Neural Networks 105 (2018) 447–462

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Survey and experimental study on metric learning methods

Dewei Li [a,b], Yingjie Tian [b,c,*]

[a] School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing 100049, China
[b] Research Center on Fictitious Economy and Data Science, Chinese Academy of Sciences, China
[c] Key Laboratory of Big Data Mining and Knowledge Management, Chinese Academy of Sciences, Beijing 100190, China

ARTICLE INFO

Article history:
Received 24 August 2017
Received in revised form 14 March 2018
Accepted 5 June 2018
Available online 19 June 2018

Keywords:
Metric learning
Distance
Classification
Similarity
Nearest neighbor

ABSTRACT

Distance metric learning has been a hot research spot recently due to its high effectiveness and efficiency in improving the performance of distance related methods, such as k nearest neighbors (kNN). Metric learning aims to learn a data-dependent metric to make intra-class distance smaller and inter-class larger. A large number of methods have been proposed for various applications and a survey to evaluate and compare these methods is imperative. The existing surveys just analyze the algorithms theoretically or compare them experimentally with a narrow time scope. Therefore, the paper reviews classical and influential methods that were proposed between 2003 and 2017 and presents a taxonomy based on the most distinct character of each method. All the methods are categorized into five classes, including pairwise cost, probabilistic framework, boost-like approaches, advantageous variants and specific applications. A comprehensive experimental study is made to compare all the selected methods, exploring the ability in improving accuracy, the relation between distance change and accuracy, the relation between accuracy and kNN neighbor size.

1. Introduction

In machine learning, measuring similarity is significant in making prediction since any two similar patterns probably have the same output (label, cluster, etc.). An unknown pattern can be predicted correctly under proper similarity measurement. Distance is one of the most common means for similarity, which is applied in classical machine learning methods, including kNN (Cover, 1968; Cover & Hart, 1967) and k-means (Jain, 2008; Likas, Vlassis, & Verbeek, 2001). The performances of kNN and k-means rely heavily on distance and Euclidean distance is employed in most of the cases. However, Euclidean distance treats all the components of a feature vector equally, disregarding their different significances in determining the output of the vector. For example, in face identification, the distance between two face images should be mostly determined by the nose, eyes, mouth, other than all the components of the image. Metric learning can solve such task but its advantages are not limited to it: (1) Learning a data-dependent metric to discriminate the importance of different attributes may depict similarity more precisely. (2) Normalization is not necessary in metric learning since it can rescale each component of the input vectors. (3) Metric learning can make dimension reduction when the metric is decomposed by the transpose of a linear transformation multiply the transformation. (4) A learned metric can be generalized to test data well since all the information of supervision and pairwise distances is extracted in metric learning. Distance metric learning has been developed since 2003 to improve the performance of distance related methods and it attracts more and more attention recently due to its good effectiveness. The basic idea of metric learning is to make each point nearer to the points with the same label and farther from the points with different labels, namely, reduce intra-class distance and enlarge inter-class distance as much as possible. There are many difficulties and challenges in metric learning: (1) How to define intra-class distance and inter-class distance. A desired metric is learned under the constraints that intra-class distance is minimized and inter-class distance is maximized. The definitions of intra-class and inter-class are important in getting the optimal value of the metric. (2) How to deal with the constraint of positive semidefiniteness of the metric. For the completeness of the metric, a proper metric should be constrained by the property of positive semidefinite. The constraint is a big trouble for optimization, leading to high complexity and uneasily solving. (3) How to reduce complexity and make it as low as possible. In distance related methods, the information of pairwise distance should be used, which brings high computational complexity, especially when the data size is very large. A method with high complexity is intractable whatever its promising performance. (4) How to introduce metric learning into specific applications. In traditional classification task, distance is

* Corresponding author at: Research Center on Fictitious Economy and Data Science, Chinese Academy of Sciences, China.
E-mail address: tyj@ucas.ac.cn (Y. Tian).

https://doi.org/10.1016/j.neunet.2018.06.003
0893-6080/

Table 1
Selected top conferences and journals.

	Conference/Journal name		Conference/Journal name
AAAI(C)	AAAI Conference on Artificial Intelligence	AI(J)	Artificial Intelligence
ACCV(C)	Asian Conference on Computer Vision	IF(J)	Information Sciences
ACMMM(C)	ACM Multimedia Conference	JMLR(J)	Journal of Machine Learning Research
CVPR(C)	Conference on Computer Vision and Pattern Recognition	KBS(J)	Knowledge-Based Systems
ECCV(C)	European Conference on Computer Vision	ML(J)	Machine Learning
ECML(C)	European Conference on Machine Learning	NC(J)	Neural Computation
ICCV(C)	International Conference on Computer Vision	PAMI(J)	IEEE Trans. on Pattern Analysis and Machine Intelligence
ICDM(C)	IEEE International Conference on Data Mining	PR(J)	Pattern Recognition
ICML(C)	International Conference on Machine Learning	SP(J)	Signal Processing
IJCAI(C)	International Joint Conference on Artificial Intelligence	TCYB(J)	IEEE Trans. on Cybernetics
SIGKDD(C)	ACM Knowledge Discovery and Data Mining	TCSVT(J)	IEEE Trans. on Circuits and Systems for Video Technology
NIPS(C)	Conference on Neural Information Processing Systems	TIE(J)	IEEE Trans. on Industrial Electronics
UAI(C)	Conference on Uncertainty in Artificial Intelligence	TIFS(J)	IEEE Trans. on Information Forensics and Security
TIP(J)	IEEE Transactions on Image Processing	TNNLS(J)	IEEE Trans. on Neural Networks and Learning Systems

In parenthesis, (C) denotes Conference and (J) denotes Journal.

defined with two input vectors, but in specific applications, such as multi-instance learning, multi-view learning, multi-label learning, the pattern is not presented as a single vector, it is a challenge to define distance on such tasks.

A great many scholars have proposed many efficient algorithms to solve these difficulties according to their own understandings. In many hot and important applications, metric learning has been applied to obtain better performance, including face identification (Cai, Wang, Xiao, Chen, & Zhou, 2012; Cao, Ying, & Li, 2013; Guillaumin, Verbeek, & Schmid, 2009; Hu, Lu, & Tan, 2014), person re-identification (Ma, Yang, & Tao, 2014; Paisitkriangkrai, Shen, & van den Hengel, 2015; Tao, Jin, Wang, Yuan, & Li, 2013; Xiong, Gou, Camps, & Sznaier, 2014), image retrieval (Gao, Wang, Ji, Wu, & Dai, 2014; Hoi, Liu, & Chang, 2008; Hoi, Liu, Lyu, & Ma, 2006), image annotation (Feng, Jin, & Jain, 2013; Verma & Jawahar, 2012), image set classification (Lu, Wang, Deng, Moulin, & Zhou, 2015; Lu, Wang, & Moulin, 2013), document classification (Lebanon, 2006), target detection (Dong, Zhang, Zhang, & Du, 2015; Du & Zhang, 2014a, b). But there is no one method fit for all the tasks, confirms the No-free-lunch theorem (Shalev-Shwartz & Ben-David, 2014; Wolpert & Macready, 1997). It is necessary to make a survey on these literatures to analyze and compare them, giving suggestions on the selection of these methods. Most of the existing surveys divide metric learning methods into several categories generally, according to the availability of supervision information or the formation of the metric(linear or nonlinear) (Bellet, Habrard, & Sebban, 2013; Kulis, 0000; Liu, 2006; Moutafis, Leng, & Kakadiaris, 2017; Wang & Sun, 2015). And brief analyses are provided without systematic comparisons or empirical verification in these surveys. Moutafis et al. make experimental survey recently, which covers only a few papers, from 2011 to 2013. It ignores many classical and effective methods that proposed before 2011. So we give an experimental survey more comprehensively in this paper, complementary to the previous surveys. In this paper, we make an experimental study on distance metric learning literatures that presented from 2003 to 2017. The selected papers are all published on famous conference or journals, which are given in Table 1. We try to investigate as many papers as possible, but the papers with high citations are preferred due to their influence and the space limitation of our paper. We will first give a taxonomy for metric learning and introduce the methods briefly. The algorithms are classified into five categories, pairwise cost approaches, probabilistic framework methods, boost-like approaches, advantageous variants and specific applications. The key point of our paper is to explore the ability of these methods in improving the performance of kNN classification, filling the gaps in the field. The survey can be regarded as a complementary material of the previous surveys.

The rest of the paper is organized as follows. In Section 2, the definition of metric learning is given and a summary overview is introduced to make readers be familiar with previous surveys on metric learning. Metric learning taxonomy is presented in Section 3 and the previous methods will be introduced in five categories respectively. In Section 4, numerical experiments will be conducted to compare the performance of the selected methods on various datasets. Conclusions are summarized in Section 5.

2. The nature of metric learning

In this section, we will describe the definition of metric learning and then introduce the brief history for metric learning.

2.1. Problem definition

In traditional machine learning, the performance of distance related methods depends heavily on distance measurement. Give a classification training set

$$T = \{(x_i, y_i)\}_{i=1}^{m}, \tag{2.1}$$

where $x_i \in R^n$ is a input feature vector and $y_i \in \{1, 2, \ldots, c\}$ is the corresponding label. In kNN classification, an unlabeled example is predicted by the majority label of its nearest samples. Generally, the distance between two examples $x_i, x_j \in R^n$ is defined by Euclidean distance(squared),

$$d(x_i, x_j) = (x_i - x_j)^\top (x_i - x_j) = \|x_i - x_j\|^2. \tag{2.2}$$

Due to the drawbacks of Euclidean distance, it can be improved to be a generalized Mahalanobis distance with respect to a data-dependent metric M, which can be learned from the training data. And then the new distance between $x_i, x_j \in R^n$ is computed by

$$d_M(x_i, x_j) = (x_i - x_j)^\top M(x_i - x_j), \tag{2.3}$$

where M should satisfy four conditions, including distinguishability, non-negativity, symmetry and triangular inequality (Royden & Fitzpatrick, 1988; Wang & Sun, 2014). In short, M must be a positive semidefinite matrix. The key is how to learn a desired metric from the data at hand. The target of metric learning is to make similar points nearer and dissimilar points farther from each other. In supervised condition, where labels are available or semisupervised condition, where pairwise label relations are given, the constraints that shrink intra-class distance and expand inter-class distance can be developed based on label information. A desired metric can be learned under such constraints or their variants. Two sets can be constructed, similar set

$$S = \{(x_i, x_j) | y_i = y_j\}, \tag{2.4}$$

and dissimilar set

$$D = \{(x_i, x_j) | y_i \neq y_j\}, \tag{2.5}$$

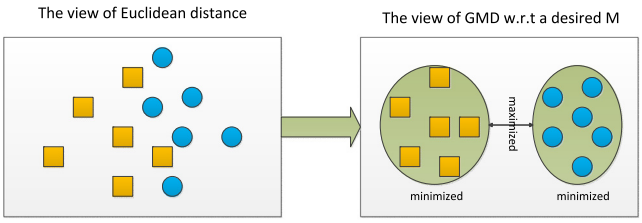


Fig. 1. A concept illustration of metric learning. There are two classes, orange squares and blue circles. Metric learning aims to separate the two classes in global range or local neighborhood. (GMD denotes generalized Mahalanobis distance, w.r.t denotes with respect to.)

based on supervision information. All the methods in metric learning make efforts in one or two sides of the following two optimizations,

$$\min_{M} \sum_{(x_i,x_j)\in S} d_M(x_i, x_j) \tag{2.6}$$

and

$$\max_{M} \sum_{(x_i,x_j)\in D} d_M(x_i, x_j). \tag{2.7}$$

Once the total distance in Eq. (2.6) can be minimized and/or the distance in Eq. (2.7) can be maximized, each example is nearer to its similar examples and farther from its dissimilar examples, leading to improvements on the prediction accuracy of kNN classification. A concept illustration of metric learning is given in Fig. 1.

Since any positive semidefinite matrix can be decomposed into $M = L^\top L$, then Eq. (2.3) can be rewritten as

$$d_M(x_i, x_j) = (x_i - x_j)^\top L^\top L(x_i - x_j) = \|Lx_i - Lx_j\|^2. \tag{2.8}$$

The equation shows that the generalized Mahalanobis distance d_M equals to Euclidean distance in a transformed space. So learning a positive semidefinite metric M is equivalent to learning a linear transformation L. In fact, the transformation is not limited to linear case, nonlinear transformation may obtain better performance. The distance between $x_i, x_j \in R^n$ under nonlinear mapping ϕ can be defined as

$$d_\phi(x_i, x_j) = \|\phi x_i - \phi x_j\|^2. \tag{2.9}$$

In a word, a generalized metric learning means to learn a linear or nonlinear transformation to map the original examples into a new space, in which within-class distance is shrinked and between-class distance is expanded. Then kNN classification can perform better in the transformed space.

2.2. A summary overview of metric learning

Metric learning has been an active topic in machine learning and many researchers had made comprehensive surveys in recent years to emphasize the significance of metric learning. Yang and Jin had summarized the metric learning methods that proposed from 1980s to 2006 (Liu, 2006). They argue that manifold learning is a kind of unsupervised metric learning since the main idea of manifold leaning is to learn an underlying low-dimensional manifold where geometric relationships are still preserved. In discussion of supervised metric learning, global metric learning and local metric learning are introduced separately. Global metric learning make constraints on the whole dataset but local metric learning constrains data points in a local neighborhood. Due to the similarity between SVM and metric learning, both of which aim to maintain large margin between different classes, some variants of metric learning based on SVM had been introduced. At last, kernel methods for metric learning are summarized. Compared with the previous linear transformation related metric learning methods, kernel methods make nonlinear transformation and extract nonlinear information to get better performance. The survey is one of the earliest summarizations of metric learning methods. However, many methods in this survey are outdated and the taxonomy is too general to show the essential characters of the methods. In 2012, Brian Kulis made a survey on metric learning in the view of space transformation (Kulis, 0000). In linear transformation case, regularized transformation learning with different regularizers are introduced, including Frobenius norm regularizer, linear regularizer and LogDet regularizer. The optimization techniques that useful in metric learning are explained detailedly. In nonlinear case, the kernel versions of linear methods are illustrated in detail. In the end, some extensions on metric learning are summarized, metric learning for kernel regression, ranking and dimension reduction are included. This work make deeper analysis on metric learning, especially generalized common points of math formulation, such as regularizer, optimization algorithm. Bellet et al. gave a survey on metric learning for feature vectors and structured data in 2014 (Bellet et al., 2013). First, supervised Mahalanobis metric learning methods are introduced and they are divided into several categories, nearest neighbor approaches, information-theoretic approaches, online approaches, multi-tasks approaches, etc. Then some other extensions and highly related topics are presented, such as nonlinear metric learning, local metric learning, semi-supervised metric learning, similarity learning. The followed part focuses on metric learning methods for structured data. Metric is regarded as an attractive proxy to access structured data. Later, Wang and Sun summarized distance metric learning methods and their relationships with dimension reduction, in 2015 (Wang & Sun, 2015). According to the provided label information, three classes are introduced respectively, unsupervised metric learning, supervised metric learning and semi-supervised metric learning. Some advanced topics are summed up in the following chapter, online metric learning, transfer metric learning, Bayesian active metric learning. The mentioned methods can be fit into a general framework to make readers understand the principles more easily. The above mentioned surveys just introduce the existing methods generally and give brief analysis for these methods. They do not make comprehensive comparisons among the proposed methods theoretically or experimentally. Moutafis et al. make empirical study on metric learning methods recently (Moutafis et al., 2017). The study focuses on the literatures that published from 2011 to 2013 and divides them into five classes, ensemble, nonlinear, regularized, probabilistic and cost-variant. The experiments are conducted on the selected LFW dataset to compare the performances of different methods. The experimental survey does not cover the papers that published before 2011, which may be very effective in learning metric. The experiments on only one dataset cannot really validate the ability of the methods in improving kNN comprehensively. In a word, the study is too narrow, should be extended with more literatures. Therefore, an experimental survey is made in this paper to cover the shortages of the existing surveys, analyzing the literatures briefly and providing experimental evidences to validate the ability of these methods in improving performance of kNN. It should be noted that, in our opinion, manifold learning does not change the intrinsic structure of the original space and cannot use label information to alter intra-class distance and inter-class distance. So unsupervised metric learning will not discussed in our paper.

3. Metric learning taxonomy

In this section, a taxonomy for metric learning methods is proposed to review this research field from various sides. The time

range of our study is from 2003, when the first supervised method on metric learning is introduced, to 2017. And all the papers are selected from top conference and journals, which are shown in Table 1. According to their most distinguished character, the selected methods will be divided into five categories: (1) Pairwise cost, the main idea of which is to construct optimization problem based on the cost of pairwise distance. (2) Probabilistic framework, in which the probability idea is embedded. (3) Boost-like approaches, which decompose a single metric into a combination of several weak metrics for easier learning. (4) Advantageous variants. Some other algorithms are used to improve the performance of metric learning. (5) Specific applications. Metric learning has been applied in various particular tasks.

3.1. Pairwise cost

The basic idea in metric learning is to shrink intra-class distance and expand inter-class distance. The most direct way is to construct optimization problem based on the cost information of pairwise distance. For the given training set (2.1) and a defined metric M, pairwise distance can be classified into two categories, within-class distance,

$$d_M(x_i, x_j) = (x_i - x_j)^\top M(x_i - x_j), (x_i, x_j) \in S \tag{3.1}$$

which should be minimized for high cohesiveness, and between-class distance,

$$d_M(x_i, x_j) = (x_i - x_j)^\top M(x_i - x_j), (x_i, x_j) \in D \tag{3.2}$$

which should be maximized for high scatterness.

A generalized framework with pairwise cost can be presented as

$$\mathcal{J} = \lambda_1\mathcal{L}_1(d^W) + \lambda_2\mathcal{L}_2(d^B) + \lambda_3\mathcal{L}_3(d^W, d^B) \tag{3.3}$$

where d^W, d^B denote the total within-class distance and between-class distance respectively. And $\lambda_1, \lambda_2, \lambda_3$ are all non-negative trade-offs. In Eq. (3.3), the first and second term represent the absolute cost in terms of intra-class distance and inter-class distance severally. The third term measures the relative cost in terms of the weighted difference/ratio between intra-class distance and inter-class distance, which can be crystallized as $\mathcal{L}_3(d^W - Cd^B)$ or $\mathcal{L}_3(Cd^W/d^B)$ (C is a positive weight). The metric learning methods in the category of pairwise cost can be divided into two classes, absolute cost($\lambda_3 = 0$) and relative cost($\lambda_3 \neq 0$).

3.1.1. Absolute cost

Methods in this subclass focus on optimizing the absolute pairwise distance, give an upper bound for intra-class distance and/or a lower bound for inter-class distance or just minimize intra-class distance and/or maximize inter-class distance directly. One of the earliest method in metric learning is proposed by Xing, Jordan, Russell, and Ng (2002) in 2003, which formulates metric learning as a convex optimization problem using side-information. The method, termed as MLSI in this paper, aims to minimize within-class distance, subject to that the between-class is confined to be lower than a threshold. The problem is solved by an iterative projection algorithm, resulting in high computational complexity and time-consuming convergence. DML-eig (Ying & Li, 2012) is proposed to make improvements on MLSI, which maximize the minimal distances between dissimilar samples with within-class distance constrained by an upper bound. The primary problems can be converted into an equivalent formulation, presented as eigenvalue optimization.

Logdet-linear (Jain, Kulis, Davis, & Dhillon, 2012) and PCCA (Mignon & Jurie, 2012) both give an upper bound for distance of similar pairs and a lower bound for distance of dissimilar pairs. The difference between them is the cost function, logdet-linear aims to minimize the logdet divergence between the target metric M and a predefined metric M_0 while PCCA seeks to maximize the difference between each distance and its corresponding bound. DDML (Hu, Lu, & Tian, 2014) combines the idea of metric learning and deep learning, which looks for multiple nonlinear transformations by constructing multi-layer neural networks. The distance between similar points is required to be smaller than a predefined constant τ minus one and the distance between dissimilar points is forced to be larger than $\tau + 1$. Similar to PCCA, the method maximizes the total differences between each distance and its corresponding bound using logistic loss function. RCA (Bar-Hillel, Hertz, Shental, & Weinshall, 2005) directly minimizes the sum of distances between each point and its corresponding class center.

3.1.2. Relative cost

In view of kNN, the absolute values of intra-class distance and inter-class distance are not so important since correct prediction of an unlabeled sample is resulting from that similar point is nearer than dissimilar point to the sample. Briefly speaking, the fact that within-class distance is smaller than between-class distance is favorable to kNN. So it is promising to develop metric learning methods in consideration of relative distance. If $\lambda_3 \neq 0$ in Eq. (3.3), the weighted difference between intra-class distance and inter-class distance is an important term in pairwise cost. Minimizing relative cost enforces inter-class distance larger than intra-class distance as much as possible. The most directly way to learn metric using relative cost is let $\lambda_1 = \lambda_2 = 0$ and just to optimize $\mathcal{L}_3$. SMLP (Rosales & Fung, 2006), CMM (Wang, 2011), FrobMetric (Shen, Kim, Liu, Wang, & Van Den Hengel, 2014), DML (Hu, Lu, & Tan, 2016) and MLLS (Song, Xiang, Jegelka, & Savarese, 2016) are all proposed to maximize the difference between average inter-class distance and intra-class distance. A regularizer of Frobenius norm is placed in both SMLP and FrobMetric to learn a sparse metric. Linear programming method is applied in solving SMLP. An L_1-penalized log-determinant regularization is used in SDML (Qi, Tang, Zha, Chua, & Zhang, 2009) to learn an efficient sparse metric. Since any positive semi-definite matrix can be decomposed into $M = L^\top L$, the primary problem of CMM can be solved by eigenvalue decomposition by enforcing $LL^\top$ be equal to identity matrix. In FrobMetric, an efficient dual approach is derived to obtain a desired metric. DML learns a nonlinear distance metric by a neural network. Lifted structured feature embedding is implemented in MLLS for high ability of discrimination. RDC (Zheng, Gong, & Xiang, 2013) also minimize the relative cost $\mathcal{L}_3(d^W - Cd^B)$ individually, but with logistic loss function. To solve the optimization problem easier, the inner product of any two column vectors of the metric is forced to be zero. LRML (Hoi et al., 2008) puts the cost of unlabeled examples into the original relative cost to learn a metric from semi-supervised problems. The weighted difference between d^W and d^B can be put into constraints with cost variants in the objective function. By enforcing $d^W - d^B \geq t$ (t is a threshold), LDMRC (Schultz & Joachims, 2004) and SML (Ying, Huang, & Campbell, 2009) aims to minimize the norm of metric or its factor. But LMNN (Weinberger, Blitzer, & Saul, 2005) and LMCA (Torresani & Lee, 2006) just to optimize the intra-class distance, in which $\lambda_1 \neq 0$, $\lambda_2 = 0$. A variant of LMNN, termed as mLMNN, which was proposed in company with LMNN, learns multiple metrics for each class since a global linear transformation may not be sufficiently powerful to extract class-specific information. PLML (Wang, Kalousis, & Woznica, 2012) is a multi-metric version of LMNN, each metric corresponds to a local region. LMNLM (Chai, Liu, Chen, & Bao, 2010) is a variant of LMNN, enforcing each sample be near to its corresponding class centroid as much as possible and far away from other centroids in a unit distance. The principle of topology preserving is employed in RSSML (Wang, Yuen, & Feng, 2013) to obtain desired metric from

semi-supervised problems. The ratio of within-class distance and between-class distance can also be used as relative cost. DCA (Hoi et al., 2006) and MLPC (Baghshah & Shouraki, 2009) are two classical methods in adopting such ratio. But in MLPC, an extra goal is sought that the linear reconstruction error should be minimized. Both methods can be kernelized to extract nonlinear information from training data. Similar as MLPC, RMML (Lu, Wang, Deng, & Jia, 2015) is a reconstruction-based multimetric learning method, which aims to make intra-class reconstruction residual as small as possible and inter-class reconstruction residual as large as possible.

3.2. Probabilistic framework

Methods in probability framework construct optimization problems based on probability theory, using distance information. The method first defines probability distribution based on distance and metric, $p(x_i, x_j; M)$, and then establish objective function by Maximum Likelihood Estimate (MLE)

$$\max_M \quad p(x_i, x_j; M) \tag{3.4}$$

$$s.t. \quad M \succeq 0 \tag{3.5}$$

or match the distribution with a predefined and ideal distribution $p(x_i, x_j; M_0)$

$$\max_M \quad d_p(p(x_i, x_j; M), p(x_i, x_j; M_0)) \tag{3.6}$$

$$s.t. \quad M \succeq 0 \tag{3.7}$$

where M_0 is a predefined matrix and $d_p(\cdot)$ measures the difference between two distribution, one common function is KL divergence. The idea of MLE is used in NCA, KISSME, LDML, RS-KISS, LCA, LDM and SERAPH. And MCML and ITML try to match two probability distribution. NCA (Goldberger, Hinton, Roweis, & Salakhutdinov, 2004) defines probability distribution based on the idea that each point selects another point as a neighbor with some probability related with their distance. The probability of that each point is similar to its neighbor is maximized in NCA. The method is suffer from high computational complexity since the leave-one-out performance is optimized. NCA had been developed into two variants (Hong, Li, Jiang, & Tu, 2011): sparse version and mixed version. The trace norm of the metric is added in the sparse model. In the mixture model, divide and conquer approach is employed and multiple metrics are learned to fit different local regions. MCML (Globerson & Roweis, 2005) and LDM (Yang, Jin, Sukthankar, & Liu, 2006) define similar probability distribution as NCA based on the distance information of similar pairs and dissimilar pairs. MCML aims to minimize the Kullback–Leibler divergence between the defined distribution and the ideal 'bi-level' distribution, which tries to collapse similar points to a single point and push dissimilar points infinitely far away from each other. In LDM, maximum log-likelihood estimation is used and the bound optimization algorithm (Salakhutdinov & Roweis, 2003) is applied to solve the method. NCMML (Mensink, Verbeek, Perronnin, & Csurka, 2013) is proposed on the basis of multi-class logistic regression, using the distance between each example and its corresponding class centroid. The log-likelihood of the correct predictions is to be optimized to learn the metric. Gaussian-like probability distribution is adopted in ITML (Davis, Kulis, Jain, Sra, & Dhillon, 2007), KISSME (Koestinger, Hirzer, Wohlhart, Roth, & Bischof, 2012) and LCA (Der & Saul, 2012) to approximate the true distribution. In ITML, the relative entropy between two multivariate Gaussians, parametrized by the target metric and a predefined metric respectively, is minimized. And the constraints that the pairwise distances are restrained by absolute bounds are incorporated. KISSME constructs a log-likelihood ratio test from the view of statistical inference. The distance metric can be obtained by projection and eigenanalysis. RS-KISS (Tao et al., 2013) makes improvements on KISS, which combines regularization and smoothing techniques for robust matrices. LCA introduces latent variables into the probability distribution which can locate examples in a latent space with lower dimension. LDML (Guillaumin et al., 2009) defines the probability that two points are similar using sigmoid function, parametrized by the distance of the two points. The constructed linear discriminant model can be optimized by gradient descent. Similar as ITML, SERAPH (Niu, Dai, Yamada, & Sugiyama, 2014) is also a information-theoretic approach, in which the entropy of the probability on labeled data is maximized and on unlabeled data is minimized. BAYES (Yang, Jin, & Sukthankar, 2012) estimates a posterior distribution by a Bayesian framework. SCA (Changpinyo, Liu, & Sha, 2013) constructs a probabilistic graphical model.

3.3. Boost-like methods

In metric learning, the constraint of positive semidefinite on the target metric is a necessary condition for completeness. However, the constraint is hard to deal with, which often leads to intractable solution or very time-consuming algorithm. Boost-like methods try to decompose the data-dependent metric into a linear combination of sub-metrics with weak constraints, namely,

$$M = \sum_i \alpha_i M_i \tag{3.8}$$

where M_i is a matrix which can be learned in an easier way. Commonly, M_i is restricted to be a trace-one rank-one matrix. The target metric can be regarded as a strong learner and the sub-metrics are weak learners. The strong learner can be generated by adding weak learners to it incrementally. BoostMetric (Shen, Kim, Wang, & Hengel, 2009), DRMetric (Liu & Vemuri, 2012) and MetricBoost (Bi et al., 2011) decompose a positive semidefinite matrix into a convex combination of rank-one trace-one positive semidefinite matrices and employ the idea that distance between dissimilar pairs should be larger than distance between similar pairs as much as possible. BoostMetric constructs a nonlinear problem with exponential loss and a trace regularized term. Coordinate descent optimization is applied to solve the Lagrange dual problem after the base metrics be learned by eigen-decomposition. DRMetric aims to maximize a soft margin in linear formulation and regularizer variants can be added into the primary problem or the dual problem for the solutions. MetricBoost minimizes the overall error rate characterized by the give distribution of triplets and the indicator function with respect to relative distance information over the triplets. A bipartite strategy is employed to reduce computation cost in the method. REMetric (Kozakaya, Ito, & Kubota, 2011) is proposed based on ensemble learning scheme, which optimize local objective function for sampled training data. The whole training data is randomly subsampled and multiple discriminative projection vectors are learned from linear support vector machines. The target metric is obtained by combining these vectors. BoostMDM (Chang, 2012) defines a leave-one-out error with θ-clipped loss function and learns the metric by adding base learners to the metric iteratively. The EM algorithm (Moon, 1996) is applied to get the solution.

3.4. Advantageous variants

In this section, we will introduce diverse variants in metric learning, which combine metric learning with other learning framework, to improve the performance of classification models that only the technique of metric learning used. POLA (Shalev-Shwartz, Singer, & Ng, 2004), LEGO (Jain, Kulis, Dhillon, & Grauman,

2009), SOML (Gao, Hoi, Zhang, Wan, & Li, 2014) and MDML (Kunapuli & Shavlik, 2012) all learn distance metric by online learning, where constraints are available incrementally and the metric can be obtained with an updating rule. For the time t, model receives several samples $x_{t1}, \ldots, x_{tm}$, the metric can be update by

$$M_t = M_{t-1} + Q(x_{t1}, \ldots, x_{tm}; M_{t-1}) \tag{3.9}$$

where Q is a matrix generated by $x_{t1}, \ldots, x_{tm}$ and M_{t-1}. The model with online learning have several advantages: (1) they can deal with large-scale problem; (2) the updation can ensure the property of positive semidefinite; (3) for the problem with data stream, the metric can be updated timely. So the combination of metric learning and online learning obtains competitive performance in classification. POLA iteratively receives pairs of instances and computes their similarity using a pseudo-metric, then updates the pseudo-metric by successive projections onto the positive semidefinite cone and onto half-space constraints imposed by the received examples. LEGO updates a target metric based on LogDet regularization and gradient descent. The method performs well in both online case and offline counterpart. SOML learns a sparse distance function, updated by online gradient descent, to deal with high-dimensional image data. The advantages of sparsity and high computational efficiency make the model be competitive to other methods. MDML is an online regularized metric learning, updating metric based on the framework of composite objective mirror descent. It is scalable to large-scale datasets and kernelizable to nonlinear metric learning. Structural learning is combined with metric learning in MLR (McFee & Lanckriet, 2010) and R-MLR (Lim, Lanckriet, & McFee, 2013), optimizes W to minimize a ranking loss function $\Delta : Y \times Y \rightarrow R$ over permutations Y induced by distance. Inspired by structural SVM (Joachims, Finley, & Yu, 2009; Yu & Joachims, 2009), MLR casts the problem of prediction by nearest neighbor as a ranking problem and regard the error rate on predicted label as a loss function. R-MLR is a robust extension to MLR, enforcing group sparsity on the learned metric. It can identify informative features when a large proportion redundant features exist. To extract nonlinear information from input features, nonlinear transformation is often applied directly or in kernel formulation to map original features into a new space. GB-LMNN (Kedem, Tyree, Sha, Lanckriet, & Weinberger, 2012) and DNLML (Cai et al., 2012) both introduce nonlinear mapping directly to learn a metric. GB-LMNN applies gradient boosting to learn a nonlinear transformation, instead of traditional linear transformation, to shrink intra-class distance and expand inter-class distance in the mapped space. The approach enjoys the advantages of robustness, speed and insensitivity. DNLML constructs a multi-layer neural network and make nonlinear transformation by nonlinear activate function. In the transformed space, the probability of that two points are similar or not is optimized. MLKR (Weinberger & Tesauro, 2007), ML-MKL (Wang, Do, Woznica, & Kalousis, 2011), KDML (He, Chen, Chen, & Mao, 2013), MKMLR (Galleguillos, McFee, & Lanckriet, 2014) and EWFC (Wang, Deng, Choi, Jiang, Luo, Chung, & Wang, 2016) introduce kernel function to metric learning to learn metric from kernelized data. The method introduces nonlinear mapping φ and learns metric M(or linear transformation L) in the new space, the distance between x_i, x_j can be denoted as

$$d(x_i, x_j) = (\varphi(x_i) - \varphi(x_j))^\top M(\varphi(x_i) - \varphi(x_j)). \tag{3.10}$$

If M can be parametrized as $M(\varphi)$, the kernel function $K(x_i, x_j) = \varphi(x_i)^\top \varphi(x_j)$ can be introduced. So kernel learning will be embedded in metric learning for better performance. MLKR learns a distance metric for a distance-based kernel function, to improve the performance of nonlinear regression. The leave-one-out regression error is minimized to obtained the optimal metric. Kernel density estimation is applied in KDML to make nonlinear mapping, and the distance metric is learned from the new space. KDML can deal with not only numerical features bust also categorical ones. Metric learning with multiple kernels is implemented in ML-MKL since a linear transformation in traditional metric learning is not always appropriate and predefined kernel limits the expressiveness of the method. MKMLR employs multiple kernel learning and metric learning to integrate heterogeneous features in a nearest neighbor setting effectively. EWFC (Wang et al., 2016) combines soft subspace clustering and metric learning in a composite kernel space. MPCK-means (Bilenko, Basu, & Mooney, 2004) first divides the whole dataset into several clusters by K-means and then learn a metric for each cluster by minimizing cluster dispersion. The local metrics can extract more information to obtain better performance. MDM (Yu, Jiang, & Zhang, 2011) is aiming to maximize the minimal between-class distance, which is measured between the two class centroids. MLMNP (Sohn, 2016) constructs multi-class N-pair loss to address the problem of slow convergence that appeared in contrastive loss and triplet loss. Some other extensions on metric learning have been proposed, including graph learning (JLLDM (Liu, Wang, Hong, Zha, & Hua, 2010)), hashing learning (HDML (Norouzi, Fleet, & Salakhutdinov, 2012)), similarity learning (Sub-SML (Cao et al., 2013)), particular distance function (P/S-SDML (Zhu, Zhang, Zuo, & Zhang, 2013)), new empirical risk function (LLML (Bian & Tao, 2011)).

3.5. Specific applications

Due to the strong ability in measuring similarity, metric learning has been applied in different specific applications, including multi-instance problem, multi-view problem, multi-task problem, multi-label problem, transfer learning, for better performance.

Multi-instance problem is a particular kind of classification task, in which each example is presented as a bag, a set with multiple instances. A bag is labeled as positive if at least one instance is positive, otherwise negative. The similarity between bags is an important issue in multi-instance learning. Learning a data-dependent bag distance function may improve the performance of the state-of-the-art methods. MildML (Guillaumin et al., 2010), MIML (Xu et al., 2011) and MLMIML (Jin et al., 2009b) all introduce metric learning into multi-instance problem. For multi-view learning, each training sample is provided with several views, the information from which is different but may be complementary. It is a challenge to integrate different views to improve learning performance. Since metric learning can be used to make similarity measurement with a data-dependent metric on a single view, it is tractable to learn multiple metrics, each metric corresponds to a single view due to heterogeneous features in different views. Several methods make efforts in multi-view metric learning, including LM^{3}L (Hu, Lu, Yuan et al., 2014), SSM-DML (Yu et al., 2012), HMML (Zhang et al., 2013). Multi-task problem is a kind of joint learning, multiple tasks are learned in parallel but interact with each other. Traditional learning paradigm in machine learning is a single task learning, with only one model be output. Multi-task learning, often originated from a complex problem, which cannot be solved by a single model, constructs multiple models and combine these models to process the source problem. For each subtask, a desired metric can be learned but the connection between different metrics should be considered. mt-LMNN (Parameswaran & Weinberger, 2010), mt-von (Yang et al., 2013) and mtMCML (Ma et al., 2014) can deal with such problem. Multi-label problem is a classification task where each example is labeled by more than one label, and every two examples may be have both similar and dissimilar labels. The similarity between two examples in such problem is not determined, which give a challenge to metric learning. Three methods, MLMIML (Jin et al., 2009b), LC2I-L1 (Wang, Gao et al., 2012), LM-kNN (Liu & Tsang, 2015), have tried to solve multi-label classification by learning a metric. Transfer learning aims to

Table 2
Characters of metric learning methods.

Category	Algorithm	Property taxonomy								
		Publication	L/G	L/N	DR	R/A	RT	SP	KE	CR
Pairwise cost	MLSI (Xing et al., 2002)	2003NIPS	Global	Linear	No	Absolute	No	Semi-supervised	No	1991/13
	LDMRC (Schultz & Joachims, 2004)	2003NIPS	Global	Linear	No	Relative	Yes	Supervised	No	372/13
	RCA (Bar-Hillel et al., 2005)	2005JMLR	Global	Linear	Yes	Absolute	No	Semi-supervised	No	483/11
	LMNN (Weinberger et al., 2005)	2005NIPS	Local	Linear	Yes	Relative	No	Supervised	No	1029/11
	SMLP (Rosales & Fung, 2006)	2006KDD	Global	Linear	Yes	Relative	Yes	Supervised	No	57/10
	LMCA (Torresani & Lee, 2006)	2006NIPS	Local	Linear	Yes	Relative	No	Supervised	Yes	150/10
	DCA (Hoi et al., 2006)	2006CVPR	Global	Linear	No	Relative	No	Semi-supervised	Yes	219/10
	LRML (Hoi et al., 2008)	2008CVPR	Both	Linear	No	Relative	No	Semi-supervised	No	98/8
	MLPC (Baghshah & Shouraki, 2009)	2009IJCAI	Global	Linear	No	Relative	No	Semi-supervised	Yes	67/7
	SML (Ying et al., 2009)	2009NIPS	Global	Linear	Yes	Relative	Yes	Supervised	No	64/7
	online-reg (Jin, Wang, & Zhou, 2009a)	2009NIPS	Global	Linear	No	Absolute	Yes	Supervised	No	81/7
	SDML (Qi et al., 2009)	2009ICML	Global	Linear	No	Absolute	Yes	Supervised	No	52/7
	LMNLM (Chai et al., 2010)	2010SP	Global	Linear	No	Relative	Yes	Supervised	No	18/6
	SPML (Shaw, Huang, & Jebara, 2011)	2011NIPS	Global	Linear	No	Relative	Yes	Supervised	No	34/5
	CMM (Wang, 2011)	2011TCYB	Global	Linear	No	Relative	No	Semi-supervised	Yes	37/5
	logdet-linear (Jain et al., 2012)	2011JMLR	Global	Linear	No	Absolute	No	Supervised	Yes	94/4
	DML-eig (Ying & Li, 2012)	2012JMLR	Global	Linear	No	Absolute	No	Supervised	No	128/4
	PCCA (Mignon & Jurie, 2012)	2012CVPR	Global	Linear	No	Absolute	No	Semi-supervised	Yes	196/4
	PLML (Wang, Kalousis et al., 2012)	2012NIPS	Local	Linear	No	Relative	Yes	Supervised	No	40/4
	RSSML (Wang et al., 2013)	2013PR	Global	Linear	No	Relative	Yes	Semi-supervised	No	17/3
	RDC (Zheng et al., 2013)	2013PAMI	Global	Linear	No	Relative	No	Supervised	No	194/3
	FrobMetric (Shen et al., 2014)	2013TNN	Global	Linear	No	Relative	Yes	Supervised	No	11/2
	DDML (Hu, Lu, & Tian, 2014)	2014CVPR	Global	Nonlinear	No	Absolute	Yes	Supervised	No	90/2
	RMML (Lu et al., 2015)	2015TIFS	Global	Linear	No	Relative	Yes	Supervised	No	12/1
	DML (Hu et al., 2016)	2016TCSVT	Global	Nonlinear	No	Relative	Yes	Supervised	No	4/1
	MLLS (Song et al., 2016)	2016CVPR	Global	Nonlinear	No	Relative	No	Supervised	No	61/1
Probabilistic framework	NCA (Goldberger et al., 2004)	2004NIPS	Local	Linear	Yes	Absolute	No	Supervised	No	960/12
	MCML (Globerson & Roweis, 2005)	2005NIPS	Global	Linear	Yes	Absolute	No	Supervised	Yes	514/11
	LDM (Yang et al., 2006)	2006AAAI	Local	Linear	No	Absolute	No	Supervised	No	134/10
	ITML (Davis et al., 2007)	2007ICML	Global	Linear	No	Absolute	No	Both	Yes	987/9
	LDML (Guillaumin et al., 2009)	2009ICCV	Global	Linear	No	Absolute	No	Supervised	No	407/7
	ms-NCA (Hong et al., 2011)	2011ICCV	Global	Linear	Yes	Absolute	Yes	Supervised	No	20/5
	KISSME (Koestinger et al., 2012)	2012CVPR	Global	Linear	No	Absolute	No	Semi-supervised	No	264/4
	LCA (Der & Saul, 2012)	2012NIPS	Global	Linear	Yes	Absolute	No	Supervised	No	10/4
	BAYES (Yang et al., 2012)	2012UAI	Global	Linear	No	Absolute	No	Supervised	No	43/4
	RS-KISS (Tao et al., 2013)	2013TCSVT	Global	Linear	No	Relative	Yes	Supervised	No	61/3
	NCMML (Mensink et al., 2013)	2013PAMI	Global	Linear	No	Absolute	No	Supervised	No	40/3
	SCA (Changpinyo et al., 2013)	2013NIPS	Global	Linear	No	Absolute	No	Supervised	No	5/3
	SERAPH (Niu et al., 2014)	2014NC	Global	Linear	No	Absolute	Yes	Semi-supervised	No	25/2
Boost-like methods	BoostMetric (Shen et al., 2009)	2009NIPS	Global	Linear	No	Relative	Yes	Supervised	No	70/7
	MetricBoost (Bi et al., 2011)	2011CVPR	Global	Linear	No	Relative	No	Supervised	No	21/5
	REMetric (Kozakaya et al., 2011)	2011ICCV	Local	Linear	Yes	Absolute	Yes	Supervised	No	15/5
	BoostMDM (Chang, 2012)	2011PR	Globsl	Linear	No	Relative	No	Supervised	No	13/4
	DRMetric (Liu & Vemuri, 2012)	2013ECCV	Global	Linear	No	Relative	Yes	Supervised	No	12/4
Advantageous variants	POLA (Shalev-Shwartz et al., 2004)	2004ICML	Global	Linear	No	Absolute	No	Supervised	Yes	251/12
	MPCK-means (Bilenko et al., 2004)	2004ICML	Global	Linear	No	Absolute	Yes	Semi-supervised	No	685/12
	MLKR (Weinberger & Tesauro, 2007)	2007JMLR	Global	Linear	Yes	Absolute	No	Supervised	Yes	72/9
	LEGO (Jain et al., 2009)	2009NIPS	Global	Linear	No	Absolute	No	Both	No	119/7
	MLR (McFee & Lanckriet, 2010)	2010ICML	Global	Nonlinear	No	Relative	Yes	Supervised	No	163/6
	R-MLR (Lim et al., 2013)	2013JMLR	Global	Nonlinear	No	Relative	Yes	Supervised	No	39/3
	JLLDM (Liu et al., 2010)	2010TCYB	Global	Linear	No	Absolute	Yes	Semi-supervised	No	21/6
	ML-MKL (Wang et al., 2011)	2011NIPS	Global	Nonlinear	No	Absolute	Yes	Supervised	Yes	39/5
	LLML (Bian & Tao, 2011)	2011IJCAI	Global	Linear	No	Absolute	No	Semi-supervised	No	25/5
	MDM (Yu et al., 2011)	2011PR	Global	Linear	No	Absolute	Yes	Supervised	No	9/7
	HDML (Norouzi et al., 2012)	2012NIPS	Global	Nonlinear	Yes	Relative	Yes	Supervised	No	95/4
	GB-LMNN (Kedem et al., 2012)	2012NIPS	Local	Nonlinear	Yes	Relative	No	Supervised	No	70/4
	DNLML (Cai et al., 2012)	2012ACMMM	Global	Linear	No	Absolute	No	Semi-supervised	No	22/4
	MDML (Kunapuli & Shavlik, 2012)	2012ECML	Global	Linear	No	Absolute	Yes	Supervised	Yes	13/4
	SOML (Gao et al., 2014)	2014AAAI	Global	Linear	No	Relative	Yes	Supervised	Yes	19/2
	Sub-SML (Cao et al., 2013)	2013ICCV	Global	Linear	No	Absolute	Yes	Supervised	No	62/3
	P/S-SDML (Zhu et al., 2013)	2013ICCV	Global	Linear	No	Absolute	Yes	Supervised	No	24/3
	KDML (He et al., 2013)	2013ICDM	Local	Nonlinear	No	Relative	No	Supervised	Yes	13/3
	MKMLR (Galleguillos et al., 2014)	2014IJCV	Global	Linear	No	Absolute	Yes	Supervised	Yes	7/2
	MLMNP (Sohn, 2016)	2016NIPS	Global	Nonlinear	No	Relative	Yes	Supervised	No	0/1
	EWFC (Wang et al., 2016)	2016PR	Local	Nonlinear	No	Absolute	Yes	Unsupervised	Yes	7/1

(*continued on next page*)

predict target domain data by the model trained by source domain data. Metric learning can be applied in this field by transferring a generic metric from source domain to a specific metric in target domain. The information carried by the generic metric can be used to strengthen the ability of the specific metric. TML (Li et al., 2012), M^{2}SL (Wang, Jiang, et al., 2012) and CDML (Wang et al., 2014) make transfer metric learning with competitive performance. Metric learning is applied in other applications, including structured

Table 2 (*continued*)

Category	Algorithm	Property taxonomy								
		Publication	L/G	L/N	DR	R/A	RT	SP	KE	CR
	HDLR (Davis & Dhillon, 2008)	2008KDD	Global	Linear	No	Absolute	No	Supervised	No	49/8
	MLMIML (Jin, Wang, & Zhou, 2009b)	2009CVPR	Global	Linear	No	Absolute	No	Supervised	No	53/7
	MildML (Guillaumin, Verbeek, & Schmid, 2010)	2010ECCV	Global	Linear	No	Absolute	No	Supervised	No	82/6
	mt-LMNN (Parameswaran & Weinberger, 2010)	2010NIPS	Local	Linear	No	Relative	Yes	Supervised	No	130/6
	USML (Cinbis, Verbeek, & Schmid, 2011)	2011ICCV	Global	Linear	No	Absolute	No	Unsupervised	No	70/5
	MIML (Xu, Ping, & Campbell, 2011)	2011ICDM	Global	Linear	No	Absolute	No	Semi-supervised	No	13/5
	AggkNN (Verma, Mahajan, Sellamanickam, & Nair, 2012)	2012CVPR	Global	Linear	No	Absolute	Yes	Supervised	No	42/4
Specific applications	TML (Li, Zhao, & Wang, 2012)	2012ACCV	Global	Linear	No	Relative	Yes	Supervised	No	110/4
	LC2I-L1 (Wang, Gao, & Chia, 2012)	2012ECCV	Global	Linear	No	Relative	Yes	Supervised	No	7/4
	2PKNN (Verma & Jawahar, 2012)	2012ECCV	Local	Linear	No	Relative	No	Supervised	No	55/4
	mt-von (Yang, Huang, & Liu, 2013)	2013ML	Global	Linear	No	Absolute	Yes	Semi-supervised	No	14/3
	LM^3L (Hu, Lu, Yuan, & Tan, 2014)	2014ACCV	Global	Linear	No	Absolute	Yes	Supervised	No	20/2
	mtMCML (Ma et al., 2014)	2014TIP	Global	Linear	No	Absolute	Yes	Supervised	No	42/2
	3DML (Gao et al., 2014)	2014TIE	Global	Linear	No	Absolute	No	Supervised	No	104/2
	SSM-DML (Yu, Wang, & Tao, 2012)	2012TIP	Global	Linear	No	Absolute	Yes	Semi-supervised	No	130/4
	NRML (Lu, Zhou, Tan, Shang, & Zhou, 2014)	2014PAMI	Local	Linear	No	Absolute	No	Supervised	No	81/2
	M^2SL (Wang, Jiang, Huang, & Tian, 2012)	2012CVPR	Global	Linear	No	Relative	Yes	Supervised	Yes	13/4
	HMML (Zhang, Zhang, Nasrabadi, & Huang, 2013)	2013IF	Local	Linear	No	Relative	No	Supervised	Yes	9/3
	LMKML (Lu et al., 2013)	2013ICCV	Global	Nonlinear	No	Relative	No	Supervised	Yes	44/3
	CDML (Wang, Wang, Zhang, & Xu, 2014)	2014AAAI	Global	Linear	No	Absolute	No	Semi-supervised	No	1/2
	Bag-SVRML (Zou, Wang, Chen, & Chen, 2014)	2014KBS	Global	Linear	No	Absolute	Yes	Supervised	No	2/2
	MLPP (Lajugie, Arlot, & Bach, 2014)	2014ICML	Global	Linear	No	Absolute	Yes	Unsupervised	No	7/2
	LM-kNN (Liu & Tsang, 2015)	2015AI	Global	Linear	No	Relative	Yes	Supervised	No	3/1
	DeepMDML (Yu, Yang, Gao, & Tao, 2016)	2017TCYB	Global	Linear	No	Absolute	No	Supervised	No	17/1

problem (HDLR (Davis & Dhillon, 2008)), unsupervised problem (USML (Cinbis et al., 2011)), image annotation (2PKNN (Verma & Jawahar, 2012)), 3D object detection (3DML (Gao et al., 2014)), image set classification (LMKML (Lu et al., 2013)), regression (Bag-SVRML (Zou et al., 2014)), partition problem (MLPP (Lajugie et al., 2014)), taxonomy specific problem (AggkNN (Verma et al., 2012)) and image ranking (DeepMDML (Yu et al., 2016)). Next we will give a summary of the above mentioned metric learning methods with a property taxonomy:

- L/G (Local or Global): the constraints are constructed on local or global view;
- L/N (Linear or Nonlinear): linear means the method learns metric M or transformation L, nonlinear means a nonlinear transformation φ is used or to be learned in the method;
- DR (Dimension Reduction): whether the method can make dimension reduction;
- R/A (Relative distance or Absolute distance): the optimization problem is constrained by relative distance or absolute distance;
- RT (Regularized Term): whether there is a regularized term in the objective function;
- SP (Supervised Pattern): the method is proposed for supervised problem, semi-supervised problem or unsupervised problem;
- KE (Kernel Extension): whether kernel extension is made on the method;
- CR (Citation Ratio): the ratio between citation number and the year number that it has been published (2017 minus its publication year).

All the metric learning methods referred in this paper have been summarized in chronological order in Table 2.

4. Experiments

In this section, numerical experiments will be made to compare metric learning methods comprehensively. We will first select classical and influential algorithms and then design different experiments on representative datasets. The experimental results will be analyzed to compare model performance and evaluate the ability of the selected methods in learning informative metric.

4.1. Algorithms selected and settings

We will select representative methods in each category to make comparisons, in consideration of their citation ratio and the representativeness according to our understanding. For the category of Pairwise Cost, three methods, **MLSI**, **DML-eig** and **PCCA** in absolute cost are considered and two methods, **LMNN** and **SPML**, in relative cost are selected. MLSI and LMNN are two of the most popular metric learning methods and they have been cited for more than one thousand times. DML-eig is a particular method solved by eigenvalue decomposition, resulting in low computational complexity and competitive performance. PCCA constructs an unconstrained optimization problem with respect to a linear transformation. SPML learns a metric from a network with structure preserved. **mLMNN** will be selected due to its typicalness and advantages in extracting class-specific information. In the field of Probabilistic Framework, seven methods are considered: **NCA**, **MCML**, **ITML**, **LDML**, **KISSME**, **NCMML**, **SERAPH**. NCA and MCML are two of the earliest methods with probabilistic framework. ITML is proposed with low training time based on information theory. LDML and KISSME are two methods with concise optimization formulation and they are both presented with clear task, for face identification and large-scale problem respectively. Regarding the Boost-like methods, **BoostMetric** and **MetricBoost** are selected

Table 3
Statistics of the benchmark datasets.

Dataset	Instance	Attribute	Class	Class distribution
Arrhythmia(AR)	452	279	2	{245,207}
Bupa liver(BU)	345	6	2	{145,200}
CMC(CM)	1473	9	2	{629,944}
Dermatology(DE)	366	34	2	{173,193}
Heart(HE)	270	13	2	{150,120}
Hepatitis(HP)	155	19	2	{32,123}
Ionosphere(IO)	351	34	2	{225,126}
Libras(LI)	360	90	15	≈24 per class
Parkinson(PA)	1040	26	2	{520,520}
Seeds(SE)	210	7	3	≈70 per class
Sonar(SO)	208	60	2	{97,111}
Spectf(SP)	267	44	2	{212,55}
Thyroid(TH)	215	5	3	≈72 per class
Vehiabc(VA)	282	18	4	≈70 per class
Vehicle(VE)	846	18	4	≈212 per class
Votes(VT)	435	16	2	{168,267}
Wine(WI)	178	13	3	≈60 per class
WPBC(WP)	198	33	2	{151,47}

since they have the top citation rates. For the methods in Advantageous Variants, three algorithms are chosen: **MLR**, **GB-LMNN** and **Sub-SML**. kNN classification with Euclidean distance(**Eucl**) is used as the baseline method. So there are 19 methods in total will be evaluated and compared with each other.

The algorithm settings are given as follows. In MLSI, the maximal iteration is 10. In DML-eig, β and μ are set to be 10^{-4}, 10^{-3} respectively. For PCCA, the parameter β of the logistic loss function is set to be 1 and the learning rate $\eta = 0.01$. The size of neighborhood K is 25 in LMNN and the regularization parameter λ is 10^{-6} in SPML. For mLMNN, the maximal iteration is 200. In ITML, the slack parameter γ is 0.1. The parameter ν is set to be 10^{-7} in BoostMetric.

4.2. Experimental design and datasets

In this section, we will make three sides experiments to evaluate the ability of these methods. First, a toy example is given to verify that the selected methods make efforts in mapping original examples into a new space to improve the performance of Eucl. Two artificial datasets, Three1 and Moon, are used to test the mapping ability. The two datasets are generated by Python 3.6 using sklearn.datasets library. Both datasets contain 500 instances with three classes. For Eucl, kNN classification is implemented with $k = 3$. For all the other methods, a desired metric is learned with default setting and then applied in 3NN classification.

Second, metric learning on benchmark datasets will be made to compare the ability in improving the classification performance of Eucl. We have selected 18 benchmark datasets, which are widely used in classification, from UCI repository (http://archive.ics.uci.edu/ml/index.php) and LIBSVM website (https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/). The statistics of these datasets are shown in Table 3, including the number of instances, attributes, classes and the number of instances in each class. The name of each dataset is abbreviated to two letters.

Third, we will explore the ability of the methods in changing intra-class distance and inter-class distance and its relationship to the neighbor size of kNN classification. Two famous image datasets, Corel and Caltech, are selected to implement experiments. Corel contains 500 images from ten classes, architecture, bus, dinosaur, elephant, face, flower, food, horse, sky and snowberg. The size of every image is 384×256 or 256×384. Caltech consists of 900 images from six classes, airplane, car, face, leave, motorbike and background. Fig. 2 shows some images from Corel and Caltech, each image corresponds to a class. LBP feature (Ojala, Pietikäinen, & Mäenpää, 2002) is extracted from the image set and the length of each extracted feature vector is 4096. The technique of PCA (Jolliffe, 1986) is implemented to preprocess the feature vectors for lower dimension and less training time. The dimension of each image is reduced to 300.

4.3. Model performance comparison

In the first experiment, the transformed points of Three1 and Moon have been depicted in Figs. 3 and 4 respectively. The number in the parenthesis of the caption of each subfigure denotes the classification accuracy. From the scatter diagrams, it can be seen that all the metric learning methods can change the data distribution, though the same classification accuracy from some methods, resulting from that data information is truly extracted by the methods. But unfortunately, only 11 and 7 methods perform better than Eucl on Three1 and Moon respectively, which demonstrates than not all the methods can always obtain informative metric. It may be explained by that not all the examples can strictly meet the constraints, pushing dissimilar points away may make them nearer to other dissimilar points. Certainly, such case depends on method and dataset.

The second experiment is made to compare the selected methods comprehensively. Three indexes are used to make evaluation:

1. The prediction accuracy

$$\mathrm{ACC} = \sum_{i=1}^{m} \mathbb{1}\{\tilde{y}_i = y_i\}/m$$

where $\tilde{y}_i, y_i$ are the predicted label and true label of the ith test point respectively and m is the size of test set.
2. AUC, the area under the curve of Receiver Operating Characteristic , which is usually used to evaluate a binary classifier. The larger the value of AUC, the better the classifier. AUC is insensitive to the case of class imbalance.
3. The training time TM, which is used to shown the computational efficiency of a method.

All the experimental results are given in Table 4. For each method, the three rows denote ACC, AUC and TM respectively. Each value is obtained by computing the mean of the results of ten times random partition. In each partition, 70% points are selected randomly for training and the left are used as the test set. And the value in subscript is the standard deviation of the ten results. The best value of each dataset is in boldface and the second is marked by an underline. For ACC, GB-LMNN obtains five best results and LMNN and KISSME both perform best on three datasets. For AUC, MCML and GB-LMNN both get largest AUC on three datasets. For TM, ITML runs fastest on nearly all the datasets except Arrhythmia. The results verifies that nonlinear metric learning has a greater advantage than linear metric learning in extracting useful metric. However, nonlinear metric learning often need more time to obtain data dependent information. The ROC curves of six datasets are depicted in Fig. 5. Most of the curves have similar trend and there are few differences among them.

Metrics visualization of CMC and Wine are made to show the difference between Euclidean metric and the learned metric. The metrics learned from CMC and Wine are displayed in Figs. 6 and 7 respectively. Each metric is denoted by a grid map with the same number of rows and columns as the metric. Each grid denotes the value in the corresponding location of the matrix. A colorbar is place on the right of every map and the corresponding value become larger and larger from bottom to the top. The map can reflect the relative difference among all the entries of the metric. In Fig. 6, the metrics learned from DML-eig, MCML, KISSME, BoostMetric perform better than Euclidean metric on CMC dataset. And they have one common feature, the value in the fourth column and

(a) Corel

(b) Caltech

Fig. 2. Images of Corel and Caltech.

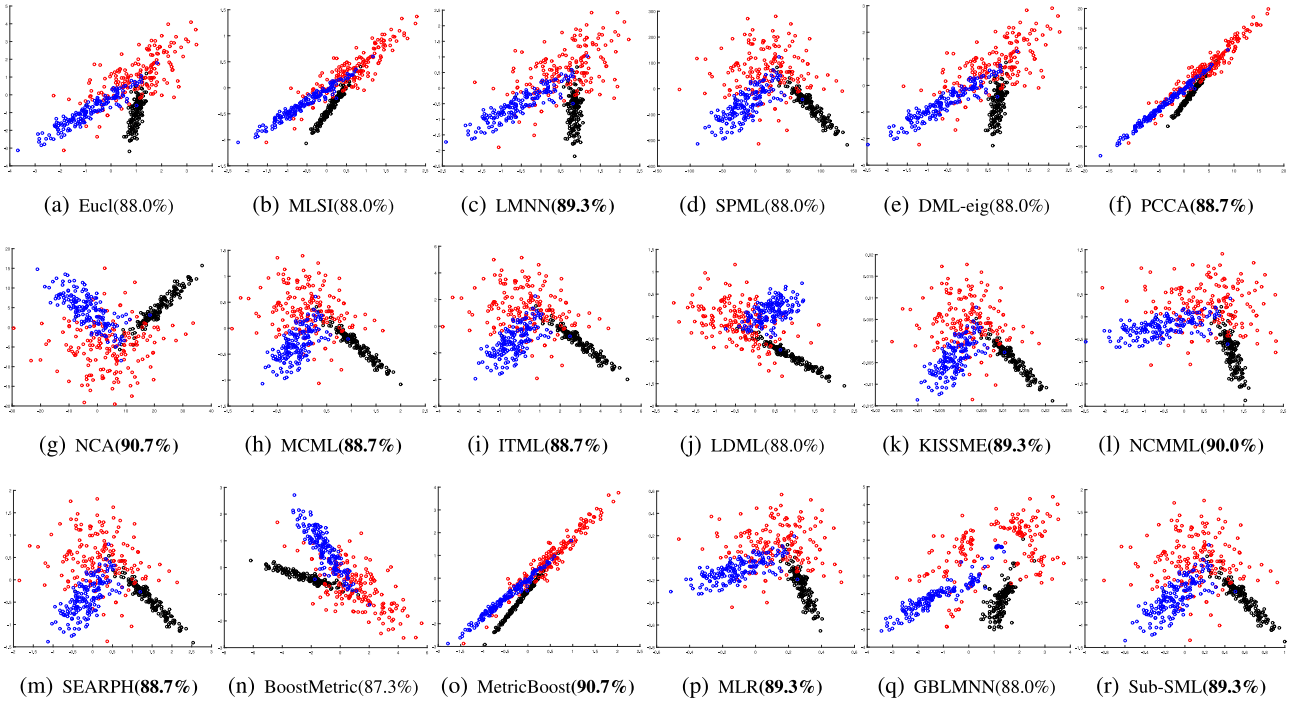


Fig. 3. Scattered points of Three1. The original points are displayed in (a) and the transformed points with corresponding transformation are displayed in (b)–(r) respectively.

fourth row is much bigger than its neighbors. It shows that these methods can extract similar information from the dataset. In Fig. 7, LMNN, SPML, DML-eig, NCA, MCML and SERAPH perform better than Eucl on Wine dataset. Similarly, the metrics of LMNN, SPML, DML-eig and MCML share a similar character that the value in the seventh column and seventh row is bigger than its neighbors. The character is not exist in NCA and SERAPH, which can be illustrated by that an informative metric can be presented as diverse formations.

To clearly explore the statistical difference between different methods with respect to ACC and AUC, Wilcoxon test is applied to compare each pair of algorithms. A significance level of $\alpha = 0.1$ is used. For each dataset, pairwise comparison will be made and a method will get 1 score if it is significantly better than the other one and get 0.5 score if there is no statistical difference between the two methods. The numbers of win, draw and lose are given in Table 5. The methods are listed according to the order of total score. The average TMs are sorted from low to high and the ranks are shown in the last column of Table 5. It can be seen that GB-LMNN, MCML and SPML performs the first, second and third respectively on both ACC and AUC. But the three methods are all implemented with high computational complexity. BoostMetric, LMNN and SERAPH obtain almost the same performance, all rank among top seven in ACC and AUC. The time ranks of BoostMetric, LMNN and SERAPH are fourth, eighth and ninth respectively. The seven methods, PCCA, mLMNN, MLSI, MetricBoost, Sub-SML, MLR and LDML, perform worse than Eucl, verifies their weak ability in learning informative metrics from benchmark datasets. It is worthy to point out that the top four methods with respect to ACC are belong to different categories, which verifies that learning metric from different views can always reach ideal goal. For the category of pairwise cost, LMNN and SPML perform better than DML-eig, PCCA and MLSI, shown that learning metric using relative distance cost is prior to absolute distance cost. For the 11 methods that perform better than Eucl on ACC, there are 6 ones belong to the category of probabilistic framework. Using distance information to obtain favorable probability distribution can learn better metric easier. BoostMetric and MetricBoost performs much better and much worse than Eucl respectively, proving that metric decomposition should be implemented restrainedly. GBLMNN performs the

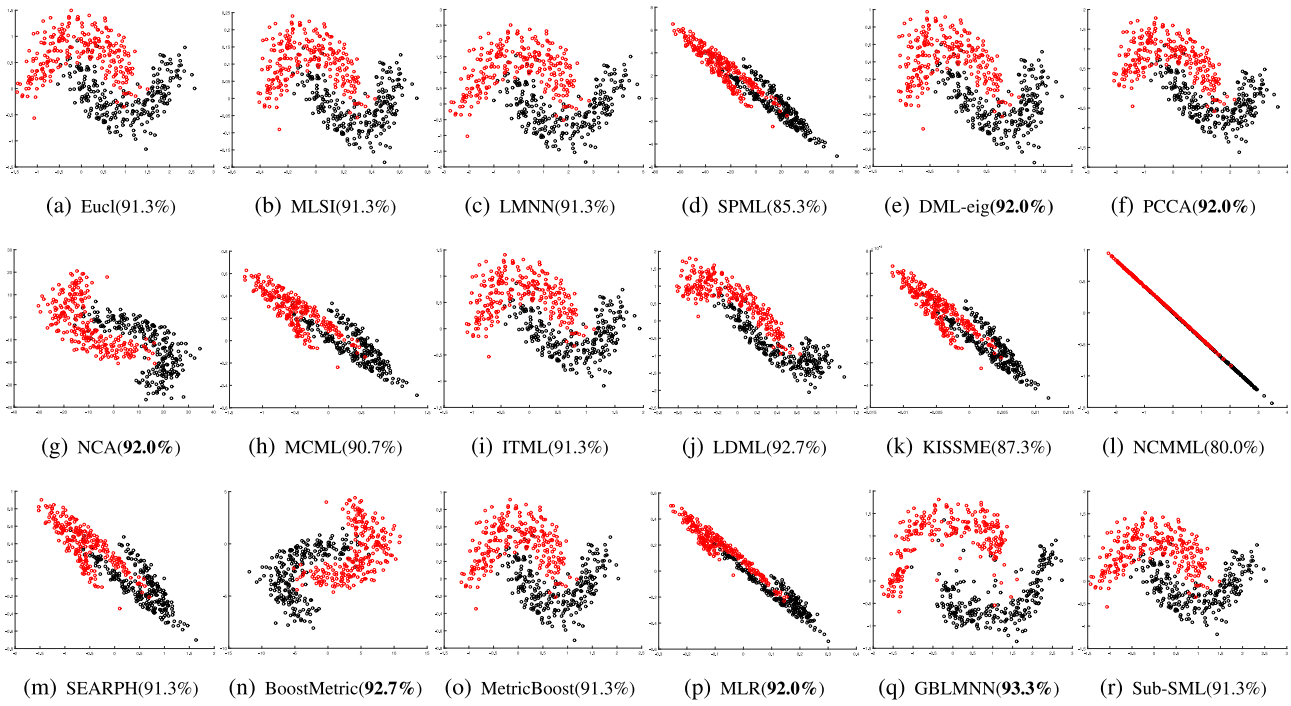


Fig. 4. Scattered points of Moon. The original points are displayed in (a) and the transformed points with corresponding transformation are displayed in (b)–(r) respectively.

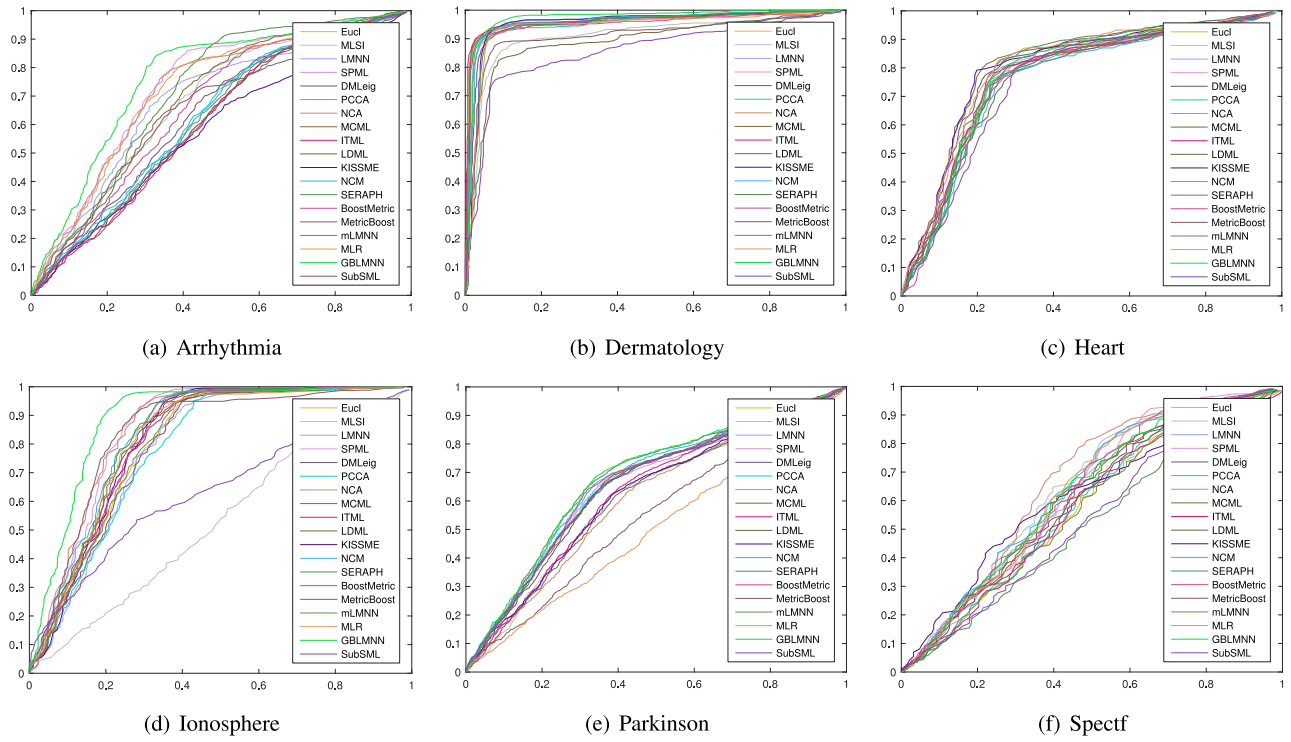


Fig. 5. ROC curve.

best on both ACC and AUC, owing to nonlinear feature transformation. Sub-SML and MLR perform not well, probably because they are only suitable for particular datasets.

In summary, the methods in top three grades with respect to ACC and AUC are: (1) GBLMNN, MCML and SPML; (2) BoostMetric, LMNN and SERAPH; (3) NCA, ITML and DML-eig. In consideration of complexity, the above methods can be divided into three grades with respect to TM, the first ones are ITML, DML-eig and BoostMetric. The second grade contains LMNN, SERAPH and SPML. MCML, NCA and GBLMNN are the three slowest methods. For the selection of metric learning methods, GB-LMNN ,MCML and SPML are recommended for large ACC and AUC. And BoostMetric, LMNN and SERAPH are recommended for fast training and relative high accuracy.

Further, to validate the classification ability of the selected methods on large scale datasets, we make experiments on 6 large

Table 4
The experimental results of the selected metric learning methods, including ACC, AUC and TM.

Datasets	AR	BU	CM	DE	HE	HP	IO	LI	PA	SE	SO	SP	TH	VA	VE	VT	WI	WP	Mean
Eucl	$62.4_{3.7}$ $60.1_{3.6}$ 0.02	$61.5_{2.8}$ $59.8_{4.8}$ 0.01	$60.9_{2.5}$ $59.1_{2.5}$ 0.01	$96.6_{1.6}$ **96.8**$_{1.8}$ 0.06	$77.7_{2.5}$ $74.9_{3.3}$ 0.00	$80.4_{4.3}$ $65.2_{8.5}$ 0.01	$84.5_{3.0}$ $77.8_{3.9}$ 0.01	$74.4_{4.8}$ – 0.01	$65.0_{1.8}$ $65.2_{2.0}$ 0.01	$93.0_{2.5}$ – 0.00	$84.0_{5.5}$ $81.3_{6.1}$ 0.01	$72.5_{3.9}$ $57.1_{6.6}$ 0.00	$94.0_{2.2}$ – 0.00	$62.9_{5.4}$ – 0.00	$70.1_{1.9}$ – 0.01	$92.0_{2.4}$ $90.6_{3.6}$ 0.01	$96.3_{1.7}$ – 0.00	$71.3_{5.2}$ $58.2_{7.7}$ 0.00	77.8 70.3 0.00
MLSI	$67.9_{3.8}$ $66.3_{3.4}$ 949	$59.5_{3.7}$ $57.3_{6.3}$ 1.75	$58.6_{1.3}$ $56.9_{2.3}$ 33.8	$91.6_{5.7}$ $89.7_{6.2}$ 12.5	$78.8_{4.4}$ $76.0_{4.6}$ 0.99	$81.9_{4.0}$ $69.3_{7.3}$ 0.30	62.5_{11} $52.1_{9.6}$ 24.18	$66.8_{3.8}$ – 7.3	$62.1_{2.2}$ $61.0_{2.3}$ 61.0	$91.6_{3.0}$ – 0.89	$83.2_{4.4}$ $82.1_{5.4}$ 5.89	$77.3_{4.2}$ $62.6_{6.6}$ 0.91	$95.7_{2.7}$ – 0.68	$57.9_{4.6}$ – 1.69	$57.1_{3.2}$ – 14.7	$91.6_{2.0}$ $90.5_{3.2}$ 2.49	$97.0_{2.2}$ – 0.57	$72.8_{3.7}$ $58.9_{5.8}$ 0.55	75.2 68.6 62.2
LMNN	$69.3_{4.7}$ $67.6_{5.3}$ 2.48	$62.0_{4.3}$ $61.1_{6.3}$ 1.75	$61.2_{1.9}$ $58.7_{1.1}$ 12.2	$96.9_{1.4}$ $94.8_{1.9}$ 0.42	$78.1_{3.1}$ $75.1_{3.4}$ 0.93	$81.7_{5.1}$ $71.2_{8.7}$ 0.29	$84.2_{2.1}$ $76.1_{4.2}$ 1.23	$79.9_{3.6}$ – 2.24	$66.3_{2.1}$ $66.1_{1.9}$ 9.04	**96.0**$_{2.4}$ – 0.48	$81.1_{5.4}$ $79.0_{6.0}$ 0.62	$75.8_{4.7}$ $60.9_{9.2}$ 0.66	$93.2_{3.1}$ – 0.56	**74.8**$_{3.5}$ – 1.93	$77.2_{1.2}$ – 7.24	$94.2_{2.4}$ $92.6_{2.8}$ 1.17	**98.9**$_{2.3}$ – 0.32	$73.3_{5.7}$ 63.5_{11} 0.58	80.3 71.6 2.45
SPML	$74.1_{4.4}$ $72.2_{4.2}$ 15.2	$60.5_{4.7}$ $58.3_{5.8}$ 3.41	$61.3_{2.1}$ $59.8_{2.9}$ 6.26	$96.5_{1.5}$ $94.5_{1.9}$ 4.25	$78.5_{2.3}$ $76.2_{3.4}$ 3.19	$78.7_{6.2}$ 59.9_{15} 3.05	$88.5_{1.7}$ $82.3_{2.9}$ 4.59	$75.8_{3.7}$ – 5.97	$64.9_{2.6}$ $65.0_{2.7}$ 6.70	$94.8_{2.6}$ – 2.94	**86.2**$_{4.3}$ $82.8_{4.7}$ 4.94	**79.4**$_{3.7}$ $59.6_{5.7}$ 4.72	$96.8_{2.1}$ – 2.91	$71.1_{3.3}$ – 4.61	$74.3_{2.2}$ – 6.00	$95.0_{2.1}$ $93.6_{3.0}$ 4.49	$98.1_{0.9}$ – 2.90	$75.7_{5.8}$ 57.9_{12} 3.98	80.6 71.8 5.01
DML-eig	$64.6_{4.3}$ $62.3_{3.1}$ 5.47	$59.1_{5.0}$ $57.7_{7.1}$ 0.11	$63.6_{2.5}$ $61.9_{3.1}$ 0.79	$95.7_{1.6}$ $94.1_{1.7}$ 0.56	$80.2_{4.5}$ $78.2_{5.4}$ 0.27	$78.5_{6.4}$ $61.6_{9.4}$ 0.32	$84.5_{3.6}$ $76.9_{4.5}$ 0.71	$69.1_{4.2}$ – 0.95	$65.3_{2.9}$ $65.1_{3.4}$ 0.70	$93.5_{2.9}$ – 0.20	$83.5_{4.2}$ $80.8_{5.5}$ 0.54	$74.0_{3.5}$ $56.9_{8.9}$ 0.68	$94.8_{1.7}$ – 0.14	$65.6_{3.8}$ – 0.30	$69.7_{1.9}$ – 0.41	$93.8_{1.7}$ $91.8_{2.7}$ 0.38	$97.8_{1.5}$ – 0.26	$71.8_{5.9}$ $56.8_{8.1}$ 0.39	78.1 70.4 0.73
PCCA	$63.3_{3.9}$ $61.2_{3.7}$ 492	$61.4_{2.7}$ $59.7_{4.6}$ 10.1	$60.6_{3.0}$ $58.7_{2.9}$ 195	$96.1_{2.1}$ $94.3_{2.1}$ 16.5	$77.5_{3.2}$ $74.8_{3.8}$ 6.74	$80.4_{4.1}$ $65.7_{8.5}$ 2.37	$83.2_{3.2}$ $76.0_{4.4}$ 15.7	$72.4_{5.3}$ – 49.9	$65.0_{1.9}$ $65.2_{2.0}$ 122	$93.2_{2.7}$ – 3.83	$84.3_{4.3}$ $81.6_{5.0}$ 10.6	$73.2_{4.6}$ $57.7_{7.5}$ 11.3	$94.0_{2.2}$ – 4.01	$63.3_{4.7}$ – 7.77	$70.0_{2.0}$ – 70.8	$91.8_{2.0}$ $90.6_{2.3}$ 17.6	$97.0_{1.3}$ – 2.90	$71.5_{4.7}$ $57.5_{7.8}$ 5.01	77.7 70.3 58.0
NCA	$72.5_{4.4}$ $70.8_{4.0}$ 148	$60.0_{3.8}$ $58.0_{4.3}$ 6.41	$62.4_{2.1}$ $61.2_{2.5}$ 72.1	$96.0_{1.8}$ $94.1_{2.2}$ 4.1	$77.9_{4.4}$ $76.1_{4.5}$ 6.27	$76.8_{5.3}$ 58.9_{13} 1.69	$88.2_{3.6}$ $82.5_{4.3}$ 9.37	$73.7_{5.6}$ – 29.8	$60.6_{2.5}$ $60.5_{2.2}$ 53.0	$93.8_{2.6}$ – 2.78	$81.3_{6.5}$ $77.4_{7.5}$ 4.73	$76.8_{4.5}$ **63.8**$_{7.7}$ 6.02	$96.0_{1.1}$ – 2.00	$66.0_{9.1}$ – 9.82	$71.0_{2.7}$ – 32.9	$94.0_{2.3}$ $91.8_{2.7}$ 6.13	$97.8_{1.5}$ – 0.88	$75.3_{5.8}$ $59.6_{7.9}$ 3.75	78.9 71.2 22.2
MCML	$69.4_{3.1}$ $66.8_{3.6}$ 124	$64.8_{4.6}$ $63.9_{5.7}$ 2.47	**63.7**$_{2.3}$ **62.2**$_{3.0}$ 29.7	$96.8_{1.3}$ $94.9_{2.0}$ 7.13	$80.5_{2.8}$ $78.7_{3.7}$ 2.27	$80.9_{6.7}$ **68.5**$_{13}$ 1.37	$86.7_{2.2}$ $79.8_{4.0}$ 6.22	**80.1**$_{4.6}$ – 14.2	$65.4_{3.3}$ $65.4_{3.2}$ 26.8	$95.7_{1.8}$ – 1.90	$84.4_{5.2}$ $81.4_{6.1}$ 4.33	$73.2_{3.7}$ $60.3_{3.4}$ 5.64	$95.8_{2.1}$ – 2.12	$70.9_{4.4}$ – 4.31	$74.8_{1.4}$ – 21.7	$96.0_{2.3}$ **94.6**$_{2.5}$ 5.91	$98.5_{1.5}$ – 1.69	$73.2_{5.7}$ $58.2_{8.1}$ 2.65	80.6 72.9 14.7
ITML	$62.6_{4.7}$ $59.5_{4.8}$ 1.11	$61.8_{3.2}$ $60.3_{4.8}$ **0.02**	$61.5_{1.9}$ $59.0_{2.6}$ **0.07**	$97.0_{1.4}$ $95.2_{1.9}$ **0.03**	$78.6_{3.3}$ $75.9_{3.8}$ **0.02**	**84.3**$_{3.6}$ $68.3_{8.5}$ **0.03**	$85.8_{1.7}$ $78.9_{3.5}$ **0.05**	$76.0_{3.9}$ – **1.25**	$62.6_{1.9}$ $62.5_{2.5}$ **0.04**	$94.4_{2.3}$ – **0.06**	$81.9_{5.8}$ $78.5_{6.8}$ **0.05**	$69.9_{4.7}$ $57.8_{8.6}$ **0.04**	$94.8_{2.7}$ – **0.06**	$70.7_{4.3}$ – **0.08**	$73.5_{3.8}$ – **0.09**	$93.5_{2.3}$ $92.0_{2.7}$ **0.03**	$97.4_{1.0}$ – **0.07**	$74.3_{3.4}$ **61.7**$_{4.7}$ **0.03**	78.9 70.8 **0.17**
LDML	$62.6_{3.8}$ $59.4_{5.7}$ 4.34	$54.4_{4.1}$ $51.7_{6.1}$ 0.31	$59.0_{2.1}$ $57.1_{2.3}$ 4.89	$89.2_{3.3}$ $87.6_{4.6}$ 1.15	$76.8_{3.8}$ $74.5_{4.8}$ 0.30	$80.9_{4.5}$ 61.1_{13} 0.14	$85.2_{2.9}$ $79.5_{3.4}$ 1.22	$68.5_{5.8}$ – 1.56	$61.6_{3.1}$ $61.2_{3.3}$ 4.29	$89.5_{5.5}$ – 0.23	$79.7_{5.6}$ $77.4_{5.7}$ 0.85	$73.0_{6.7}$ $58.3_{9.8}$ 0.93	$91.2_{8.3}$ – 0.13	$50.4_{6.5}$ – 0.96	$53.1_{2.3}$ – 2.71	$86.7_{5.6}$ $84.8_{5.0}$ 1.12	$89.4_{4.4}$ – 0.15	$68.8_{6.7}$ $50.3_{9.2}$ 0.74	73.3 66.8 1.46
KISSME	$58.1_{5.0}$ $56.6_{4.5}$ 11.8	$63.5_{3.7}$ $61.2_{3.5}$ 0.41	$63.5_{3.9}$ $62.2_{3.7}$ 99.3	$95.9_{1.4}$ $93.8_{2.2}$ 0.62	**80.6**$_{2.9}$ **78.7**$_{3.5}$ 0.22	$81.3_{5.3}$ 62.8_{11} 0.06	$86.5_{1.9}$ $79.2_{4.0}$ 0.54	$45.3_{3.9}$ – 1.43	$62.1_{3.1}$ $61.9_{3.6}$ 21.9	$95.7_{2.4}$ – 0.10	$78.3_{4.3}$ $76.5_{4.9}$ 0.22	$58.9_{7.0}$ $59.2_{8.2}$ 0.32	$95.2_{2.5}$ – 0.11	$74.7_{3.7}$ – 0.25	**79.8**$_{2.0}$ – 11.6	**96.0**$_{1.9}$ $94.3_{2.4}$ 0.93	$97.6_{1.8}$ – **0.07**	$67.8_{3.1}$ $55.7_{5.6}$ 0.13	76.7 70.2 8.34
NCMML	$63.6_{4.4}$ $61.5_{3.7}$ **1.10**	$61.9_{4.0}$ $60.1_{4.9}$ 0.44	$59.6_{2.7}$ $57.7_{2.9}$ 1.10	$96.3_{1.7}$ $94.5_{2.2}$ 1.80	$76.7_{3.0}$ $74.1_{3.8}$ 1.78	$78.1_{5.7}$ 56.3_{11} 1.81	$87.5_{2.6}$ $81.2_{3.6}$ 1.86	$76.1_{5.0}$ – 0.63	$66.3_{1.4}$ $66.5_{2.1}$ 1.81	$91.9_{3.0}$ – 0.44	$84.0_{5.1}$ $80.7_{6.1}$ 1.94	$75.6_{4.2}$ $61.5_{7.9}$ 1.93	$94.6_{2.5}$ – 0.44	$61.8_{3.4}$ – 0.44	$70.0_{2.0}$ – 1.09	$90.7_{3.4}$ $90.1_{4.1}$ 1.82	$97.8_{1.9}$ – 0.44	$76.0_{6.5}$ $59.6_{7.2}$ 1.91	78.2 70.3 1.26
SERAPH	$72.8_{3.4}$ $70.1_{2.7}$ 5.00	$60.3_{4.2}$ $58.4_{5.6}$ 0.88	$62.2_{1.9}$ $60.6_{2.2}$ 26.0	$95.5_{1.87}$ $93.1_{2.6}$ 0.89	$80.1_{3.7}$ $77.9_{4.9}$ 0.51	$82.3_{7.4}$ 66.1_{11} 0.23	$87.5_{2.3}$ $80.4_{3.8}$ 0.88	$78.0_{3.3}$ – 3.27	$67.1_{2.1}$ $67.1_{2.1}$ 14.8	$93.8_{2.0}$ – 0.36	$81.4_{3.4}$ $79.0_{3.8}$ 0.58	$75.8_{3.2}$ $56.5_{3.5}$ 0.58	$95.7_{1.7}$ – 0.33	$69.9_{2.4}$ – 0.61	$74.0_{2.7}$ – 5.98	$94.3_{2.0}$ $92.4_{2.5}$ 1.14	$98.0_{1.8}$ – 0.28	$74.8_{5.2}$ $57.6_{7.3}$ 0.33	80.2 71.0 3.48
BoostMetric	$67.8_{2.9}$ $64.8_{4.2}$ 6.02	$62.5_{5.4}$ $61.0_{6.5}$ 0.08	$62.5_{0.9}$ $60.6_{1.4}$ 0.09	$96.8_{1.4}$ $95.1_{1.4}$ 0.58	$75.4_{3.7}$ $72.8_{2.7}$ 0.16	$77.2_{5.5}$ 57.9_{12} 0.38	$85.7_{3.3}$ $78.7_{4.6}$ 0.74	$76.1_{4.6}$ – **1.25**	$65.6_{1.6}$ $64.5_{2.2}$ 1.03	$94.9_{2.8}$ – 0.26	$85.4_{4.8}$ $83.0_{7.0}$ 0.69	$75.9_{3.9}$ $61.7_{7.0}$ 0.86	$95.8_{2.1}$ – 0.21	$72.6_{4.6}$ – 0.43	$78.5_{1.8}$ – 0.50	$93.9_{2.0}$ $92.6_{2.7}$ 0.41	$96.5_{2.7}$ – 0.31	**77.8**$_{2.9}$ $56.9_{6.7}$ 0.72	80.4 72.1 0.82
MetricBoost	$64.9_{7.0}$ $63.8_{6.5}$ 8.03	$58.5_{4.1}$ $55.9_{4.7}$ 0.37	$59.2_{4.5}$ $57.7_{5.2}$ 1.75	$92.2_{2.8}$ $90.1_{3.6}$ 0.53	$77.4_{2.7}$ $75.2_{3.0}$ 0.34	$77.0_{4.8}$ 56.3_{12} 0.20	$87.1_{3.4}$ $81.9_{4.4}$ 0.51	$64.0_{6.8}$ – 1.33	$54.4_{3.3}$ $54.4_{3.1}$ 1.41	$89.2_{5.3}$ – 0.24	$80.0_{5.7}$ $77.6_{7.2}$ 0.48	$75.9_{3.6}$ $59.2_{6.5}$ 0.46	**97.4**$_{1.8}$ – 0.23	$62.8_{7.0}$ – 0.36	$55.3_{4.0}$ – 1.04	$93.6_{2.3}$ $91.8_{2.7}$ 0.51	$96.5_{4.3}$ – 0.22	$72.8_{3.4}$ $55.3_{6.3}$ 0.29	75.5 68.3 1.02
mLMNN	$63.2_{3.7}$ $60.9_{3.8}$ 20.7	$61.2_{2.8}$ $59.3_{5.3}$ 6.17	$59.3_{2.6}$ $57.6_{2.4}$ 32.1	$96.1_{2.1}$ $94.3_{2.1}$ 16.5	$77.7_{2.5}$ $74.9_{3.3}$ 1.51	$80.4_{4.3}$ $65.2_{8.5}$ 1.05	$84.8_{2.9}$ $78.2_{3.8}$ 2.77	$74.4_{4.2}$ – 15.2	$65.0_{2.3}$ $65.1_{2.4}$ 36.5	$93.2_{2.4}$ – 2.94	$84.1_{5.4}$ $81.4_{6.1}$ 2.06	$72.3_{3.5}$ $56.7_{5.8}$ 3.19	$94.0_{2.2}$ – 2.97	$62.9_{5.2}$ – 4.62	$69.0_{2.3}$ – 27.8	$92.7_{2.4}$ $91.1_{2.7}$ 2.14	$96.7_{1.5}$ – 2.20	$71.2_{4.9}$ $58.1_{7.5}$ 1.91	77.7 70.3 9.30
MLR	$72.3_{3.3}$ $70.6_{4.4}$ 570	$49.2_{9.0}$ $50.8_{6.6}$ 2.75	$47.8_{7.0}$ $50.1_{2.6}$ 119	$95.2_{0.86}$ $92.8_{1.3}$ 10.3	$80.4_{3.4}$ $78.7_{4.7}$ 2.61	$83.4_{3.6}$ 66.9_{10} 1.36	$86.0_{2.5}$ $79.6_{5.1}$ 6.67	51.1_{13} – 19.3	$49.4_{2.6}$ $50.1_{3.6}$ 44.2	78.6_{18} – 1.47	$78.7_{3.4}$ $75.8_{5.3}$ 2.89	$75.4_{3.6}$ $61.9_{5.7}$ 3.69	65.4_{16} – 1.09	41.2_{12} – 2.01	$46.0_{7.7}$ – 20	$92.3_{3.6}$ $90.4_{4.3}$ 10.8	$95.9_{5.1}$ – 1.74	$71.8_{6.1}$ $51.9_{9.3}$ 2.10	70.0 68.2 45.7
GB-LMNN	**76.9**$_{2.9}$ **75.1**$_{3.6}$ 267	**66.5**$_{3.8}$ **66.0**$_{4.6}$ 0.66	$62.9_{2.2}$ $61.4_{2.6}$ 1.76	**97.5**$_{1.7}$ $95.8_{2.0}$ 1.51	$78.5_{3.6}$ $76.2_{3.5}$ 0.91	$81.7_{5.5}$ $66.7_{8.9}$ 0.90	**91.5**$_{2.8}$ **86.7**$_{4.2}$ 13.3	$79.6_{3.6}$ – 90.3	**67.6**$_{1.8}$ $67.5_{2.5}$ 28.4	$93.8_{3.8}$ – 0.87	$85.2_{5.7}$ $81.5_{6.7}$ 31.7	$75.3_{4.1}$ $60.6_{6.7}$ 7.48	$92.8_{3.8}$ – 0.49	$72.5_{5.8}$ – 1.91	$74.9_{2.5}$ – 3.44	$95.0_{1.1}$ $93.2_{2.2}$ 0.74	$98.0_{2.2}$ – 1.38	$72.2_{6.0}$ $57.0_{8.2}$ 9.44	**81.2** **74.0** 25.7
Sub-SML	$62.4_{3.7}$ $60.1_{3.6}$ 1e3	$54.4_{5.3}$ $50.8_{5.4}$ 46.4	$57.2_{2.9}$ $54.9_{3.2}$ 996	$86.3_{3.4}$ $84.7_{3.1}$ 75.0	$76.9_{4.7}$ $73.2_{5.8}$ 29.8	$82.1_{5.4}$ 55.8_{12} 10.4	54.3_{13} $62.6_{7.2}$ 68.4	$71.1_{4.3}$ – 152	$65.0_{2.0}$ $65.1_{2.2}$ 532	$93.3_{1.6}$ – 18.1	$85.6_{4.0}$ **83.0**$_{4.7}$ 33.7	$77.3_{3.1}$ $52.0_{7.5}$ 40.9	$95.2_{1.5}$ – 17.9	$64.1_{4.2}$ – 33.2	$72.1_{1.8}$ – 323	$48.6_{8.5}$ $61.1_{5.8}$ 79.8	$97.0_{2.0}$ – 13.0	$74.0_{4.5}$ $54.2_{8.2}$ 22.2	73.2 61.3 215

scale datasets. The statistics of the datasets are given in Table 6. From Table 4, we can see that most of the methods are not suitable for large scale problems, so only the methods whose time rank in top 8 are selected. The classification results are shown in Table 6. For each method and its corresponding dataset, the first number denotes ACC and its subscript denotes the standard deviation of 3-fold cross validation, the second number denotes average training time. It can be seen that the performance of BoostMetric and LMNN ranks first and second respectively. The results are consistent with previous evaluation. For large scale datasets, ITML, DML-eig and BoostMetric run much faster than LMNN, but the former two obtain worse performance than LMNN. NCMML performs much worse than Eucl, probably resulting from that computing distance using class centroid will loss much useful information in large scale problems.

In the third experiment, for all the methods, the learned metrics will be applied in kNN classification with different neighborhood size k, selected from the set $\{1, 3, 5, 7, 9\}$. The corresponding

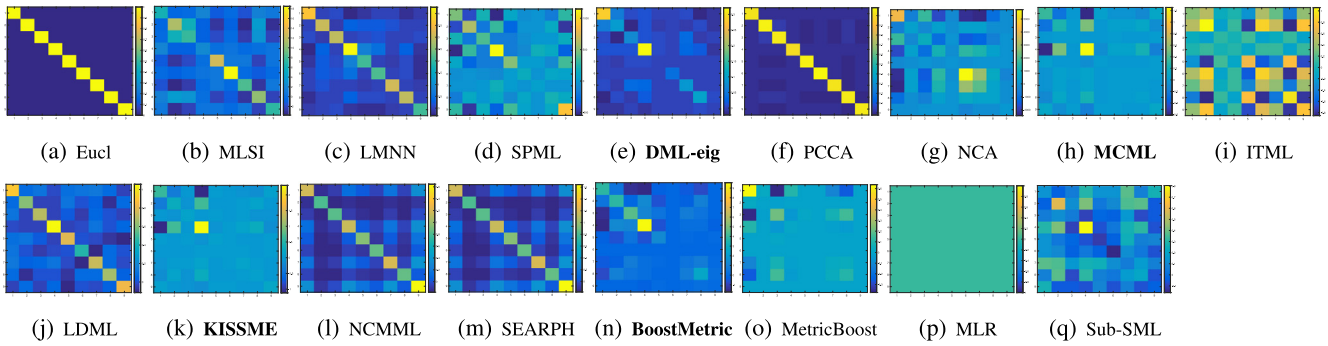


Fig. 6. CMC metrics.

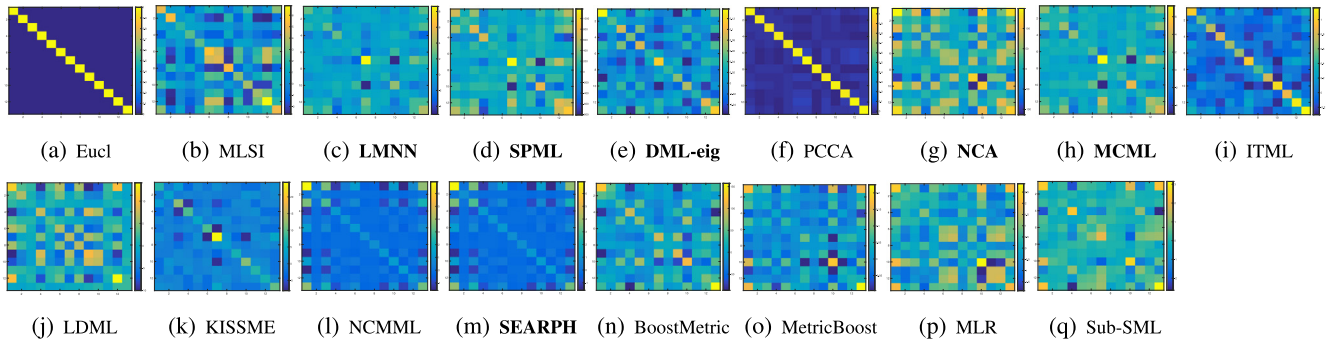


Fig. 7. Wine metrics.

Table 5
Wilcoxon test on all the selected methods.

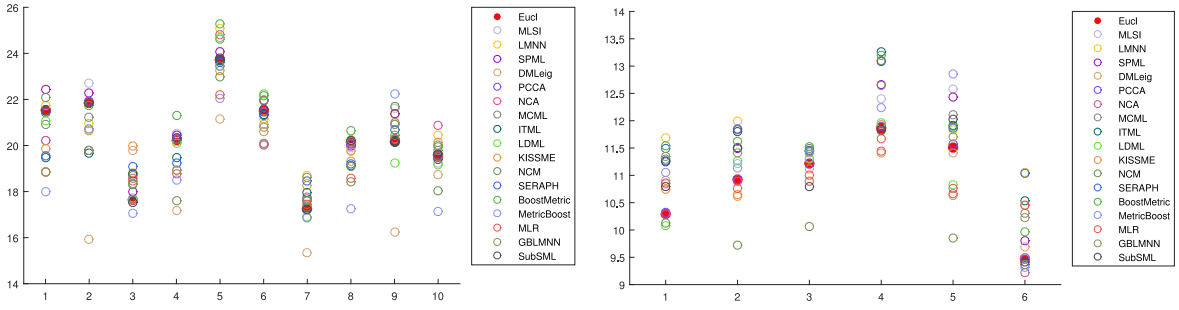
Algorithm	ACC				AUC				Total score	Time rank
	Win	Draw	Lose	Score	Win	Draw	Lose	Score		
GBLMNN	144	169	11	228.5	97	123	3	155.5	384	15
MCML	135	172	17	221	81	134	8	145	366	13
SPML	126	180	18	216	58	159	6	134	350	10
BoostMetric	113	183	28	204.5	76	143	15	131.5	336	4
LMNN	109	192	23	205	45	168	21	118	323	8
SERAPH	105	194	25	202	46	149	28	117.5	319.5	9
NCA	83	192	49	179	47	158	24	119	298	14
ITML	75	190	59	170	37	164	33	107	277	2
DMLeig	57	207	60	160.5	35	173	26	113.5	274	3
KISSME	92	132	100	158	47	127	48	109	267	11
NCMML	53	206	65	156	33	151	39	105.5	261.5	6
Eucl	45	198	81	144	35	166	33	109	253	1
PCCA	43	204	77	145	28	161	33	106	251	17
mLMNN	44	199	81	143.5	30	155	38	104.5	248	12
MLSI	40	164	120	122	35	136	63	93.5	215.5	18
MetricBoost	47	156	121	125	17	138	67	84	209	5
Sub-SML	49	150	125	124	15	83	124	55	179	19
MLR	38	116	170	96	32	102	88	82	178	16
LDML	15	126	183	78	8	110	105	62.5	140.5	7

Table 6
Classification accuracy on large scale datasets.

Datasets (inst. ×attr.)	Abalone 4177 × 8	Letter 20000 × 16	Magic 19020 × 10	Spambase 4601 × 57	Waveform 5000 × 21	Wine-quality 4873 × 11
Eucl	$50.5_{0.9}$/0.05	$95.2_{0.6}$/1.58	$77.5_{0.3}$/0.12	$80.4_{0.5}$/0.25	$77.9_{0.7}$/0.12	$56.1_{1.4}$/0.11
LMNN	$\mathbf{50.7}_{1.4}$/131	$96.1_{0.4}$/699	$79.0_{0.1}$/99.8	$\mathbf{89.9}_{1.5}$/34.5	$78.5_{0.5}$/28.2	$59.3_{0.8}$/100
DML-eig	49.2_{0}.4/0.90	$95.5_{0.4}$/28.5	$77.8_{0.5}$/20.7	$85.0_{2.3}$/1.87	$80.3_{0.9}$/1.35	$58.0_{0.4}$/1.37
ITML	$50.2_{1.1}$/0.36	$93.3_{0.7}$/10.2	$77.9_{0.4}$/7.70	$79.7_{1.0}$/0.44	$77.8_{1.1}$/0.49	$56.6_{1.8}$/0.53
LDML	$49.0_{0.6}$/29.2	$95.2_{0.5}$/134	$77.5_{0.3}$/94.1	$80.4_{0.5}$/13.4	$72.3_{1.3}$/87.0	$56.1_{1.4}$/6.73
NCMML	$50.6_{1.7}$/0.12	$14.0_{0.6}$/1.89	$57.5_{1.0}$/0.10	$67.7_{0.6}$/0.39	$78.0_{0.2}$/0.24	$51.6_{1.3}$/0.22
BoostMetric	$48.6_{1.3}$/0.50	$\mathbf{96.7}_{0.6}$/15.8	$\mathbf{79.6}_{0.3}$/9.60	$76.9_{4.1}$/3.95	$79.3_{0.7}$/1.80	$\mathbf{59.7}_{1.7}$/1.29
MetricBoost	$42.5_{5.8}$/4.14	$82.4_{0.8}$/70.4	$67.3_{0.2}$/41.8	$69.7_{2.2}$/12.5	$\mathbf{82.8}_{0.7}$/6.25	$58.6_{0.8}$/5.18

Table 7
The classification ACC of the selected methods on different neighborhood sizes. The numbers in the parentheses denote relative rate of growth on ACC, compared with Eucl. The numbers in red color are the largest ones in the corresponding method.

Dataset	Corel					Caltech				
k	1	3	5	7	9	1	3	5	7	9
Eucl	59.3	59.3	60.7	60.0	61.3	84.4	80.7	78.1	77.8	77.0
MLSI	54.0(-8.9)	50.7(-14.6)	53.3(-12.1)	53.3(-11.1)	55.3(-9.8)	78.9(-6.6)	80.7(0.0)	79.3(1.4)	80.7(3.8)	79.6(3.4)
LMNN	**78.7**(32.6)	**72.0**(21.3)	**75.3**(24.2)	**77.3**(28.9)	**76.7**(25.0)	**94.8**(12.3)	**95.6**(18.3)	**94.8**(21.3)	**94.4**(21.4)	**94.4**(22.6)
SPML	67.3(13.5)	68.7(15.7)	68.7(13.2)	*70.7*(17.8)	71.3(16.3)	91.1(7.9)	90.4(11.9)	90.4(15.6)	90.0(15.7)	89.6(16.3)
DML-eig	46.7(-21.3)	*69.3*(16.9)	*69.3*(14.3)	72.0(20.0)	73.3(19.6)	91.5(8.3)	93.7(16.1)	93.0(19.0)	92.6(19.0)	93.3(21.2)
PCCA	59.3(0.0)	61.3(3.4)	60.0(-1.1)	62.7(4.4)	62.0(1.1)	84.1(-0.4)	80.7(0.0)	78.9(0.9)	77.8(0.0)	76.7(-0.5)
NCA	61.3(3.4)	63.3(6.7)	67.3(11.0)	68.7(14.4)	69.3(13.0)	90.0(6.6)	89.3(10.6)	88.5(13.3)	87.0(11.9)	87.8(13.9)
MCML	*65.3*(10.1)	67.3(13.5)	68.7(13.2)	68.7(14.4)	*70.7*(15.2)	93.0(10.1)	*92.6*(14.7)	93.0(19.0)	*91.9*(18.1)	*91.5*(18.8)
ITML	63.3(6.7)	66.7(12.4)	68.0(12.1)	69.3(15.6)	67.3(9.8)	85.9(1.8)	84.1(4.1)	85.2(9.0)	84.1(8.1)	83.7(8.7)
LDML	44.0(-25.8)	42.0(-29.2)	42.7(-29.7)	45.3(-24.5)	46.0(-25.0)	60.7(-28.1)	65.9(-18.3)	63.3(-19.0)	65.5(-15.8)	65.6(-14.8)
KISSME	53.3(-10.1)	52.7(-11.2)	56.0(-7.7)	54.0(-10.0)	54.0(-12.0)	83.0(-1.8)	83.0(2.8)	82.2(5.2)	81.5(4.8)	83.0(7.2)
NCMML	16.7(-71.9)	16.7(-71.9)	17.3(-71.4)	18.0(-70.0)	18.0(-70.7)	22.2(-73.7)	21.1(-73.9)	22.6(-71.1)	22.6(-71.0)	21.5(-72.1)
SERAPH	67.3(13.5)	70.7(19.1)	72.0(18.7)	68.0(13.3)	70.0(14.1)	*92.6*(9.6)	90.4(11.9)	*92.2*(18.0)	91.5(17.6)	91.1(18.3)
BoostMetric	63.3(6.7)	61.3(3.4)	62.7(3.3)	65.3(8.9)	66.0(7.6)	91.5(8.3)	91.1(12.8)	90.7(16.1)	90.0(15.7)	89.6(16.3)
MetricBoost	30.0(-49.4)	30.7(-48.3)	30.7(-49.5)	27.3(-54.4)	28.7(-53.3)	30.7(-63.6)	26.3(-67.4)	28.9(-63.0)	34.8(-55.2)	30.7(-60.1)
MLR	53.3(-10.1)	52.0(-12.4)	49.3(-18.7)	50.0(-16.7)	47.3(-22.8)	84.4(0.0)	80.4(-0.5)	77.0(-1.4)	75.9(-2.4)	77.0(0.0)
GB-LMNN	62.7(5.6)	58.0(-2.2)	56.0(-7.7)	51.3(-14.4)	52.0(-15.2)	84.0(-0.4)	82.6(2.3)	79.6(1.9)	78.5(1.0)	80.0(3.8)
Sub-SML	61.3(3.4)	62.7(5.6)	62.7(3.3)	63.3(5.6)	61.3(0.0)	20.7(-75.4)	20.7(-74.3)	21.9(-72.0)	22.2(-71.4)	22.2(-71.2)



(a) Corel (b) Caltech

Fig. 8. The ratio between inter-class distance and intra-class distance. The horizontal axis denotes class number.

ACCs are shown in Table 7. For each neighbor size, the methods that performs best, second and third will be marked in bold type, underline and italic type respectively. It can be seen that LMNN performs best ten times, DML-eig, SERAPH and MCML performs second six, three and two times severally. The results are not consistent with the Wilcoxon test in Table 5 very much. A reasonable explanation is that images are unstructured data and LMNN is good at handling such kind of data. The performance of nonlinear metric learning, GBLMNN, is not as well as that in benchmark datasets. Comprehensively, LMNN, SERAPH and MCML performs comparatively good in benchmark datasets and image sets.

The numbers in the parentheses are the relative rate of growth (RRG) compared with Eucl,

$$RRG = \frac{ACC_M - ACC_E}{ACC_E} \tag{4.1}$$

where ACC_M denotes the *ACC* of metric learning method and ACC_E denotes the *ACC* of Eucl.

The numbers in red color are the largest RRGs in the corresponding method. For Corel, 8 methods obtain the largest RRGs when $k = 7$. And 10 methods obtains the largest RRGs when $k = 9$ in Caltech. A proper deduction is that each dataset corresponds to a neighbor size, with which metric learning method can improve *k*NN performance greatly. Since metric learning is equivalent to transforming original examples into new space, the ratio of inter-class distance to intra-class distance for each class in the transformed space are displayed in Fig. 8. Compared with Eucl, the methods that performs better than Eucl can always enlarge the ratios of several and even all the classes. The distances between dissimilar patterns are expanded and the distances between similar points are shrinked. Therefore, the ratio of inter-class distance to intra-class distance can be used as an index to evaluate the ability in learning useful metric.

5. Conclusions

In this paper, an experimental survey is made to compare the performance of different metric learning methods. First, a taxonomy is provided and the methods are divided into five categories according to their most distinguished characters: pairwise cost, probabilistic framework, boost-like approaches, advantages variants and specific application. For each category, several classical and influenced methods are selected from famous journals and conferences. The proposed taxonomy can help the scholars in metric learning understand the methods systematically and then propose new methods in consideration of the advantages and drawbacks of each category. In experiments, 18 representative methods are evaluated and the classification performances on benchmark datasets are compared from statistical view. Then image datasets are selected to explore the relation between classification accuracy and neighborhood size and the relation between accuracy and distance change. It is confirmed that there is no one method can perform best on all the datasets, but the experimental results provide the evidence that several methods have high ability in learning informative metrics. We will give some suggestions on the selection of metric learning methods: (1) As three representative methods in their corresponding categories, BoostMetric, LMNN and SERAPH are first recommended since they perform the second grade with respect to the statistical score (ACC and AUC), with relative lower computational time; (2) GBLMNN and MCML

are recommended when seeking for high ACC and AUC and the computational cost is a secondary factor. But for unstructured data, such as image datasets, GBLMNN is not a good option; (3) ITML and DML-eig are two good choices when learning metric for large scale dataset due to the much lower computational complexity. (4) MLSI, Sub-SML and MLR are not recommended since they neither performs well with respect to ACC and AUC nor training fast. Further, to tap the potentials of each method as much as possible, a learned metric should be applied to *k*NN with different *k* and select the one with highest classification accuracy.

Acknowledgments

This work has been partially supported by grants from National Natural Science Foundation of China (Nos. 71731009, 61472390, 71331005, and 91546201), the Beijing Natural Science Foundation (No. 1162005).

References

Baghshah, M. S., & Shouraki, S. B. (2009). Semi-supervised metric learning using pairwise constraints. In *IJCAI, Vol. 9* (pp. 1217–1222). Citeseer.

Bar-Hillel, A., Hertz, T., Shental, N., & Weinshall, D. (2005). Learning a mahalanobis metric from equivalence constraints. *Journal of Machine Learning Research (JMLR)*, *6*(Jun), 937–965.

Bellet, A., Habrard, A., & Sebban, M. (2013). A survey on metric learning for feature vectors and structured data, Computer Science.

Bi, J., Wu, D., Lu, L., Liu, M., Tao, Y., & Wolf, M. (2011). Adaboost on low-rank psd matrices for metric learning. In *2011 IEEE conference on computer vision and pattern recognition* (pp. 2617–2624). IEEE.

Bian, W., & Tao, D. (2011). Learning a distance metric by empirical loss minimization. In *IJCAI proceedings-international joint conference on artificial intelligence, Vol. 22* (p. 1186).

Bilenko, M., Basu, S., & Mooney, R. J. (2004). Integrating constraints and metric learning in semi-supervised clustering. In *Proceedings of the twenty-first international conference on machine learning* (p. 11). ACM.

Cai, X., Wang, C., Xiao, B., Chen, X., & Zhou, J. (2012). Deep nonlinear metric learning with independent subspace analysis for face verification. In *Proceedings of the 20th ACM international conference on multimedia* (pp. 749–752). ACM.

Cao, Q., Ying, Y., & Li, P. (2013). Similarity metric learning for face recognition. In *Proceedings of the IEEE international conference on computer vision* (pp. 2408–2415).

Chai, J., Liu, H., Chen, B., & Bao, Z. (2010). Large margin nearest local mean classifier. *Signal Processing*, *90*(1), 236–248.

Chang, C.-C. (2012). A boosting approach for supervised mahalanobis distance metric learning. *Pattern Recognition*, *45*(2), 844–862.

Changpinyo, S., Liu, K., & Sha, F. (2013). Similarity component analysis. In *Advances in neural information processing systems* (pp. 1511–1519).

Cinbis, R. G., Verbeek, J., & Schmid, C. (2011). Unsupervised metric learning for face identification in tv video. In *2011 IEEE international conference on computer vision* (pp. 1559–1566). IEEE.

Cover, T. M. (1968). Rates of convergence for nearest neighbor procedures. In *Hawaii international conference on system sciences*.

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, *13*(1), 21–27.

Davis, J. V., & Dhillon, I. S. (2008). Structured metric learning for high dimensional problems. In *Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 195–203). ACM.

Davis, J. V., Kulis, B., Jain, P., Sra, S., & Dhillon, I. S. (2007). Information-theoretic metric learning. In *Proceedings of the 24th international conference on machine learning* (pp. 209–216). ACM.

Der, M., & Saul, L. K. (2012). Latent coincidence analysis: A hidden variable model for distance metric learning. In *Advances in neural information processing systems* (pp. 3230–3238).

Dong, Y., Zhang, L., Zhang, L., & Du, B. (2015). Maximum margin metric learning based target detection for hyperspectral images. *Isprs Journal of Photogrammetry & Remote Sensing*, *108*, 138–150.

Du, B., & Zhang, L. (2014a). A discriminative metric learning based anomaly detection method. *IEEE Transactions on Geoscience & Remote Sensing*, *52*(11), 6844–6857.

Du, B., & Zhang, L. (2014b). Target detection based on a dynamic subspace. *Pattern Recognition*, *47*(1), 344–358.

Feng, Z., Jin, R., & Jain, A. (2013). Large-scale image annotation by efficient and robust kernel metric learning. In *Proceedings of the IEEE international conference on computer vision* (pp. 1609–1616).

Galleguillos, C., McFee, B., & Lanckriet, G. R. (2014). Iterative category discovery via multiple kernel metric learning. *International Journal of Computer Vision*, *108*(1–2), 115–132.

Gao, X., Hoi, S. C., Zhang, Y., Wan, J., & Li, J. (2014). Soml: Sparse online metric learning with application to image retrieval. In *AAAI* (pp. 1206–1212).

Gao, Y., Wang, M., Ji, R., Wu, X., & Dai, Q. (2014). 3-d object retrieval with hausdorff distance learning. *IEEE Transactions on Industrial Electronics*, *61*(4), 2088–2098.

Globerson, A., & Roweis, S. T. (2005). Metric learning by collapsing classes. In *Advances in neural information processing systems* (pp. 451–458).

Goldberger, J., Hinton, G. E., Roweis, S. T., & Salakhutdinov, R. (2004). Neighbourhood components analysis. In *Advances in neural information processing systems* (pp. 513–520).

Guillaumin, M., Verbeek, J., & Schmid, C. (2009). Is that you? metric learning approaches for face identification. In *2009 IEEE 12th international conference on computer vision* (pp. 498–505). IEEE.

Guillaumin, M., Verbeek, J., & Schmid, C. (2010). Multiple instance metric learning from automatically labeled bags of faces. In *Computer vision–ECCV 2010* (pp. 634–647). Springer.

He, Y., Chen, W., Chen, Y., & Mao, Y. (2013). Kernel density metric learning. In *2013 IEEE 13th international conference on data mining* (pp. 271–280). IEEE.

Hoi, S. C., Liu, W., & Chang, S.-F. (2008). Semi-supervised distance metric learning for collaborative image retrieval. In *IEEE conference on computer vision and pattern recognition, 2008* (pp. 1–7). IEEE.

Hoi, S. C., Liu, W., Lyu, M. R., & Ma, W.-Y. (2006). Learning distance metrics with contextual constraints for image retrieval. In *2006 IEEE computer society conference on computer vision and pattern recognition, Vol. 2* (pp. 2072–2078). IEEE.

Hong, Y., Li, Q., Jiang, J., & Tu, Z. (2011). Learning a mixture of sparse distance metrics for classification and dimensionality reduction. In *2011 IEEE international conference on computer vision* (pp. 906–913). IEEE.

Hu, J., Lu, J., & Tan, Y.-P. (2014). Discriminative deep metric learning for face verification in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1875–1882).

Hu, J., Lu, J., & Tan, Y. P. (2016). Deep metric learning for visual tracking. *IEEE Transactions on Circuits & Systems for Video Technology*, *26*(11), 2056–2068.

Hu, J., Lu, J., Yuan, J., & Tan, Y.-P. (2014). Large margin multi-metric learning for face and kinship verification in the wild. In *Computer vision–ACCV 2014* (pp. 252–267). Springer.

Jain, A. K. (2008). *Data clustering: 50 years beyond K-means*. Springer Berlin Heidelberg.

Jain, P., Kulis, B., Davis, J. V., & Dhillon, I. S. (2012). Metric and kernel learning using a linear transformation. *Journal of Machine Learning Research (JMLR)*, *13*(1), 519–547.

Jain, P., Kulis, B., Dhillon, I. S., & Grauman, K. (2009). Online metric learning and fast similarity search. In *Advances in neural information processing systems* (pp. 761–768).

Jin, R., Wang, S., & Zhou, Y. (2009a). Regularized distance metric learning: Theory and algorithm. In *Advances in neural information processing systems* (pp. 862–870).

Jin, R., Wang, S., & Zhou, Z.-H. (2009b). Learning a distance metric from multi-instance multi-label data. In *IEEE conference on computer vision and pattern recognition, 2009* (pp. 896–902). IEEE.

Joachims, T., Finley, T., & Yu, C. N. J. (2009). Cutting-plane training of structural svms. *Machine Learning*, *77*(1), 27–59.

Jolliffe, I. T. (1986). *Principal component analysis and factor analysis*. Springer New York.

Kedem, D., Tyree, S., Sha, F., Lanckriet, G. R., & Weinberger, K. Q. (2012). Non-linear metric learning. In *Advances in neural information processing systems* (pp. 2573–2581).

Koestinger, M., Hirzer, M., Wohlhart, P., Roth, P. M., & Bischof, H. (2012). Large scale metric learning from equivalence constraints. In *2012 IEEE conference on computer vision and pattern recognition* (pp. 2288–2295). IEEE.

Kozakaya, T., Ito, S., & Kubota, S. (2011). Random ensemble metrics for object recognition. In *2011 IEEE international conference on computer vision* (pp. 1959–1966). IEEE.

Kulis, B. (0000). Metric learning: A survey, Foundations and Trends in Machine Learning 5(4).

Kunapuli, G., & Shavlik, J. (2012). Mirror descent for metric learning: A unified approach. In *Machine learning and knowledge discovery in databases* (pp. 859–874). Springer.

Lajugie, R., Arlot, S., & Bach, F. (2014). Large-margin metric learning for constrained partitioning problems. In *Proceedings of the 31st international conference on machine learning*.

Lebanon, G. (2006). Metric learning for text documents. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*(4), 497–508.

Li, W., Zhao, R., & Wang, X. (2012). Human reidentification with transferred metric learning. In *ACCV (1)* (pp. 31–44).

Likas, A., Vlassis, N., & Verbeek, J. J. (2001). The global k -means clustering algorithm. *Pattern Recognition*, *36*(2), 451–461.

Lim, D., Lanckriet, G., & McFee, B. (2013). Robust structural metric learning. In *Proceedings of the 30th international conference on machine learning* (pp. 615–623).

Liu, Y. (2006). Distance metric learning: A comprehensive survey, Michigan State Universiy.

Liu, W., & Tsang, I. W. (2015). Large margin metric learning for multi-label prediction. In *Twenty-ninth AAAI conference on artificial intelligence*.

Liu, M., & Vemuri, B. C. (2012). A robust and efficient doubly regularized metric learning approach. In *Computer vision–ECCV 2012* (pp. 646–659). Springer.

Liu, B., Wang, M., Hong, R., Zha, Z., & Hua, X.-S. (2010). Joint learning of labels and distance metric. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, *40*(3), 973–978.

Lu, J., Wang, G., Deng, W., & Jia, K. (2015). Reconstruction-based metric learning for unconstrained face verification. *IEEE Transactions on Information Forensics and Security*, *10*(1), 79–89.

Lu, J., Wang, G., Deng, W., Moulin, P., & Zhou, J. (2015). Multi-manifold deep metric learning for image set classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1137–1145).

Lu, J., Wang, G., & Moulin, P. (2013). Image set classification using holistic multiple order statistics features and localized multi-kernel metric learning. In *Proceedings of the IEEE international conference on computer vision* (pp. 329–336).

Lu, J., Zhou, X., Tan, Y.-P., Shang, Y., & Zhou, J. (2014). Neighborhood repulsed metric learning for kinship verification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *36*(2), 331–345.

Ma, L., Yang, X., & Tao, D. (2014). Person re-identification over camera networks using multi-task distance metric learning. *IEEE Transactions on Image Processing*, *23*(8), 3656–3670.

McFee, B., & Lanckriet, G. R. (2010). Metric learning to rank. In *Proceedings of the 27th international conference on machine learning* (pp. 775–782).

Mensink, T., Verbeek, J., Perronnin, F., & Csurka, G. (2013). Distance-based image classification: Generalizing to new classes at near-zero cost. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(11), 2624–2637.

Mignon, A., & Jurie, F. (2012). Pcca: A new approach for distance learning from sparse pairwise constraints. In *2012 IEEE conference on computer vision and pattern recognition* (pp. 2666–2672). IEEE.

Moon, T. K. (1996). The expectation–maximization algorithm. *IEEE Signal Processing Magazine*, *13*(6), 47–60.

Moutafis, P., Leng, M., & Kakadiaris, I. A. (2017). An overview and empirical comparison of distance metric learning methods. *IEEE Transactions on Cybernetics*, *47*(3), 612–625.

Niu, G., Dai, B., Yamada, M., & Sugiyama, M. (2014). Information-theoretic semi-supervised metric learning via entropy regularization. *Neural Computation*, *26*(8), 1717–1762.

Norouzi, M., Fleet, D. J., & Salakhutdinov, R. R. (2012). Hamming distance metric learning. In *Advances in neural information processing systems* (pp. 1061–1069).

Ojala, T., Pietikäinen, M., & Mäenpää, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(7), 971–987.

Paisitkriangkrai, S., Shen, C., & van den Hengel, A. (2015). Learning to rank in person re-identification with metric ensembles. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1846–1855).

Parameswaran, S., & Weinberger, K. Q. (2010). Large margin multi-task metric learning. In *Advances in neural information processing systems* (pp. 1867–1875).

Qi, G.-J., Tang, J., Zha, Z.-J., Chua, T.-S., & Zhang, H.-J. (2009). An efficient sparse metric learning in high-dimensional space via l 1-penalized log-determinant regularization. In *Proceedings of the 26th annual international conference on machine learning* (pp. 841–848). ACM.

Rosales, R., & Fung, G. (2006). Learning sparse metrics via linear programming. In *Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 367–373). ACM.

Royden, H. L., & Fitzpatrick, P. (1988). *Real analysis, Vol. 198*. Macmillan New York.

Salakhutdinov, R., & Roweis, S. T. (2003). Adaptive overrelaxed bound optimization methods. In *ICML* (pp. 664–671).

Schultz, M., & Joachims, T. (2004). Learning a distance metric from relative comparisons. *Advances in Neural Information Processing Systems (NIPS)*, 41.

Shalev-Shwartz, S., & Ben-David, S. (2014). Understanding machine learning. from theory to algorithms, Lirias.kuleuven.be.

Shalev-Shwartz, S., Singer, Y., & Ng, A. Y. (2004). Online and batch learning of pseudo-metrics. In *Proceedings of the twenty-first international conference on machine learning* (p. 94). ACM.

Shaw, B., Huang, B., & Jebara, T. (2011). Learning a distance metric from a network. In *Advances in neural information processing systems* (pp. 1899–1907).

Shen, C., Kim, J., Liu, F., Wang, L., & Van Den Hengel, A. (2014). Efficient dual approach to distance metric learning. *IEEE Transactions on Neural Networks and Learning Systems*, *25*(2), 394–406.

Shen, C., Kim, J., Wang, L., & Hengel, A. (2009). Positive semidefinite metric learning with boosting. In *Advances in neural information processing systems* (pp. 1651–1659).

Sohn, K. (2016). Improved deep metric learning with multi-class n-pair loss objective. In *30th conference on neural information processing systems*.

Song, H. O., Xiang, Y., Jegelka, S., & Savarese, S. (2016). Deep metric learning via lifted structured feature embedding. In *Computer vision and pattern recognition* (pp. 4004–4012).

Tao, D., Jin, L., Wang, Y., Yuan, Y., & Li, X. (2013). Person re-identification by regularized smoothing kiss metric learning. *IEEE Transactions on Circuits and Systems for Video Technology*, *23*(10), 1675–1685.

Torresani, L., & Lee, K.-c. (2006). Large margin component analysis. In *Advances in neural information processing systems* (pp. 1385–1392).

Verma, Y., & Jawahar, C. (2012). Image annotation using metric learning in semantic neighbourhoods. In *Computer vision–ECCV 2012* (pp. 836–849). Springer.

Verma, N., Mahajan, D., Sellamanickam, S., & Nair, V. (2012). Learning hierarchical similarity metrics. In *2012 IEEE conference on computer vision and pattern recognition* (pp. 2280–2287). IEEE.

Wang, F. (2011). Semisupervised metric learning by maximizing constraint margin. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, *41*(4), 931–939.

Wang, J., Deng, Z., Choi, K. S., Jiang, Y., Luo, X., Chung, F. L., & Wang, S. (2016). Distance metric learning for soft subspace clustering in composite kernel space. *Pattern Recognition*, *52*(C), 113–134.

Wang, J., Do, H. T., Woznica, A., & Kalousis, A. (2011). Metric learning with multiple kernels. In *Advances in neural information processing systems* (pp. 1170–1178).

Wang, Z., Gao, S., & Chia, L.-T. (2012). Learning class-to-image distance via large margin and l1-norm regularization. In *Computer vision–ECCV 2012* (pp. 230–244). Springer.

Wang, S., Jiang, S., Huang, Q., & Tian, Q. (2012). Multi-feature metric learning with knowledge transfer among semantics and social tagging. In *2012 IEEE conference on computer vision and pattern recognition* (pp. 2240–2247). IEEE.

Wang, J., Kalousis, A., & Woznica, A. (2012). Parametric local metric learning for nearest neighbor classification. In *Advances in neural information processing systems* (pp. 1601–1609).

Wang, F., & Sun, J. (2014). Survey on distance metric learning and dimensionality reduction in data mining. *Data Mining and Knowledge Discovery*, *29*(2), 534–564.

Wang, F., & Sun, J. (2015). Survey on distance metric learning and dimensionality reduction in data mining. *Data Mining and Knowledge Discovery*, *29*(2), 534–564.

Wang, H., Wang, W., Zhang, C., & Xu, F. (2014). Cross-domain metric learning based on information theory. In *Twenty-eighth AAAI conference on artificial intelligence*.

Wang, Q., Yuen, P. C., & Feng, G. (2013). Semi-supervised metric learning via topology preserving multiple semi-supervised assumptions. *Pattern Recognition*, *46*(9), 2576–2587.

Weinberger, K. Q., Blitzer, J., & Saul, L. K. (2005). Distance metric learning for large margin nearest neighbor classification. In *Advances in neural information processing systems* (pp. 1473–1480).

Weinberger, K. Q., & Tesauro, G. (2007). Metric learning for kernel regression. In *International conference on artificial intelligence and statistics* (pp. 612–619).

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, *1*(1), 67–82.

Xing, E. P., Jordan, M. I., Russell, S., & Ng, A. Y. (2002). Distance metric learning with application to clustering with side-information. In *Advances in neural information processing systems* (pp. 505–512).

Xiong, F., Gou, M., Camps, O., & Sznaier, M. (2014). Person re-identification using kernel-based metric learning methods. In *Computer vision–ECCV 2014* (pp. 1–16). Springer.

Xu, Y., Ping, W., & Campbell, A. T. (2011). Multi-instance metric learning. In *2011 IEEE 11th international conference on data mining* (pp. 874–883). IEEE.

Yang, P., Huang, K., & Liu, C.-L. (2013). Geometry preserving multi-task metric learning. *Machine Learning*, *92*(1), 133–175.

Yang, L., Jin, R., & Sukthankar, R. (2012). Bayesian active distance metric learning, arXiv preprint arXiv:1206.5283.

Yang, L., Jin, R., Sukthankar, R., & Liu, Y. (2006). An efficient algorithm for local distance metric learning. In *AAAI, Vol. 2*.

Ying, Y., Huang, K., & Campbell, C. (2009). Sparse metric learning via smooth optimization. In *Advances in neural information processing systems* (pp. 2214–2222).

Ying, Y., & Li, P. (2012). Distance metric learning with eigenvalue optimization. *Journal of Machine Learning Research (JMLR)*, *13*(1), 1–26.

Yu, Y., Jiang, J., & Zhang, L. (2011). Distance metric learning by minimal distance maximization. *Pattern Recognition*, *44*(3), 639–649.

Yu, C. N. J., & Joachims, T. (2009). Learning structural svms with latent variables. In *International conference on machine learning* (pp. 1169–1176).

Yu, J., Wang, M., & Tao, D. (2012). Semisupervised multiview distance metric learning for cartoon synthesis. *IEEE Transactions on Image Processing*, *21*(11), 4636–4648.

Yu, J., Yang, X., Gao, F., & Tao, D. (2016). Deep multimodal distance metric learning using click constraints for image ranking. *IEEE Transactions on Cybernetics*, *pp*(99), 1–11.

Zhang, Y., Zhang, H., Nasrabadi, N. M., & Huang, T. S. (2013). Multi-metric learning for multi-sensor fusion based classification. *Information Fusion*, *14*(4), 431–440.

Zheng, W.-S., Gong, S., & Xiang, T. (2013). Reidentification by relative distance comparison. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(3), 653–668.

Zhu, P., Zhang, L., Zuo, W., & Zhang, D. (2013). From point to set: Extend the learning of distance metrics. In *Proceedings of the IEEE international conference on computer vision* (pp. 2664–2671).

Zou, P.-C., Wang, J., Chen, S., & Chen, H. (2014). Bagging-like metric learning for support vector regression. *Knowledge-Based Systems*, *65*, 21–30.