \sum_{ij} p_{ij} \log \frac{p_{ij}}{q_{ij}} = \sum_{i} KL(P_i \| Q_i) \tag{36}$$

Therefore, SNE is an embedding method aiming at preserving the neighborhood distribution around each data point, and it will also get the embedding coordinates as well. From the expression of p_{ij} in Eq. (34), we can observe that when $(\mathbf{x}_i, \mathbf{x}_k)$ are very far apart, $\exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2) \to 0$, i.e., p_{ij} will emphasize more on the impact of the data pairs whose Euclidean distance is small. For minimizing Eq. (36) Hinton and Roweis (2002) adopted a gradient descent based approach. To achieve a better local optimal, the authors in Hinton and Roweis (2002) also inject random jitter that decreases with time into the gradient descent procedure. SNE is a local and nonlinear method. The learned distance is measured by the Euclidean distance on the projected low-dimensional space. As SNE directly learns the data embedding coordinates, it is not easy for SNE to extend to out-of-sample data.

Summary Till now we have introduced different unsupervised distance metric learning approaches. All of them formulate the learning procedure as some optimization problem, where the objective can either be geometry based or information theoretic. For all these methods except simple PCA, there are some free parameters need to be pre-specified, such as the kernel parameters for kernel methods, neighborhood size or scaling parameter. All these methods involve expensive optimization procedures, such as eigenvalue decomposition or semi-definite programming [although SNE employs gradient descent, adding the jittered noise will still make it slow Hinton and Roweis (2002)]. Therefore, scalability is still an open issue for applying those algorithms on large datasets.

In terms of extensibility to out-of-sample data (i.e., those data not in the training set), linear and kernel methods have some advandages as they can naturally get learn the projection mappings straightforwardly. For the methods getting embedding coordinates directly (e.g., Isomap, LLE, LE), we need additional efforts to make them extensible Bengio et al. (2004). For real world use, as all these approaches are unsupervised, they only explore the characteristics from data. Thus most of these approaches can be used to explore the data manifold. Only one exception is UMMP, which is clustering driven, i.e., it seeks for the projection space where the data clusters are maximally separated.

3.2 Supervised algorithms

In this section we survey supervised distance metric learning algorithms, which learn distance metrics on both data points and their labels. Connecting to Eq. (7), supervised approaches optimize $\mathcal{J}(\mathcal{D})$ with $\lambda_2 = 0$. Similar as in unsupervised approaches, we also categorize those supervised methodologies into different types according to their characteristics. The details can be found in Table 3.

In this survey, we will review the algorithms with two types of supervision: (1) *Labels*, which indicate the class information each training data point belongs to. The

Table 3 Supervised distance metric learning algorithms (all the abbreviations can be found in the main text)

	Local	Global
Linear	NCA Goldberger et al. (2004), ANMM Wang and Zhang (2007), LMNN Weinberger et al. (2005)	LDA Fukunaga (1990), LSI Xing et al. (2002), ITML Davis et al. (2007), MMDA Kocsor et al. (2004), RCA Shental et al. (2002)
Nonlinear	KANMM Wang and Zhang (2007), KLMNN Weinberger et al. (2005)	KLDA Mika et al. (1999), KMMDA Kocsor et al. (2004), KRCA Tsang et al. (2005)

assumption is that distance between data points with the same label should be closer to distance between data points from different labels. (2) *Pairwise constraints*, indicate whether a pair of data points should belong to the same class (*must-links*) or not (*cannot-links*).

3.2.1 Linear discriminant analysis (LDA)

LDA (Fukunaga 1990) is one of the most popular supervised linear embedding methods. It seeks for the projection directions under which the data from different classes are well separated. More concretely, supposing that the data set belongs to C different classes, LDA defines the *compactness matrix* and *scatterness matrix* as

$$\boldsymbol{\Sigma}_{\mathcal{C}} = \frac{1}{C}\sum_{c}\frac{1}{n_c}\sum_{\mathbf{x}_i\in c}(\mathbf{x}_i-\bar{\mathbf{x}}_c)(\mathbf{x}_i-\bar{\mathbf{x}}_c)^\top \tag{37}$$

$$\boldsymbol{\Sigma}_{\mathcal{S}} = \frac{1}{C}\sum_{c}(\bar{\mathbf{x}}_c-\bar{\mathbf{x}})(\bar{\mathbf{x}}_c-\bar{\mathbf{x}})^\top \tag{38}$$

The goal of LDA is to find a $\mathbf{W}$ which can be obtained by solving the following optimization problem

$$\min_{\mathbf{W}^\top\mathbf{W}=\mathbf{I}} \frac{tr(\mathbf{W}^\top\boldsymbol{\Sigma}_{\mathcal{C}}\mathbf{W})}{tr(\mathbf{W}^\top\boldsymbol{\Sigma}_{\mathcal{S}}\mathbf{W})} \tag{39}$$

By expanding the numerator and denominator of the above expression, we can observe that the numerator corresponds to the sum of distances between each data point to its class center after projection, and the denominator represents the sum of distances between every class center to the entire data center after projection. Therefore, minimizing the objective will maximize the between-class scatterness while minimize the within-class scatterness after projection. Solving problem (39) is hard, some researchers (Guo et al. 2003; Jia et al. 2009) have done research on this topic. LDA is a linear and global method. The learned distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ is the Euclidean distance between $\mathbf{W}^\top\mathbf{x}_i$ and $\mathbf{W}^\top\mathbf{x}_j$. It is easy to extend LDA to out-of-sample data as the explicit mapping $\mathbf{W}$ is learned, and the computational technique involved is eigenvalue decomposition.

Kernelization of LDA Similar to the case of PCA, we can extend LDA to the nonlinear case via the kernel trick, which is called *Kernel Discriminant Analysis* (*KDA*) (Mika et al. 1999). After mapping the data into the feature space using ϕ, we can compute the compactness and scatterness matrices as

$$\boldsymbol{\Sigma}_{\mathcal{C}}^{\phi} = \frac{1}{C}\sum_{c}\frac{1}{n_c}\sum_{\mathbf{x}_i \in c}(\phi(\mathbf{x}_i) - \bar{\boldsymbol{\phi}}_c)(\phi(\mathbf{x}_i) - \bar{\boldsymbol{\phi}}_c)^{\top} \tag{40}$$

$$\boldsymbol{\Sigma}_{\mathcal{S}} = \frac{1}{C}\sum_{c}(\bar{\boldsymbol{\phi}}_c - \bar{\boldsymbol{\phi}})(\bar{\boldsymbol{\phi}}_c - \bar{\boldsymbol{\phi}})^{\top} \tag{41}$$

Suppose the projection matrix we want to get is $\mathbf{W}^{\phi}$ in the feature space, then with the representer theorem

$$\mathbf{W}^{\phi} = \boldsymbol{\Phi}\boldsymbol{\alpha} \tag{42}$$

where $\boldsymbol{\Phi} = [\phi(\mathbf{x}_1), \phi(\mathbf{x}_2), \cdots, \phi(\mathbf{x}_n)]$ and $\boldsymbol{\alpha}$ is the coefficient vector over all $\phi(\mathbf{x}_i)$ for $1 \leq i \leq n$. We define $\mathbf{K} = \boldsymbol{\Phi}^{\top}\boldsymbol{\Phi}$ as the kernel matrix.

Then

$$\begin{aligned}(\mathbf{W}^{\phi})^{\top}\boldsymbol{\Sigma}_{\mathcal{C}}^{\phi}\mathbf{W}^{\phi} &= \boldsymbol{\alpha}^{\top}\left[\frac{1}{C}\sum_{c=1}^{C}\frac{1}{n_c}\sum_{\mathbf{x}_i \in c}\boldsymbol{\Phi}^{\top}(\phi(\mathbf{x}_i) - \bar{\boldsymbol{\phi}}_c)(\phi(\mathbf{x}_i) - \bar{\boldsymbol{\phi}}_c)^{\top}\boldsymbol{\Phi}\right]\boldsymbol{\alpha} \\ &= \boldsymbol{\alpha}^{\top}\left[\frac{1}{C}\sum_{c=1}^{C}\frac{1}{n_c}\sum_{\mathbf{x}_i \in c}(\mathbf{K}_{\cdot i} - \bar{\mathbf{K}}_{\cdot c})(\mathbf{K}_{\cdot i} - \bar{\mathbf{K}}_{\cdot c})^{\top}\right]\boldsymbol{\alpha} \\ &= \boldsymbol{\alpha}^{\top}\mathbf{M}_{\mathcal{C}}\boldsymbol{\alpha}\end{aligned} \tag{43}$$

where $\mathbf{K}_{\cdot i}$ $=$ and $\bar{\mathbf{K}}_{\cdot c} = \frac{1}{n_c}\sum_{\mathbf{x}_i \in c}\mathbf{K}_{\cdot i}$, and $\mathbf{M}_{\mathcal{C}} = \frac{1}{C}\sum_{c=1}^{C}\frac{1}{n_c}\sum_{\mathbf{x}_i \in c}(\mathbf{K}_{\cdot i} - \bar{\mathbf{K}}_{\cdot c})(\mathbf{K}_{\cdot i} - \bar{\mathbf{K}}_{\cdot c})^{\top}$.

$$\begin{aligned}(\mathbf{W}^{\phi})^{\top}\boldsymbol{\Sigma}_{\mathcal{S}}^{\phi}\mathbf{W}^{\phi} &= \boldsymbol{\alpha}^{\top}\left[\frac{1}{C}\sum_{c}(\bar{\mathbf{K}}_{\cdot c} - \bar{\mathbf{K}}_{\cdot *})(\bar{\mathbf{K}}_{\cdot c} - \bar{\mathbf{K}}_{\cdot *})^{\top}\right]\boldsymbol{\alpha} \\ &= \boldsymbol{\alpha}^{\top}\mathbf{M}_{\mathcal{S}}\boldsymbol{\alpha}\end{aligned} \tag{44}$$

where $\mathbf{K}_{\cdot *} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{K}_{\cdot i}$, and $\mathbf{M}_{\mathcal{S}} = \frac{1}{C}\sum_{c}(\bar{\mathbf{K}}_{\cdot c} - \bar{\mathbf{K}}_{\cdot *})(\bar{\mathbf{K}}_{\cdot c} - \bar{\mathbf{K}}_{\cdot *})^{\top}$. Therefore we can get $\boldsymbol{\alpha}$ by solving

$$\min_{\boldsymbol{\alpha}^{\top}\boldsymbol{\alpha}=\mathbf{I}}\frac{tr\left(\boldsymbol{\alpha}^{\top}\mathbf{M}_{\mathcal{C}}\boldsymbol{\alpha}\right)}{tr\left(\boldsymbol{\alpha}^{\top}\mathbf{M}_{\mathcal{S}}\boldsymbol{\alpha}\right)} \tag{45}$$

3.2.2 Margin maximizing discriminant analysis (MMDA)

MMDA (Kocsor et al. 2004) can be viewed as a supervised version of UMMP approach in Sect. 3.1.3. This is the supervised version of UMMP, where we need to solve the following optimization problem

$$\begin{aligned} \min_{\mathbf{W},\mathbf{b},\xi_r \geq 0} \quad & \frac{1}{2}\sum_{r=1}^{d}\|\mathbf{w}_r\|^2 + \frac{C}{n}\sum_{r=1}^{d}\sum_{i=1}^{n}\xi_{ri} \\ s.t. \quad & \forall i = 1,\ldots,n, \quad r = 1,\ldots,d \\ & l_i\left(\left(\mathbf{w}^r\right)^T \mathbf{x}_i + b\right) \geq 1 - \xi_{ri}, \\ & \mathbf{W}^T\mathbf{W} = \mathbf{I} \end{aligned} \tag{46}$$

Comparing problem (46) with problem (23), we can see that the only difference between MMDA and UMMP is MMDA does not need to solve the data cluster labels because they are available in supervised setting. Therefore MMDA is a global and linear approach. One can also apply kernel trick to make it nonlinear, the details can be found in Kocsor et al. (2004). The learned distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ is just the Euclidean distance between $\mathbf{W}^\top\mathbf{x}_i$ and $\mathbf{W}^\top\mathbf{x}_j$. It is easy to extend MMDA to out-of-sample data as the explicit mapping $\mathbf{W}$ is learned, and the computational technique involved is quadratic programming.

3.2.3 Learning with side information (LSI)

Both LDA and MMDA use data labels as the supervision information. As we introduced at the beginning of Sect. 3.2, another type of supervision information we considered is pairwise constraints. The data label information is more strict in the sense that we can convert data labels into pairwise constraints, but not vise versa.

One of the earliest research that making use of pairwise constraints for learning a distance metric is the LSI approach Xing et al. (2002). We denote the set of *must-link* constraints as $\mathcal{M}$ and the set of *cannot-link* constraints as $\mathcal{C}$, then the goal of LSI is to solve the following optimization problem

$$\begin{aligned} \max_{\mathbf{M}} \quad & \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{C}} (\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{M}(\mathbf{x}_i - \mathbf{x}_j) \\ s.t. \quad & \sum_{(\mathbf{x}_u,\mathbf{x}_v)\in\mathcal{M}} (\mathbf{x}_u - \mathbf{x}_v)^\top \mathbf{M}(\mathbf{x}_u - \mathbf{x}_v) \leqslant 1 \\ & \mathbf{M} \succeq 0 \end{aligned} \tag{47}$$

This is a quadratic optimization problem and Xing et al. (2002) proposed an iterative projected gradient ascent method to solve it. As $\mathbf{M}$ is positive semi-definite, we can always factorize it as $\mathbf{M} = \mathbf{W}\mathbf{W}^\top$. Thus LSI is a global and linear approach. The learned distance formulation is exactly the general Mahalanobis distance with precision matrix $\mathbf{M}$. It is easy to extend LSI to out-of-sample data as the explicit mapping $\mathbf{W}$ is learned, and the computational technique involved is eigenvalue decomposition.

3.2.4 Relevant component analysis (RCA)

RCA (Shental et al. 2002; Bar-Hillel et al. 2005) is another representative distance metric learning algorithm utilizing pairwise data constraints. The goal of RCA is to find a transformation that amplifies *relevant variability* and suppresses *irrelevant variability*. Here *variability* is just sample variance. We consider that data variability is correlated with a specific task if the removal of this variability from the data deteriorates (on average) the results of clustering or retrieval. Variability is irrelevant if it is maintained in the data but not correlated with the specific task Shental et al. (2002). We also define small clusters called *chunklets*, which are connected components derived by all the must-links. The specific steps involved in RCA include:

- Construct *chunklets* according to equivalence (must-link) constraints, such that the data in each chunklet are connected by must-link constraints pairwisely.
- Assume a total of p points in k chunklets, where chunklet j consists of points $\{\mathbf{x}_{ji}\}_{i=1}^{n_j}$ and its mean is $\bar{\mathbf{m}}_j$. RCA computes the following weighted within-chunklet covariance matrix:

$$\mathbf{C} = \frac{1}{p}\sum_{j=1}^{k}\sum_{i=1}^{n_j}(\mathbf{x}_{ji} - \bar{\mathbf{m}}_j)(\mathbf{x}_{ji} - \bar{\mathbf{m}}_j)^\top \tag{48}$$

- Compute the whitening transformation $\mathbf{W} = C^{1/2}$, and apply it to the original data points: $\tilde{\mathbf{x}} = \mathbf{W}\mathbf{x}$. Alternatively, use the inverse of $\mathbf{C}$ as the precision matrix of a generalized Mahalanobis distance.

Therefore, RCA is a global, linear approach. It is easy to extend RCA to out-of-sample data as the explicit mapping $\mathbf{W}$ is learned, and the computational technique involved is eigenvalue decomposition.

3.2.5 Information theoretic metric learning (ITML)

Information theoretic objective is one mechanism to develop a supervised distance metric. ITML (Davis et al. 2007) is one such representative algorithm. Suppose we have an initial generalized Mahalanobis distance parameterized by precision matrix $\mathbf{M}_0$, a set $\mathcal{M}$ of must-link constraints and a set $\mathcal{C}$ of cannot-link constraints. ITML solves the following optimization problem

$$\begin{aligned} \min\nolimits_{\mathbf{M}\succeq 0} \quad & d_{logdet}(\mathbf{M}, \mathbf{M}_0) \\ s.t. \quad & (\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{M}(\mathbf{x}_i - \mathbf{x}_j) \geqslant l, \ (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{C} \\ & (\mathbf{x}_u - \mathbf{x}_v)^\top \mathbf{M}(\mathbf{x}_u - \mathbf{x}_v) \leqslant u, \ (\mathbf{x}_u, \mathbf{x}_v) \in \mathcal{M} \end{aligned} \tag{49}$$

where

$$d_{logdet}(\mathbf{M}, \mathbf{M}_0) = tr(\mathbf{M}\mathbf{M}_0^{-1}) - \log det(\mathbf{M}\mathbf{M}_0^{-1}) - n \tag{50}$$

where d_{logdet} is the LogDet divergence, which is also known as Stein's loss. It can be shown that Stein's loss is the unique scale invariant loss-function for which the uniform minimum variance unbiased estimator is also a minimum risk equivariant estimator (Davis et al. 2007). The authors in Davis et al. (2007) also proposed an efficient *Bregman projection* approach to solve problem (49). ITML is a global and linear approach. The learned distance metric is the Mahalanobis distance with precision matrix $\mathbf{M}$. It is easy to extend ITML to out-of-sample data as the precision matrix $\mathbf{M}$ is learned, which can be used to evaluate the Mahalanobis distance between any data pairs, and the computational technique involved is Bregman projection.

3.2.6 Neighborhood component analysis (NCA)

All the supervised approaches we introduced above are global methods. Next we will also introduce several representative local supervised metric learning algorithms. First we will overview NCA (Goldberger et al. 2004). Similar as in SNE described in unsupervised metric learning, each point $\mathbf{x}_i$ selects another point $\mathbf{x}_j$ as its neighbor with some probability p_{ij}, and inherits its class label from the point it selects. NCA defines the probability that point i selects point j as a neighbor:

$$p_{ij} = \frac{\exp\left(-\|\mathbf{W}^\top \mathbf{x}_i - \mathbf{W}^\top \mathbf{x}_j\|^2\right)}{\sum_{k\neq i}\exp\left(-\|\mathbf{W}^\top \mathbf{x}_i - \mathbf{W}^\top \mathbf{x}_k\|^2\right)} \tag{51}$$

Under this stochastic selection rule, NCA computes the probability that point i will be correctly classified

$$p_i = \sum_{j\in\mathcal{L}_i} p_{ij} \tag{52}$$

where $\mathcal{L}_i = \{j|l_i = l_j\})$ that is the set of points in the same class as point i.

The objective NCA maximizes is the expected number of points correctly classified under this scheme:

$$\mathcal{J}(\mathbf{W}) = \sum_i p_i = \sum_i \sum_{j\in\mathcal{L}_i} p_{ij} \tag{53}$$

Goldberger et al. (2004) proposed a truncated gradient descent approach to minimize $\mathcal{J}(\mathbf{W})$. NCA is a local and linear approach. The learned distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ is the Euclidean distance between $\mathbf{W}^\top\mathbf{x}_i$ and $\mathbf{W}^\top\mathbf{x}_j$. It is easy to extend NCA to out-of-sample data as the explicit mapping $\mathbf{W}$ is learned, and the computational technique involved is eigenvalue decomposition.

3.2.7 Average neighborhood margin maximization (ANMM)

ANMM (Wang and Zhang 2007) is another local supervised metric learning approach, which aims to find projection directions where the local class discriminability is maximized. To define local discriminability, Wang and Zhang (2007) first defines the following two types of neighborhoods:

Definition 1 (*Homogeneous Neighborhoods*) For a data point $\mathbf{x}_i$, its ξ *nearest homogeneous neighborhood* $\mathcal{N}_i^o$ is the set of ξ most similar[5] data which are in the same class with $\mathbf{x}_i$.

[5] In this paper two data vectors are considered to be similar if the Euclidean distance between them is small, two data tensors are considered to be similar if the Frobenius norm of their difference tensor is small.

Definition 2 (*Heterogeneous Neighborhoods*) For a data point $\mathbf{x}_i$, its ζ *nearest heterogeneous neighborhood* $\mathcal{N}_i^e$ is the set of ζ most similar data which are not in the same class with $\mathbf{x}_i$.

Then the *average neighborhood margin* γ_i for $\mathbf{x}_i$ is defined as

$$\gamma_i = \sum_{k:\mathbf{x}_k \in \mathcal{N}_i^e} \frac{\|\mathbf{y}_i - \mathbf{y}_k\|^2}{|\mathcal{N}_i^e|} - \sum_{j:\mathbf{x}_j \in \mathcal{N}_i^o} \frac{\left\|\mathbf{y}_i - \mathbf{y}_j\right\|^2}{|\mathcal{N}_i^o|},$$

where $|\cdot|$ represents the cardinality of a set. This margin measures the difference between the average distance from $\mathbf{x}_i$ to the data points in its heterogeneous neighborhood and the average distance from it to the data points in its homogeneous neighborhood. The maximization of such a margin can push the data points whose labels are different from $\mathbf{x}_i$ away from $\mathbf{x}_i$ while pull the data points having the same class label with $\mathbf{x}_i$ towards $\mathbf{x}_i$. It is easy to extend ANMM to out-of-sample data as the explicit mapping $\mathbf{W}$ is learned, and the computational technique involved is eigenvalue decomposition.

Therefore, the total *average neighborhood margin* can be defined as

$$\gamma = \sum\nolimits_i \gamma_i = \sum_i \left(\sum_{k:\mathbf{x}_k \in \mathcal{N}_i^e} \frac{\|\mathbf{y}_i - \mathbf{y}_k\|^2}{|\mathcal{N}_i^e|} - \sum_{j:\mathbf{x}_j \in \mathcal{N}_i^o} \frac{\left\|\mathbf{y}_i - \mathbf{y}_j\right\|^2}{|\mathcal{N}_i^o|} \right) \tag{54}$$

and the *ANMM criterion* is to maximize γ. By replacing $\mathbf{y}_i = \mathbf{W}^\top \mathbf{x}_i$, Wang and Zhang (2007) obtains the optimal $\mathbf{W}$ by performing eigenvalue decomposition of some discriminability matrix. Thus ANMM is a local and linear approach. The learned distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ is the Euclidean distance between $\mathbf{W}^\top \mathbf{x}_i$ and $\mathbf{W}^\top \mathbf{x}_j$. The authors in Wang and Zhang (2007) also proposed a kernelized version of ANMM to handle nonlinear data called KANMM, thus KANMM is local and nonlinear approach.

3.2.8 Large margin nearest neighbor classifier (LMNN)

The last local supervised metric learning approach we want to introduce is the LMNN (Weinberger et al. 2005). The goal of LMNN is similar as ANMM, i.e., to pull the data with same labels closer while push data with different labels far apart. LMNN deploys a different margin formulation. Specifically, LMNN defines the pull energy term as

$$\varepsilon_{pull} = \sum_{\mathbf{x}_j \in \mathcal{N}_i^o} \left\| \mathbf{W}^\top (\mathbf{x}_i - \mathbf{x}_j) \right\|^2 \tag{55}$$

which is the sum of pairwise distances between a data point $\mathbf{x}_i$ and the data point in $\mathbf{x}_i$'s homogeneous neighborhood after projection. LMNN defines the push energy as

$$\varepsilon_{push} = \sum_{i} \sum_{\mathbf{x}_j \in \mathcal{N}_i^o} \sum_{l} (1 - \delta_{il}) \left[1 + \left\| \mathbf{W}^\top (\mathbf{x}_i - \mathbf{x}_j) \right\|^2 - \left\| \mathbf{W}^\top (\mathbf{x}_i - \mathbf{x}_l) \right\|^2 \right]_+ \tag{56}$$

where $\delta_{il} = 1$ is the labels of $\mathbf{x}_i$ and $\mathbf{x}_l$ are the same, and $\delta_{il} = 0$ otherwise. The intuition is to require the data points from different classes should be at least separated from it by the distance 1. This formulation is very similar to the margin formulation in multiclass SVM (Crammer and Singer 2001) The above push energy term, for every data point, LMNN also pushes the data with different labels to at least distance one from its homogeneous neighborhood. The goal of LMNN is to minimize

$$\varepsilon = \mu \varepsilon_{pull} + (1 - \mu) \varepsilon_{push} \tag{57}$$

The authors in Weinberger et al. (2005) proposed a semi-definite programming technique to solve for $\mathbf{M} = \mathbf{W}\mathbf{W}^\top$. Thus LMNN is a local and linear approach. The learned distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ is the Euclidean distance between $\mathbf{W}^\top \mathbf{x}_i$ and $\mathbf{W}^\top \mathbf{x}_j$. It is easy to extend LMNN to out-of-sample data as the explicit mapping $\mathbf{W}$ is learned, and the computational technique involved is quadratic programming.

Summary Similar to unsupervised approaches, there is no free lunch for supervised approaches. All of the supervised metric learning approaches we listed above require some pre-specified free parameters, and all of them involve some expensive computational procedures such as eigenvalue decomposition or semi-definite programming. One exception is the ITML approach, as it deploys a Bregman projection strategy which may make the solution relatively efficient. However, ITML is sensitive to the initial choice of $\mathbf{M}_0$, which makes it difficult to apply in the case when we do not have enough prior knowledge. In practice, according to the type of supervision information provided, we can select a proper supervised metric learning approach that can handle those supervision information. However, in many real world applications, it may be expensive and time consuming to get those supervision information. There are also works aiming at exploring other forms of supervision that are easier to get Schultz and Joachims (2004). Next we survey semi-supervised approaches that can leverage both labeled and unlabeled information.

3.3 Semi-supervised distance metric learning

Semi-supervised approaches aim to learn a distance metric from the data where the supervision information is only available on a small portion of them. Those algorithms utilize both data with and without supervision information in the learning process. Therefore one straightforward way one can think of to construct a semi-supervised algorithm is to deploy an objective as in Eq. (7) with $\lambda_1 \neq 0, \lambda_2 \neq 0$, and $\mathcal{U}(\mathcal{D})$ is constructed on the entire data with some methodologies from Sect. 3.1, $\mathcal{L}(\mathcal{D})$ is constructed on the data with supervision information only with some approaches from Sect. 3.2. Finally we put some constraints on the learned distance metric to balance both parts. Table 4 summarizes the semi-supervised distance metric learning algorithms we will introduce.

Table 4 Semi-supervised distance metric learning algorithms (all the abbreviations can be found in the main text)

	Local	Global
Linear	LRML Hoi et al. (2008)	LRML Hoi et al. (2008), CMM Wang et al. (2008)
Nonlinear	KLRML Hoi et al. (2008), SSDM Yang et al. (2006), MPCK-means Bilenko et al. (2004)	KLRML Hoi et al. (2008), KCMM Wang et al. (2008), SSDM Yang et al. (2006)

3.3.1 Laplacian regularized metric learning (LRML)

LMRL (Hoi et al. 2008) is one semi-supervised distance metric learning approach. LRML adopts LPP formulation (described in Sect. 3.1.4) as the unsupervised term $\mathcal{U}(\mathcal{D})$; in terms of the supervised term, LRML chooses ANMM type of objective as $\mathcal{L}(\mathcal{D})$. The optimization problem that LRML aims to solve is

$$\begin{aligned} \min_{\mathbf{M}} \quad & \underbrace{t}_{\mathcal{U}(\mathcal{D})} + \underbrace{\gamma_1 t_2 - \gamma_2 t_3}_{\mathcal{L}(\mathcal{D})} \\ s.t. \quad & t_1 \leqslant t \\ & \mathbf{M} \succeq 0 \end{aligned} \tag{58}$$

where the smoothness term is

$$t_1 = \sum_{i,j} \|\mathbf{W}^\top \mathbf{x}_i - \mathbf{W}^\top \mathbf{x}_j\|^2 \omega_{ij} = tr(\mathbf{W}^\top \mathbf{X}\mathbf{L}\mathbf{X}^\top \mathbf{W}) = tr(\mathbf{X}\mathbf{L}\mathbf{X}^\top \mathbf{M}) \tag{59}$$

where $\mathbf{M} = \mathbf{W}\mathbf{W}^\top$. The supervised terms consisting of compactness and scatterness are

$$t_2 = \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{M}} \|\mathbf{W}^\top \mathbf{x}_i - \mathbf{W}^\top \mathbf{x}_j\|^2 = tr\left[\mathbf{M} \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{M}} (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top\right] \tag{60}$$

$$t_3 = \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{C}} \|\mathbf{W}^\top \mathbf{x}_i - \mathbf{W}^\top \mathbf{x}_j\|^2 = tr\left[\mathbf{M} \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{C}} (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top\right] \tag{61}$$

where $\mathcal{M}$ and $\mathcal{C}$ are the sets of must-links and cannot-links, respectively.

Hoi et al. (2008) proposed a semi-definite programming approach for solving problem (58). LRML is a mixture of local (its unsupervised part) and global (its supervised part) approach, and it is linear. The learned distance is the Mahalanobis distance with precision matrix **M**. It is easy to extend LRML to out-of-sample data as the precision matrix **M** is learned, and the computational technique involved is quadratic programming.

3.3.2 Constraint margin maximization (CMM)

Similarly, CMM (Wang et al. 2008) selects the same supervised term as LRML, but a different PCA type unsupervised term in its objective. Specifically, the optimization problem CMM aims to solve is

$$\begin{aligned} \max_{\mathbf{W}} \quad & \underbrace{t_4}_{\mathcal{U}(\mathcal{D})} - \underbrace{\gamma_1 t_2 - \gamma_2 t_3}_{\mathcal{L}(\mathcal{D})} \\ s.t. \quad & \mathbf{W}^\top \mathbf{W} = \mathbf{I} \end{aligned} \tag{62}$$

where the unsupervised term is $t_4 = tr(\mathbf{W}^\top \mathbf{X} \boldsymbol{\Sigma} \mathbf{X}^\top \mathbf{W})$ is the PCA objective. Note that before we apply CMM, all data points need to be centered, i.e., their mean should be subtracted from the data matrix. The intuition of CMM is to maximally unfold the data points in the projection space while at the same time satisfying those pairwise constraints. The authors in Wang et al. (2008) showed that the optimal $\mathbf{W}$ can be obtained by standard eigenvalue decomposition procedure. Therefore CMM is a global and linear approach. Wang et al. (2008) also showed how to derive its kernelized version for handling nonlinear data. The learned distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ is the Euclidean distance between $\mathbf{W}^\top \mathbf{x}_i$ and $\mathbf{W}^\top \mathbf{x}_j$. It is easy to extend CMM to out-of-sample data as the explicit mapping $\mathbf{W}$ is learned, and the computational technique involved is eigenvalue decomposition.

3.3.3 MPCK-means

MPCK-Means is a method proposed in Bilenko et al. (2004) which combines K-Means clustering and distance metric learning together with some available pairwise constraints. The goal of MPCK-Means is to partition the data set into a set of homogeneous clusters with K-means type algorithm, associated with each cluster there is a learned generalized Mahalanobis distance metric with different precision matrix. Specifically, MPCK-means aims to minimize the following objective

$$\begin{aligned} \mathcal{J} = & \sum_i \left(\|\mathbf{x}_i - \boldsymbol{\mu}_{l_i}\|^2_{\mathbf{M}_{l_i}} - \log(det(\mathbf{M}_{l_i})) \right) \\ & + \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M}} w_{ij} f_{\mathcal{M}}(\mathbf{x}_i, \mathbf{x}_j)(1 - \delta(l_i, l_j)) + \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{C}} \bar{w}_{ij} f_{\mathcal{C}}(\mathbf{x}_i, \mathbf{x}_j) \delta(l_i, l_j) \end{aligned} \tag{63}$$

where l_i indicates the cluster assignment for $\mathbf{x}_i$, $\mathbf{M}_{l_i}$ is the precision matrix for the generalized Mahalanobis distance for cluster l_i, $\|\mathbf{x}_i - \mathbf{x}_j\|^2_{\mathbf{M}} = (\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{M}(\mathbf{x}_i - \mathbf{x}_j)$, $det(\cdot)$ represents the matrix determinant, $\delta(l_i, l_j) = 1$ if $l_i = l_j$, and $\delta(l_i, l_j) = 0$ if $l_i \neq l_j$. $f_{\mathcal{M}}(\mathbf{x}_i, \mathbf{x}_j)$ and $f_{\mathcal{C}}(\mathbf{x}_i, \mathbf{x}_j)$ are defined as

$$f_{\mathcal{M}}(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{2}\|\mathbf{x}_i - \mathbf{x}_j\|^2_{\mathbf{M}_{l_i}} + \frac{1}{2}\|\mathbf{x}_i - \mathbf{x}_j\|^2_{\mathbf{M}_{l_j}} \tag{64}$$

$$f_{\mathcal{C}}(\mathbf{x}_i, \mathbf{x}_j) = \max_{(\mathbf{x}'_i, \mathbf{x}'_j)} \|\mathbf{x}'_i - \mathbf{x}'_j\|^2_{\mathbf{M}_{l_i}} - \|\mathbf{x}_i - \mathbf{x}_j\|^2_{\mathbf{M}_{l_i}} \tag{65}$$

Bilenko et al. (2004) proposed a expectation-maximization (EM) style iterative approach for minimizing the objective. When extending MPCK-Means to out-of-sample data, we need two steps. The first step is to assign the data point to its corresponding cluster, the second step is to measure the distance using the corresponding precision matrix of that cluster.

3.3.4 Semi-supervised dimensionality reduction (SSDR)

The last semi-supervised approach we want to introduce here is SSDR (Yang et al. 2006). SSDR deploys a similar objective as LE, however, it does not have any supervised terms, as it enforces the supervised information as hard constraints. Specifically, it aims at minimizing the following objective

$$\min_{\mathbf{Y}^\top\mathbf{Y}=\mathbf{I}} tr(\mathbf{Y}^\top\mathbf{A}\mathbf{Y}) \tag{66}$$

where $\mathbf{A}$ can be the Laplacian matrix as in LE. SSDR assumes that in the embedded space, the coordinates of some data points are known as $\mathbf{Y}_L$. Now we partition $\mathbf{Y}$ as $\mathbf{Y}^\top = [\mathbf{Y}_L^\top, \mathbf{Y}_U^\top]$, where $\mathbf{Y}_U$ corresponds to the unknown data coornidates which need to be solved, and $\mathbf{A}$ as

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{LL} & \mathbf{A}_{LU} \\ \mathbf{A}_{LU}^\top & \mathbf{A}_{UU} \end{bmatrix} \tag{67}$$

Then

$$\begin{aligned} \mathbf{Y}^\top\mathbf{A}\mathbf{Y} &= \left[\mathbf{Y}_L^\top, \mathbf{Y}_U^\top\right] \begin{bmatrix} \mathbf{A}_{LL} & \mathbf{A}_{LU} \\ \mathbf{A}_{LU}^\top & \mathbf{A}_{UU} \end{bmatrix} \begin{bmatrix} \mathbf{Y}_L \\ \mathbf{Y}_U \end{bmatrix} \\ &= \left[\mathbf{Y}_L^\top\mathbf{A}_{LL} + \mathbf{Y}_U^\top\mathbf{A}_{LU}^\top, \mathbf{Y}_L^\top\mathbf{A}_{LU} + \mathbf{Y}_U^\top\mathbf{A}_{UU}\right] \begin{bmatrix} \mathbf{Y}_L \\ \mathbf{Y}_U \end{bmatrix} \\ &= \mathbf{Y}_L^\top\mathbf{A}_{LL}\mathbf{Y}_L + \mathbf{Y}_U^\top\mathbf{A}_{LU}^\top\mathbf{Y}_L + \mathbf{Y}_L^\top\mathbf{A}_{LU}\mathbf{Y}_U + \mathbf{Y}_U^\top\mathbf{A}_{UU}\mathbf{Y}_U \end{aligned} \tag{68}$$

Thus we only need to minimize the following objective to get the optimal $\mathbf{Y}_U$

$$\mathcal{J}(\mathbf{Y}_U) = 2\mathbf{Y}_L^\top\mathbf{A}_{LU}\mathbf{Y}_U + \mathbf{Y}_U^\top\mathbf{A}_{UU}\mathbf{Y}_U \tag{69}$$

Depending on the construction of $\mathbf{A}$, SSDR can either be global (e.g., $\mathbf{A}$ is the data covariance matrix as PCA) or local (e.g., $\mathbf{A}$ is the Laplacian matrix as in LE). SSDR is a nonlinear method as no explicit mapping function is learned. The learned distance are the Euclidean distance between the corresponding coordinates from $\mathbf{Y}$. The learned distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ is the Euclidean distance between $\mathbf{y}_i$ and $\mathbf{y}_j$. It is not easy to extend SSDR to out-of-sample data as it directly learns the low-dimensional embeddings, and the computational technique involved is linear equation group solution.

Summary Semi-supervised approaches can be viewed as the marriage between supervised and unsupervised approaches. They could be helpful when the supervision information on the data is very sparse and limited. However, not necessarily all supervision information is helpful to the learned distance metric. Some researchers have done researches on how to measure the utility of the supervision information (Davidson et al. 2006). There are also research on when the unlabeled data could be helpful (Singh et al. 2008). In practice, it depends on the underlying data distribution to balance the effect of the supervision information or unlabeled data. To our best knowledge, there is no universal rules to set the optimal tradeoff parameter.

4 Advanced topics on distance metric learning

So far we have surveyed many distance metric learning algorithms, where we introduced in detail how these metric learning approaches are motivated, formulated and solved. In this section we will review some advanced distance metric learning topics emerged in recent years including online learning, distributed learning, active learning and transfer learning.

4.1 Online learning of distance metrics

Most of the distance metric learning approaches involves expensive optimization procedures such as eigen-decomposition and semi-definite programming. One way to make those algorithms more efficient is the *online learning* Shalev-Shwartz (2007) strategy, which incorporates the data points into the learning process in a sequential manner. More concretely, online learning updates the learned distance metric iteratively. At each iteration, only one or a small batch of data are involved. Another scenario that the online learning strategy can be naturally applied is to learn distance metrics for streaming data, where the data are coming in a streaming fashion so that the distance metric needs to be updated iteratively. Next we present two examples of online distance metric learning approaches.

4.1.1 Pseudo-metric online learning algorithm (POLA)

Pseudo-metric online learning (POLA) (Shalev-Shwartz et al. 2004) falls into the category of supervised metric learning with pairwise constraints. More specifically, there is a must-link constraint set $\mathcal{M}$ and a cannot-link constraint set $\mathcal{C}$. POLA assigns a label l_{ij} for each pair $(\mathbf{x}_i, \mathbf{x}_j)$ in $\mathcal{M}$ or $\mathcal{C}$, such that if $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M}$, $l_{ij} = 1$; if $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{C}$, $l_{ij} = -1$. Then it introduces a threshold b and construct the following constraints

$$\forall(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M},\ l_{ij} = 1 \Rightarrow (\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{M}(\mathbf{x}_i - \mathbf{x}_j) \leqslant b - 1 \tag{70}$$

$$\forall(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{C},\ l_{ij} = -1 \Rightarrow (\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{M}(\mathbf{x}_i - \mathbf{x}_j) \geqslant b + 1 \tag{71}$$

which can be unified as

$$l_{ij}\left[b - (\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{M}(\mathbf{x}_i - \mathbf{x}_j)\right] \geqslant 1 \tag{72}$$

Note that this formulation is similar to the constraint in standard SVM. Then the objective of POLA is the following hinge loss[6]

$$\mathcal{J}_{ij}(\mathbf{M}, b) = \max_{\mathcal{C}_{ij}^1 \& \mathcal{C}^2} (0, l_{ij}((\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{M}(\mathbf{x}_i - \mathbf{x}_j) - b) + 1) \tag{73}$$

The two constraint sets are defined as

$$\mathcal{C}_{ij}^1 = \left\{(\mathbf{M}, b) \in \mathbb{R}^{n^2 \times 1} : \mathcal{J}_{ij}(\mathbf{M}, b) = 0\right\} \tag{74}$$

$$\mathcal{C}^2 = \left\{(\mathbf{M}, b) \in \mathbb{R}^{n^2 \times 1} : \mathbf{M} \succeq 0, \ b \geqslant 1\right\} \tag{75}$$

POLA operates in an iterative way: Firstly, POLA initializes $\mathbf{M}$ as a zero matrix, then at each step, it randomly picks one data pair from the constraint set (either $\mathcal{M}$ or $\mathcal{C}$), and then do projections on $\mathcal{C}_{ij}^1$ and $\mathcal{C}^2$ alternatively. If we treat $(\mathbf{M}, b)$ as an $n^2 + 1$ dimensional vector, then the projection of a vector $\mathbf{v}$ onto a constraint set $\mathcal{C}_{ij}^1$ or $\mathcal{C}^2$. By projecting $\mathbf{M}$ and b onto $\mathcal{C}_{ij}^1$, POLA gets the updating rules for $\mathbf{M}$ and b as

$$\widehat{\mathbf{M}} = \mathbf{M} - l_{ij}\alpha_{ij}\mathbf{u}_{ij}\mathbf{u}_{ij}^\top \tag{76}$$

$$\widehat{b} = b + \alpha_{ij} l_{ij} \tag{77}$$

where

$$\mathbf{u}_{ij} = \mathbf{x}_i - \mathbf{x}_j \tag{78}$$

$$\alpha_{ij} = \frac{\mathcal{J}_{ij}(\mathbf{M}, b)}{\|\mathbf{u}_{ij}\|_2^4 + 1} \tag{79}$$

By projecting $(\mathbf{M}, b)$ onto $\mathcal{C}^2$, POLA updates $\mathbf{M}$ as

$$\widehat{\mathbf{M}} = \mathbf{M} - \lambda \boldsymbol{\mu}\boldsymbol{\mu}^\top \tag{80}$$

where $\lambda = \min\{\tilde{\lambda}, 0\}$ and $(\tilde{\lambda}, \mu)$ are the smallest eigenvalue-eigenvector pair of $\widehat{\mathbf{M}}$. Therefore, POLA incorporates the data in constraint sets in a sequential manner.

4.1.2 Online information theoretic metric learning (OITML)

Another technique we want to review here is the OITML approach (Davis et al. 2007). This method also falls into the category of supervised metric learning. It is the online version of the ITML approach we introduced in Sect. 3.2.5. Suppose at time $t + 1$, we need to randomly pick a pair of data from the constraint set, and minimize the following objective

[6] http://en.wikipedia.org/wiki/Hinge_loss

$$\mathbf{M}_{t+1} = arg\min_{\mathbf{M}} \mathcal{R}(\mathbf{M}, \mathbf{M}_t) + \eta_t \ell(\mathcal{D}_t, \widehat{\mathcal{D}}_t) \tag{81}$$

where $\widehat{\mathcal{D}}_t = (\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{M}(\mathbf{x}_i - \mathbf{x}_j)$ and $\mathcal{R}(\mathbf{M}, \mathbf{M}_t) = d_{logdet}(\mathbf{M}, \mathbf{M}_t)$ is the logdet divergence. Davis et al. (2007) showed that $\mathbf{M}_{t+1}$ can be updated with the following rule

$$\mathbf{M}_{t+1} \leftarrow \mathbf{M}_t - \frac{2\eta_t(\mathcal{D}_t - \widehat{\mathcal{D}}_t)\mathbf{M}_t(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{M}_t}{1 + 2\eta_t(\mathcal{D}_t - \widehat{\mathcal{D}}_t)(\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{M}_t(\mathbf{x}_i - \mathbf{x}_j)} \tag{82}$$

where

$$\eta_t = \begin{cases} \eta_0, \ \mathcal{D}_t - \widehat{\mathcal{D}}_t \leqslant 0 \\ \min\left\{\eta_0, \frac{1}{2(\mathcal{D}_t - \widehat{\mathcal{D}}_t)}\left(\frac{1}{(\mathbf{x}_i - \mathbf{x}_j)^\top(\mathbf{I} + (\mathbf{M}_t^{-1} - \mathbf{I})^{-1})(\mathbf{x}_i - \mathbf{x}_j)}\right)\right\}, & \text{otherwise} \end{cases} \tag{83}$$

POLA and OITML are only two examples of online distance metric learning two specific base metric learning algorithm. The key is to develop an online optimization strategy for solving the instantiation of Eq. (7).

4.2 Bayesian active distance metric learning (BADML)

Active learning (Dasgupta and Langford 2009) is a form of supervised machine learning in which the algorithm can interactively sample new data points. The goal of active distance metric learning is to select those unlabeled example pairs with the greatest uncertainty in relative distance (Yang et al. 2007). Intuitively, active distance metric learning wants to select the most *confusing* data pairs such that the distance metric can be more effectively and efficiently learned. The authors in Yang et al. (2007) proposed a Bayesian approach called BADML to achieve such a goal. Specifically, BADML first assumes the data outer product matrix $\mathbf{X}\mathbf{X}^\top$ can be factorized as

$$\mathbf{X}\mathbf{X}^\top = \sum_q \lambda_q \boldsymbol{\mu}_q \boldsymbol{\mu}_q^\top \tag{84}$$

where $(\lambda_q, \boldsymbol{\mu}_q)$ corresponds to the q-th eigenvalue-eigenvector pair of $\mathbf{X}\mathbf{X}^\top$. Moreover, BADML also supposes the precision matrix $\mathbf{M}$ takes the following parametric form

$$\mathbf{M} = \sum_q \gamma_q \boldsymbol{\mu}_q \boldsymbol{\mu}_q^\top \tag{85}$$

Then BADML defines the probability of the label of $(\mathbf{x}_i, \mathbf{x}_j)$ is l_{ij} as

$$P(l_{ij}|\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{1 + \exp\left(l_{ij}(\|\mathbf{x}_i - \mathbf{x}_j\|_{\mathbf{M}}^2 - b)\right)} \tag{86}$$

where l_{ij} is +1 if there is a must-link between $\mathbf{x}_i$ and $\mathbf{x}_j$, and l_{ij} is -1 if there is a cannot-link between $\mathbf{x}_i$ and $\mathbf{x}_j$. $\|\mathbf{x}_i - \mathbf{x}_j\|_{\mathbf{M}}$ is the generalized Mahalanobis distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ parameterized by precision matrix $\mathbf{M}$, and $P(l_{ij}|\mathbf{x}_i, \mathbf{x}_j)$ can be further rewritten as

$$P(l_{ij}|\mathbf{x}_i, \mathbf{x}_j) = \sigma(-l_{ij}\boldsymbol{\gamma}^\top \boldsymbol{\omega}_{ij}) \tag{87}$$

where $\boldsymbol{\gamma} = [b, \gamma_1, \gamma_2, \cdots, \gamma_K]^\top$, $\boldsymbol{\omega} = \left[-1, \omega_{ij}^1, \omega_{ij}^1, \cdots, \omega_{ij}^K\right]^\top$, $\sigma(z) = 1/(1 + \exp(-z))$ and

$$\omega_{ij}^q = (\mathbf{x}_i - \mathbf{x}_j)^\top \boldsymbol{\mu}_q \tag{88}$$

Thus

$$P(\mathcal{M}, \mathcal{C}|\mathbf{M}, b) = \prod_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{M} \text{ or} (\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{C}} P(l_{ij}|\mathbf{x}_i, \mathbf{x}_j) \tag{89}$$

and

$$P(\mathcal{M}, \mathcal{C}) = \int P(\mathbf{M})d\mathbf{M}P(b)db \prod_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{M} \text{ or } (\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{C}} P(l_{ij}|\mathbf{x}_i, \mathbf{x}_j) \tag{90}$$

where $P(\mathbf{M})$ is a Wishart distribution and $P(b)$ is a Gamma distribution. The authors in Jordan et al. (1999) developed a variational method to approximate $P(\mathcal{M}, \mathcal{C})$ and get the estimates of $\mathbf{M}$ and b. Finally for a given data pair, its label uncertainty can be measured by

$$\mathcal{H}_{ij} = -P(-|\mathbf{x}_i, \mathbf{x}_j) \log P(-|\mathbf{x}_i, \mathbf{x}_j) - P(+|\mathbf{x}_i, \mathbf{x}_j) \log P(+|\mathbf{x}_i, \mathbf{x}_j) \tag{91}$$

BADML just selects the data pairs $(\mathbf{x}_i, \mathbf{x}_j)$ with large $\mathcal{H}_{ij}$ as the candidate pairs for labeling.

The key to active learning methods is how to evaluate the uncertainty of a data point (when the supervision information is data label) or a pair of data points (when the supervision information is pairwise constraints). The most uncertain points and pairs of points are selected to be labeled and used to update the model. The learned distance is the resulting generalized Mahalanobis distance.

4.3 Transfer distance metric learning

Transfer learning focuses on transferring knowledge gained from tasks in source domains into solving another task in a target domain (Pan and Yang 2010). The goal of transfer metric learning (TML) (Zhang and Yeung 2010) is to learn a good distance metric in the target domain with the help of the knowledge from source domains. Specifically, TML supposes there are m different related tasks; the goal of TML is to solve the following optimization problem:

$$
\begin{aligned}
\min_{\{\mathbf{M}_i\}_{i=1},\boldsymbol{\Omega}} \quad & \sum_{k=1}^{m}\sum_{i,j} h\left(l_{ij}^k(1-\|\mathbf{x}_i^k-\mathbf{x}_j^k\|_{\mathbf{M}_k}^2)\right) \\
& +\frac{\lambda_1}{2}\sum_{k=1}^{m}\|\mathbf{M}_k\|_F^2+\frac{\lambda_2}{2}tr\left(\widehat{\mathbf{M}}\boldsymbol{\Omega}^{-1}\widehat{\mathbf{M}}^\top\right) \\
s.t. \quad & \forall k,\ \mathbf{M}_k \succeq 0 \\
& \widehat{\mathbf{M}}=[vec(\mathbf{M}_1), vec(\mathbf{M}_2),\cdots, vec(\mathbf{M}_m)] \\
& \boldsymbol{\Omega}\succeq 0 \\
& tr(\boldsymbol{\Omega})=1
\end{aligned}
$$

where $h\left(l_{ij}^k(1-\|\mathbf{x}_i^k-\mathbf{x}_j^k\|_{\mathbf{M}_k}^2)\right)$ is the hinge loss, and $\boldsymbol{\Omega}$ is used to capture the relationships between different tasks. Note that in the objective, the first term is the prediction loss on the pairwise supervisions, the second term is to make sure the predicted model does not over fit the training data, and the last term encodes the relationship between pairwise tasks. The solution procedure for the above problem is computationally expensive. The authors in Zhang and Yeung (2010) also presented an online algorithm to expedite the solution process.

5 Conclusions and future research directions

In this survey, we have reviewed various distance learning algorithms. According to the availability of the supervision information, we categorized the existing distance metric learning algorithms as unsupervised, supervised or semi-supervised. We presented a unified optimization view of distance metric learning and pointed out how those distance metric learning algorithms can be fit into the general framework. We also discussed the advantages and limitations of existing distance learning approaches, and finally introduced some novel researches in this field in recent years. For the future of distance metric learning research, we believe the following directions are promising.

- *Distance metric learning for big data* Most of the existing distance metric learning approaches involve computationally expensive procedure. How to make distance metric learning efficient and practical on large-scale data. Promising solution include online learning or distributed learning. We have introduced the most recent works on online distance metric learning in Sect. 4.1. For parallel/distributed distance metric algorithms, as the major computational techniques involved are eigenvalue decomposition and quadratic programming, we can adopt parallel matrix computation/optimization algorithms (Modi et al. 1989; Censor 1997) to make distance metric learning more scalable and efficient.
- *Empirical proof points* Although a lot of distance metric learning algorithms have been proposed, there is still lack of systematic comparison and proof points on the utility of many distance metric learning algorithms in real world applications. Such empirical discussion will be helpful to showcase the practical value of distance metric learning algorithms. Some recent works have started developing and

applying distance metric learning on healthcare for measuring similarity among patients (Wang et al. 2011a, b; Sun et al. 2010).

- *More general distance metric learning formulation* As can be seen from this survey, most of existing distance metric learning algorithms suppose the learned distance metric is *Euclidean* in the projected space. Such assumption may not be sufficient for real world applications as there is no guarantee that Euclidean distance is most appropriate to describe the pairwise data relationships. There are already some initial effort towards this direction (Elkan 2011; Li et al. 2012), and this direction is definitely worth exploring.

References

Bar-Hillel A, Hertz T, Shental N, Weinshall D (2005) Learning a mahalanobis metric from equivalence constraints. J Mach Learn Res 6(6):937–965

Belkin M, Niyogi P (2001) Laplacian eigenmaps and spectral techniques for embedding and clustering. Adv Neural Inf Process Syst 14:585–591

Bengio Y, Paiement J-F, Vincent P, Delalleau O, Le Roux N, Ouimet M (2004) Out-of-sample extensions for LLE, isomap, MDS, eigenmaps, and spectral clustering. In: Advances in neural information processing systems, vol 16, pp 177–184

Bilenko M, Basu S, Mooney RJ (2004) Integrating constraints and metric learning in semi-supervised clustering. In: Proceedings of the twenty-first international conference on Machine learning. ACM, Berlin, pp 11–18

Censor Y (1997) Parallel optimization: theory, algorithms, and applications. Oxford University Press, New York

Cox TF, Cox MAA (2000) Multidimensional scaling, 2nd edn. Chapman and Hall/CRC, Boca Raton

Crammer K, Singer Y (2001) On the algorithmic implementation of multiclass kernel-based vector machines. J Mach Learn Res 2:265–292

Dasgupta S, Langford J (2009) A tutorial on active learning. In: International conference on machine learning

Davidson I, Wagstaff KL, Basu S (2006) Measuring constraint-set utility for partitional clustering algorithms. In Proceedings of the 10th European conference on principles and practice of knowledge discovery in databases, pp 115–126

Davis JV, Kulis B, Jain P, Sra Suvrit, Dhillon IS (2007) Information-theoretic metric learning. In: International conference on machine learning (ICML), pp 209–216

Domeniconi C, Gunopulos D, Ma S, Yan B, Al-Razgan M, Papadopoulos D (2007) Locally adaptive metrics for clustering high dimensional data. Data Min Knowl Discov 14(1):63–97

Duda RO, Hart PE, Stork DG (2001) Pattern classification, vol 2, 2nd edn. Wiley, New York

Elkan C (2011) Bilinear models of affinity. Personal note

Fukunaga K (1990) Introduction to statistical pattern recognition, second edition (computer science and scientific computing series), 2nd edn. Academic Press, Boston

Goldberger J, Roweis S, Hinton G, Salakhutdinov R (2004) Neighborhood component analysis. In: Advances in neural information processing systems (NIPS)

Guo Y, Li S, Yang J, Shu T, Wu L (2003) A generalized foleysammon transform based on generalized fisher discriminant criterion and its application to face recognition. Pattern Recognit Lett 24(1–3):147–158

He J, Li M, Zhang HJ, Tong H, Zhang C (2006) Generalized manifold-ranking-based image retrieval. IEEE Trans Image Process 15(10):3170–3177

He X, Niyogi P (2004) Locality preserving projections. In: Advances in neural information processing systems (NIPS), vol 16, pp 234–241

Hinton GE, Roweis ST (2002) Stochastic neighbor embedding. In: Advances in neural information processing systems (NIPS), pp 833–840

Hoi Steven CH, Liu W, Chang S-F (2008) Semi-supervised distance metric learning for collaborative image retrieval. In: Proceedings of IEEE Computer Society conference on computer vision and pattern recognition

Jain AK, Dubes RC (1988) Algorithms for clustering data. Prentice-Hall Inc., Upper Saddle River

Jia Y, Nie F, Zhang C (2009) Trace ratio problem revisited. IEEE Trans Neural Netw 20(4):729–735

Jolliffe IT (2002) Principal component analysis, 2nd edn. Springer, New York

Jordan MI, Ghahramani Z, Jaakkola TS, Saul LK (1999) An introduction to variational methods for graphical models. Mach Learn 37(2):183–233

Kocsor A, Kovács K, Szepesvári C (2004) Margin maximizing discriminant analysis. In: Proceedings of European conference on machine learning, vol 3201 of Lecture notes in computer science. Springer, Berlin, pp 227–238

Kulis B (2010) Metric learning. In: Tutorial at International conference on machine learning

Kulis Brian (2012) Metric learning: a survey. Found Trends Mach Learn 5(4):287–364

Li Z, Cao L, Chang S, Smith JR, Huang TS (2012) Beyond mahalanobis distance: Learning second-order discriminant function for people verification. In: Prcoeedings of computer vision and pattern recognition workshops (CVPRW), 2012 IEEE computer society conference on workshops, pp 45–50

Luxburg U (2007) A tutorial on spectral clustering. Stat Comput 17(4):395–416

Mika S, Ratsch G, Weston J, Schölkopf B, Müllers KR (1999) Fisher discriminant analysis with kernels. In: Neural networks for signal processing IX, 1999. proceedings of the 1999 IEEE signal processing society workshop, pp 41–48

Modi JJ (1989) Parallel algorithms and matrix computation. Oxford University Press, Inc

Pan SJ, Yang Q (2010) A survey on transfer learning. IEEE Trans Knowl Data Eng 22:1345–1359

Pele O, Werman M (2010) The quadratic-chi histogram distance family. In: Computer vision ECCV 2010, volume 6312 of lecture notes in computer science, chapt 54. Springer, Berlin, pp 749–762

Roweis ST, Saul LK (2000) Nonlinear dimensionality reduction by locally linear embedding. Science 290(5500):2323–2326

Schölkopf B, Smola AJ (2002) Learning with kernels : support vector machines, regularization, optimization, and beyond. The MIT Press, Cambridge

Schultz M, Joachims T (2004) Learning a distance metric from relative comparisons. In: Advances in neural information processing systems (NIPS), vol 16, pp 41–48

Shalev-Shwartz S (2007, July) Online learning: theory, algorithms, and applications. The Hebrew University of Jerusalem. Ph.D. Thesis

Shalev-Shwartz S, Singer Y, Ng AY (2004) Online and batch learning of pseudo-metrics. In: Proceedings of international conference on machine learning, pp 94–101

Shental N, Hertz T, Weinshall D, Pavel M (2002) Adjustment learning and relevant component analysis. In: Proceedings of European conference on computer vision, pp 776–790

Singh A, Nowak RD, Zhu X (2008) Unlabeled data: now it helps, now it doesn't. In: Advances in neural information processing systems, pp 1513–1520

Sun J, Sow D, Hu J, Ebadollahi S (2010) Localized supervised metric learning on temporal physiological data. In: International conference on pattern recognition (ICPR)

Tenenbaum JB, Silva V, Langford JC (2000) A global geometric framework for nonlinear dimensionality reduction. Science 290(5500):2319–2323

Tsang IW, Cheung PM, Kwok JT (2005) Kernel relevant component analysis for distance metric learning. In: In IEEE International joint conference on neural networks (IJCNN), pp 954–959

Vapnik VN (1995) The nature of statistical learning theory. Springer, New York

Wang F, Chen S, Zhang C, Li T (2008) Semi-supervised metric learning by maximizing constraint margin. In: Proceedings of the 17th ACM conference on information and knowledge management, pp 1457–1458

Wang F, Sun J, Ebadollahi S (2011) Integrating distance metrics learned from multiple experts and its application in patient similarity assessment. In: SIAM data mining conference (SDM), pp 59–70

Wang F, Sun J, Hu J, Ebadollahi S (2011) Imet: interactive metric learning in healthcare applications. In: SIAM data mining conference (SDM), pp 944–955

Wang F, Zhang C (2007) Feature extraction by maximizing the average neighborhood margin. In: IEEE Computer Society conference on computer vision and pattern recognition (CVPR)

Wang F, Zhao B, Zhang C (2011) Unsupervised large margin discriminative projection. IEEE Trans Neural Netw 22(9):1446–1456

Weinberger KQ, Blitzer J, Saul LK (2005) Distance metric learning for large margin nearest neighbor classification. In: Advances in neural information processing systems

Weinberger KQ, Saul LK (2009) Distance metric learning for large margin nearest neighbor classification. J Mach Learn Res 10:207–244

Werman M, Pele O, Kulis B (2010) Distance functions and metric learning. In: Tutorial at European conference on computer vision

Xing EP, Ng AY, Jordan MI, Russell S (2002) Distance metric learning, with application to clustering with side-information. In: Advances in neural information processing systems (NIPS), vol 15, pp 505–512

Yang L, Jin R (2006) Distance metric learning: a comprehensive survey. Technical report, Department of Computer Science and Engineering, Michigan State University

Yang L, Jin R, Sukthankar R (2007) Bayesian active distance metric learning. In: Proceedings of uncertainties in artificial intelligence, AUAI Press, Corvallis, pp 442–449

Yang X, Fu H, Zha H, Barlow J (2006) Semi-supervised nonlinear dimensionality reduction. In: 23rd International conference on machine learning, pp 1065–1072

Zhang Y, Yeung D-Y (2010) Transfer metric learning by learning task relationships. In: Proceedings of the 18th ACM SIGKDD conference on knowledge discovery and data mining, pp 1199–1208