Knowledge-Based Systems 116 (2017) 152–162

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Global and local metric learning via eigenvectors

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, Beijing 100190, China
[c] Key Laboratory of Big Data Mining and Knowledge management, Chinese Academy of Sciences, Beijing 100190, China

ARTICLE INFO

Article history:
Received 14 April 2016
Revised 1 November 2016
Accepted 4 November 2016
Available online 5 November 2016

Keywords:
Metric learning
Global and local
Dimension reduction
Classification

ABSTRACT

Distance metric plays a significant role in machine learning methods(classification, clustering, etc.), especially in *k*-nearest neighbor classification(*k*NN), where the Euclidean distances are computed to decide the labels of unknown points. But Euclidean distance ignores the statistical structure which may help to measure the similarity of different inputs better. In this paper, we construct an unified framework, including two eigenvalue related methods, to learn data-dependent metric. Both methods aim to maximize the difference of intra-class distance and inter-class distance, but the optimization is considered in global view and local view respectively. Different from previous work in metric learning, our methods straight seek for equilibrium between inter-class distance and intra-class distance, and the linear transformation decomposed from the metric is to be optimized directly instead of the metric. Then we can effectively adjust the data distribution in transformed space and construct favorable regions for *k*NN classification. The problems can be solved simply by eigenvalue-decomposition, much faster than semi-definite programming. After selecting the top eigenvalues, the original data can be projected into low dimensional space, and then insignificant information will be mitigated or eliminated to make the classification more efficiently. This makes it possible that our novel methods make metric learning and dimension reduction simultaneously. The numerical experiments from different points of view verify that our methods can improve the accuracy of *k*NN classification and make dimension reduction with competitive performance.

1. Introduction

*k*NN is one of the most classic methods in pattern classification and clustering [1]. It predicts the label of an unknown point by the majority label of the point's *k* nearest neighbors. In finding these nearest neighbors, Euclidean distances will be calculated in most cases. *k*NN has obtained competitive accuracy in certain tasks despite its simple mechanism. However, the performance of *k*NN classification is restrained seriously since it depends heavily on the measurement of distance. Traditional Eculidean distance only computes the 2-norm of the difference between two vector inputs mechanically, giving the same treatment to all the attributes. It may ignore the potential statistical information or feature relations, which should be extracted to construct more advantageous neighbourhoods for *k*NN.

As a hot research topic in data mining, distance metric learning has been extensively studied [2–6] since it can improve the performance of many distance related methods in classification and clustering, including *k*-nearest neighbors(*k*NN), *k*−means, etc. An appropriate data-dependent metric can help to measure the similarity of different examples more accurately. In supervised learning, given the labels of training inputs, a desired metric is expected to make the distance of similar points smaller than dissimilar points. Based on such simple idea, many researchers have proposed different algorithms to learn a proper distance metric. These methods can be classified into two kinds: (1) Global view methods: Metric Learning with Side Information, Information-theoretic metric learning, Metric learning with Boosting, etc.; (2) Local view methods: Neighborhood component analysis, metric learning for Large margin nearest neighbor classification, etc. Metric learning has been applied in many applications, including face verification [7], image annotation [8], text classification [9]. All the methods mentioned above have obtained promising performance in classification problems. However, up to now, there are few papers paying attention to the potential affect of global and local view, let alone unifying them in a framework.

In this paper, we propose two versions of Metric Learning with EigenValue(MLEV) optimization based on global and local view, respectively. The two novel approaches, named as MLEV-G(Global version) and MLEV-L(Local version), are simply and directly constructed to meet the objectives in metric learning. For global ver-

* Corresponding author.
E-mail address: tyj@ucas.ac.cn (Y. Tian).

http://dx.doi.org/10.1016/j.knosys.2016.11.004
0950-7051/

sion, maximization of the difference between within-class distance and between-class distance is implemented to make similar points concentrate in a narrower range and dissimilar points have larger distances. In local version, the objective of global perspective is considered in neighbourhoods. For every input, its k nearest similar neighbors and $k-1$ nearest dissimilar neighbors are selected to compute local inter-class distance and intra-class distance respectively. The difference between the two kinds of distance will be then optimized to raise the number of similar points and reduce the number of dissimilar points in the neighborhoods. The original positive semi-definite matrix $M \in R^{n \times n}$ (the learned metric) can be decomposed with respect to a linear transformation $L \in R^{p \times n}$ ($p \leq n$) (n is the number of dimension), namely, $M = L^\top L$. Then we can solve the approaches simply by eigenvalue-decomposition of the Lagrange function, much faster than semi-definite programming and gradient descent. The methods can make metric learning and dimension reduction simultaneously since L can be non-square matrix. When $p < n$ is provided, the top p eigenvectors are selected to form L, then the linear transformation L projects the original data into lower dimensional space, discarding inconsequential data information. The experimental results show that our novel approaches can obtain better performance in improving kNN classification. In addition, we provide a scheme to implement the two methods properly since they are suitable for different data distribution. When the points in the same class are clustering together, MLEV-G can get better performance than MLEV-L. But MLEV-L is recommended when all the data points are scattered.

The rest of the paper is organized as follows. The background, including definitions in metric learning, metric learning methods in global and local view, is introduced in Section 2. In Section 3, the new methods are described and formulated. Section 4 makes numerical experiments to show the ability of our methods in improving kNN classification performance. The conclusions are summarized in Section 5.

2. Background

In this section, the definitions and terminologies of metric learning will be claimed and some previous related works will be introduced, including eigenvector methods based on distance and metric learning in global and local view.

2.1. Definition and terminology

For a training set with c classes

$$T = \{(x_1, y_1), \ldots, (x_m, y_m)\}, \tag{1}$$

where $(x_i, y_i) \in R^n \times \{1, 2, \ldots, c\}, i = 1, \ldots, m$ and m is the total number of samples, n is the number of features. Define the following sets

$$S = \{(x_i, x_j) | y_i = y_j\} \tag{2}$$

$$D = \{(x_i, x_l) | y_i \neq y_l\} \tag{3}$$

where S consists of data pairs combined by similar points and D denotes the set of pairs of points that are dissimilar. Metric learning aims to learn an appropriate metric M with data-information embedded to reconstruct distance relationship by

$$d_M(x_i, x_j) = \sqrt{(x_i - x_j)^\top M (x_i - x_j)} \tag{4}$$

In the new relationship, the distance between the pairs in S and D has been shrunk and/or expanded as much as possible, respectively. In the previous work, the researchers are striving to minimize

$$\sum_{(x_i, x_j) \in S} d(x_i, x_j)$$

and/or maximize

$$\sum_{(x_i, x_j) \in D} d(x_i, x_j)$$

in various formulas. In general, the new metric M should meet the following conditions [3,10]:

(1) distinguishability: $d_M(x_i, x_i) = 0$;
(2) non-negativity: $d_M(x_i, x_j) \geq 0$;
(3) symmetry: $d_M(x_i, x_j) = d_M(x_j, x_i)$;
(4) triangular inequality: $d_M(x_i, x_j) + d_M(x_i, x_k) \geq d_M(x_j, x_k)$;

These conditions are intuitive on account of the concept of distance. In our methods, we decompose M into $L^\top L$ and then

$$d_M^2(x_i, x_j) = (x_i - x_j)^\top M (x_i - x_j) \tag{5}$$

$$= (x_i - x_j)^\top L^\top L (x_i - x_j) \tag{6}$$

$$= \|Lx_i - Lx_j\|^2 \tag{7}$$

So the conditions (1), (2), (3) are satisfied naturally. For condition (4),

$$d_M(x_i, x_j) + d_M(x_i, x_k) \tag{8}$$

$$= \|Lx_i - Lx_j\|^2 + \|Lx_i - Lx_k\|^2 \tag{9}$$

$$\geq \|Lx_i - Lx_j + Lx_k - Lx_i\|^2 \tag{10}$$

$$= \|Lx_k - Lx_j\|^2 \tag{11}$$

Obviously, the learned metrics in our models are standard and can be used to measure the distance of different points in a meaningful way.

2.2. Related works

A large number of methods in metric learning have been proposed to validate the efficacy of proper data-dependent distance metric in improving the performance of kNN classification method. These methods can be classified into two categories: global view and local view. In the following paragraphs, we will first introduce some eigenvector related works which are established on the basis of distance. Then metric learning models in global and local view are analyzed and compared in detail.

2.2.1. Eigenvector related work

Eigenvector methods are proposed to seeks for linear transformation to project the original inputs into new lower-dimensional space. The linear projection can reduce the noises in the data to make the computation easier in classification, clustering, etc. In fact, the linear transformation can be regarded as learning a data-dependent metric.

Principal Component Analysis(PCA). PCA [11] is a statistical technique used to reduce dimension of high dimensional data, with the idea of distance metric learning embedded. It looks for a linear transformation to maximize the variance of the transformed points. The target problem is

$$\max_L \ \mathrm{Tr}(L^\top \Sigma^{-1} L) \tag{12}$$

$$s.t. \ \ LL^\top = I. \tag{13}$$

where Σ is the covariance matrix. The solution of the problem consists of the eigenvectors corresponding to the top eigenvalues of Σ. It can be seen that the objective function measures the total variance of the projected data, similar as maximizing the between-class distance in metric learning. PCA is widely used in data preprocessing to reduce data dimension, with the advantages of denoising and reduction of computation complexity.

Linear Discriminant Analysis(LDA). Using the information of data labels, LDA [12] maximizes the ratio of between-class distance to within-class distance, given by

$$\max_{L} \ \mathrm{Tr}\left(\frac{L^\top D_b L}{L^\top D_w L}\right) \tag{14}$$

$$s.t. \ \ LL^\top = I. \tag{15}$$

The best L is obtained from the combination of the important eigenvectors of $D_w^{-1}D_b$. The between-class distance D_b and within-class distance D_w are defined as following

$$D_b = \sum_{k=1}^{c} \mu_k \mu_k^\top \tag{16}$$

$$D_w = \sum_{k=1}^{c} \sum_{y_i=k} (x_i - \mu_k)(x_i - \mu_k)^\top \tag{17}$$

where μ_k represents the centroid of the k class. In metric learning view, LDA extracts the information of the centroid of all the classes and makes efforts in maximizing intra-class distance and minimizing inter-class distance. This kind of distance measurement seems 'coarse' and 'inaccurate'. But LDA runs with high speed since its mechanism of linear preprocessing.

PCA and LDA are two inchoate methods in exploring appropriate metric. PCA is implemented in an unsupervised way and then it can not make use of label information [13], which may restrains its performance. LDA often performs bad in pattern recognition [14,15], which may result from its improper measurement of between-class and within-class distance.

2.2.2. Metric learning in global view

Global metric learning optimizes and constrains all the criterions based on the whole dataset. The desired metric only need to consider adjusting the distance from a global view. The constraints and the objective functions are often simple and concise. It pays less attention to local structure of the data and sometimes can not extract the information completely. Popular methods in this area include metric learning with side information, information-theoretic and boosting trick based metric learning. Besides, the eigenvector related methods mentioned in the Section 2.2.1 are all in the scope of global metric learning.

Metric Learning with Side Information(MLSI). MLSI, proposed by Xing etal. [16], is an early effort in seeking for an appropriate metric. Similarity side-information is utilized to define pairwise relationships to learn a desired data-dependent metric to improve accuracy in identifying clusters. They argue that an ideal metric should satisfy that the two points in any data pair of S should be as near as possible. Meanwhile, the distance between the two points in any data pairs of D is supposed to be larger than a threshhold. The idea is embodied as a semi-definite programming(SDP) problem, constructed as following

$$\min_{M} \sum_{(x_i,x_j)\in S} d_M^2(x_i, x_j) \tag{18}$$

$$s.t. \sum_{(x_i,x_j)\in D} d_M^2(x_i, x_j) \geq 1 \tag{19}$$

$$M \succeq 0. \tag{20}$$

where $d_M^2(x_i, x_j) = \|x_i - x_j\|_M^2 = (x_i - x_j)^\top M (x_i - x_j)$. The formulation of MLSI is concise and its principle can be understood easily. However, MLSI does not make efforts in expanding intra-class distance. The information of data structure has not been extracted adequately. What's more, SDP problem is hard to be solved since its high time complexity. And extensive experiments have shown that MLSI obtains low performance in improving kNN classification [17–19].

Information-theoretic metric learning(ITML). ITML [20] is an information-theoretic method expressed by Bregman optimization problem. It minimizes the relative entropy between two multivariate Gaussian distribution, each corresponds to a target metric M and a predefined metric M_0. The entropy measures the distance of the two Mahalanobis matrices well and it can be optimized easily in light of its differentiability. To achieve the basic goal of metric learning that similar points should be close and dissimilar points should be far from each other, ITML makes the distance of similar inputs smaller than a relatively small value and dissimilar inputs larger than a sufficient large value. The optimization problem is provided as following

$$\min_{M} \ \mathrm{tr}(MM_0^{-1}) - \gamma \mathrm{logdet}(MM_0^{-1}) \tag{21}$$

$$s.t. \ \ d_M^2(x_i, x_j) \leq u, \ (x_i, x_j) \in S \tag{22}$$

$$d_M^2(x_i, x_l) \geq l, \ (x_i, x_l) \in D \tag{23}$$

where u, l are given parameters depended on the data distribution. ITML is fast and scalable since semi-definite programming and eigenvalue computation are not required. The formulation is very general and can deal with different kinds of constraints. But in practice, it is difficult to select the trade-off γ, upper bound u, lower bound l and the prior M_0, which may restrict its performance.

Metric learning with Boosting(BoostMetric). Derived from AdaBoost, BoostMetric [21,22] learns several trace-one rank-one weak metrics to gain the desired positive semi-definite matrix. The learning process is efficient and scalable. It is proposed based upon boosting technique, aiming at maximizing the relative difference between intra-class distance and inter-class distance. A set of triplets $G = \{(x_i, x_j, x_l) | (x_i, x_j) \in S, (x_i, x_l) \in D\}$ is constructed, and the primary optimization problem is formulated with applying exponential loss

$$\min_{M} \ \log\left(\sum_{r=1}^{|G|} \exp(-\rho_r)\right) + c\mathrm{Tr}(M) \tag{24}$$

$$s.t. \ \ \rho_r = < A_r, M >, r = 1, \ldots, |G| \tag{25}$$

$$M \succeq 0 \tag{26}$$

where $A_r = (x_i - x_l)(x_i - x_l)^\top - (x_i - x_j)(x_i - x_j)^\top, r = 1, \ldots, |G|$. $|G|$ is the size of the set G. BoostMetric does not need to consider finding positive semi-definite matrix but only optimize entropy maximization problem with coordinate descent or eigenvalue decomposition. But it requires a great many iterations for high-dimensional datasets.

Motivated by the idea of support vector machine, a margin-based approach called MLSVM is provided [23]. It separates different inputs by a margin, leading to quadratic semi-definite programming formulation. More SVM based methods are studied later [24–26]. KISS(keep it simple and straightforward) metric learning [27] constructs likelihood ratio test to decide whether a pair points are similar or not. Eigenanalysis is used to obtain the mahalanobis metric on the cone of positive semidefinite matrices. Some other methods in global perspective are also proposed, sparse metric for metric learning and dimension reduction [28–30], learning metric from network [31,32], metric learning with kernel framework [33].

2.2.3. Metric learning in local view

From a local point of view, metrics can be learned by optimizing and constraining the rules based on local neighborhood. Local data structure or distribution can be discovered in such framework.

Local metric learning constructs optimization problem based on local data distribution and only constrains the distance from neighborhood perspective. All the local structure information should be taken into account to meet the constraints. In fact, it possesses more sophisticated mechanism than global metric learning. And then local methods can improve the *k*NN performance better than global ones some times since *k*NN is of the principle of voting in the set of k nearest points. Classical methods contain neighborhood component analysis, large margin nearest neighbor classification.

Neighborhood component analysis(NCA). Goldberger et.al present neighborhood component analysis(NCA) [34] to learn a low-rank quadratic metric, which aims to directly optimize leave-one-out performance of *k*NN method. In NCA, every input x_i give a probability p_{ij} to another point x_j to decide the likelihood of x_j is a neighbor of x_i. The p_{ij} is defined by Euclidean distance after transformation

$$p_{ij} = \frac{\exp(-\|Lx_i - Lx_j\|^2)}{\sum_{k\neq i}\exp(-\|Lx_i - Lx_k\|^2)}, \; p_{ii} = 0. \tag{27}$$

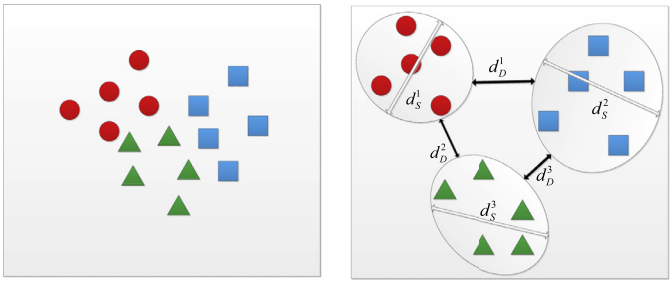
where L is the decomposition of the desired metric $M = L^\top L$. The goal of NCA is to maximize the numbers of correctly labeled points, leading to the following problem

$$\max_L \quad f(L) = \sum_i \sum_{y_j = y_i} p_{ij} \tag{28}$$

The problem can be optimized simply by gradient descent algorithm. NCA is a classic method in metric learning based on probability theory, utilizing local data information. NCA can be used to reduce data dimension since it directly optimizes the linear transformation L. However, memory overflow often happens when it deals with high-dimensional data.

Large margin nearest neighbor(LMNN). Inspired by the research on neighborhood component analysis, LMNN constructs a semi-definite programming problem to maximize margin of dissimilar points by introducing a convex hinge loss function and minimize the distance of any two close and similar points. LMNN defines a new terminology called target neighbors for every $x_i (i = 1, \ldots, m)$, which means the k nearest points with the same label as x_i. The algorithm aims to minimize the total distance between the target neighbors and x_i. Since LMNN is proposed to improve the performance of *k*NN classification, it sets the goal that the points with different labels should have larger distance to x_i than any target neighbor. The idea of LMNN leads to the following SDP problem

$$\min_M \sum_{(x_i,x_j)\in S} \eta_{ij} d_M^2(x_i, x_j) + c \sum_{\substack{(x_i,x_j)\in S \\ (x_i,x_l)\in D}} \eta_{ij}\xi_{ijl} \tag{29}$$



(a) Original (b) Transformed

Fig. 1. Geometric illustration of the mechanism of MLSI and MLEV-G. The white and black arrows denote within-class and between-class distance respectively. MLSI and MLEV-G both aim to minimize the within-class distance. In extracting the information of between-class distance, MLSI just enforces $\sum d_D^i \geq 1$, but MLEV-G seeks for a balance between $\sum d_S^i$ and $\sum d_D^i$.

$$s.t. \quad d_M^2(x_i, x_l) - d_M^2(x_i, x_j) \geq 1 - \xi_{ijl} \tag{30}$$

$$\xi_{ijl} \geq 0 \tag{31}$$

$$M \succeq 0 \tag{32}$$

where η_{ij} indicates whether x_j is a target neighbor of x_i. If the proposition is true, $\eta_{ij} = 1$, else $\eta_{ij} = 0$. Due to the extraction of local data information, LMNN performs well in practice. Unfortunately, overfitting often occurs owing to the absence of regularization term in the objective function. Several extensions on LMNN are introduced to improve LMNN, including solving LMNN more efficiently [35,36], introducing kernel into LMNN [37], multi-task version of LMNN [38]. Neighborhood repulsed metric learning [39], large margin multi-metric learning [40] are also established on the local view.

All the previous work are constructed on global or local view individually, which may limit their performance when the data distribution is unclear. Then we propose an unified framework, including two simple and straightforward models, based on global and local view respectively. The two methods are complementary to each other, considering that the information of data structure can be easier extracted from global or local perspective. Our methods can both learn data-dependent metric and reduce dimension, superior to most of the above mentioned methods.

3. Global and local metric learning with eigenvectors

In this section, we will clearly illustrate our novel approaches in metric learning and their advantages in advancing the performance of *k*NN classification.

3.1. Global version

Unlike MLSI, we expect to minimize the distance of similar points and maximize the distance of dissimilar points simultaneously. The constraints of MLSI on dissimilar points are moved into the objective function to make two-sides efforts in adjusting the original distance. The geometric illustration of MLSI and MLEV-G is displayed in Fig. 1. The principle of MLSI is shink the inter-class distance as much as possible under the condition that the total distance of dissimilar inputs is larger than 1. The key point of MLSI is to gather every class as a cluster with minimal diameter. The motivation of the global version metric learning with eigenvectors is

to find a tradeoff between inter-class distance and intra-class distance, leading to the optimization of the difference between them.

In a global view, an ideal data-dependent matrix M can be learned by the following optimization problem

$$\min_{M} \sum_{(x_i,x_j)\in S} d_M^2(x_i,x_j) - \lambda \sum_{(x_i,x_j)\in D} d_M^2(x_i,x_j) \tag{33}$$

$$s.t.\ \ M \succeq 0 \tag{34}$$

where λ is a trade-off in attuning the importance of inter-class distance and intra-class distance. Since there exist L satisfy that $M = L^\top L$, so

$$d_M^2(x_i,x_j) = \|Lx_i - Lx_j\|^2 \tag{35}$$

which verify that the distance between two points with respect to M can be converted into Euclidean distance by a linear transformation L.

Let $L = (w_1,\ldots,w_p)^\top$, where $w_i \in R^n$, $p \le n$, then

$$\|Lx_i - Lx_j\|^2 = \sum_{k=1}^{p}(w_k^\top x_i - w_k^\top x_j)^2 \tag{36}$$

$$= \sum_{k=1}^{p} w_k^\top (x_i - x_j)(x_i - x_j)^\top w_k \tag{37}$$

To avoid over-fitting, we restrict that $w_k^\top w_k = 1, k = 1,\ldots,p$. The problems (33) and (34) can be written as

$$\min_{w} \sum_{k=1}^{p} w_k^\top (Q - \lambda B) w_k \tag{38}$$

$$s.t.\ \ w_k^\top w_k = 1, k = 1,\ldots,p \tag{39}$$

where

$$Q = \sum_{(x_i,x_j)\in S} (x_i - x_j)(x_i - x_j)^\top \tag{40}$$

$$B = \sum_{(x_i,x_j)\in D} (x_i - x_j)(x_i - x_j)^\top \tag{41}$$

To solve the problem (38) and (39), we construct the following Lagrange function

$$F = \sum_{k=1}^{p} w_k^\top (Q - \lambda B) w_k - \sum_{k} \alpha_k (w_k^\top w_k - 1) \tag{42}$$

and have the KKT conditions

$$(Q - \lambda B) w_k = \alpha_k w_k \tag{43}$$

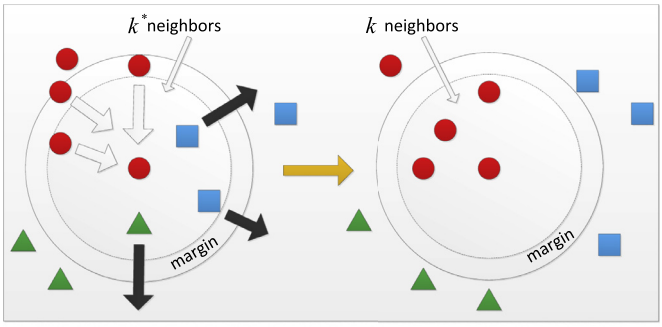
Then the dual problem is

$$\max_{w,\alpha} \sum_{k=1}^{p} \alpha_k \tag{44}$$

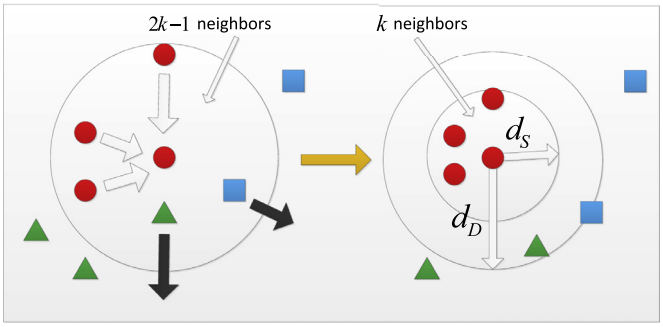
$$s.t.\ \ (Q - \lambda B) w_k = \alpha_k w_k, \tag{45}$$

$$w_k^\top w_k = 1, k = 1,\ldots,p \tag{46}$$

Notably, the solutions for $\alpha_k, k = 1,\ldots,p$ in (44)–(46) are the p largest eigenvalues of the matrix $Q - \lambda B$. Then the linear transformation L can be obtained from the combination of the eigenvectors corresponding to these eigenvalues. Since redundant or pidding data information may distort data structure, it will worsen the accuracy of kNN classification. We can weaken such negative effect by setting $p < n$ to transform raw data into lower dimensional space. The algorithm can learn data-dependent matrix and make dimension reduction simultaneously.



(a) LMNN



(b) MLEV-L

Fig. 2. Geometric illustration of LMNN and MLEV-L. In LMNN, for every x_l and its neighborhood, all the similar points will be pulled nearer to x_l and any imposters will be pushed away by a large margin. In MLEV-L, for every x_l, k nearest similar points and $k-1$ nearest dissimilar points are considered. MLEV-L balances the local inter-class distance and intra-class distance, namely, maximizing $d_D - d_S$.

3.2. Local version

The global version of metric learning with eigenvectors only utilizes global distance information. However, for kNN classification, the label of a unknown point is determined by the majority of its local neighbors. So the extraction of local data information can help to make greater improvements on kNN method. Inspired by the local idea of relative distance in LMNN, local version of metric learning with eigenvectors is presented, which shrink the inter-class distance and expand the intra-class distance in all the partitioned local regions. In Fig. 2, we explain the geometry mechanism of LMNN and our local version method. For every point x_l and its local neighborhood, LMNN pulls the target neighbors as near as possible and separate the points with different labels from x_l by a margin. Once the desired metric is learned, kNN performance can be improved obviously.

In MLEV-L, for any point x_l, define a similar neighborhood S_l^k which contains k nearest neighbors with the same label as x_l and a dissimilar neighborhood D_l^k contains $k-1$ nearest points with different labels from x_l. The two sets are not intersect with each other. A desired metric should make contribution to pulling the points in S_l^k nearer to x_l and enforcing the points in D_l^k have larger distance to x_l.

From a local view, we can learn matrix M by

$$\min_{M} \sum_{x_l} \left(\sum_{x_i \in S_l^k} d_M^2(x_l,x_i) - \eta \sum_{x_j \in D_l^k} d_M^2(x_l,x_j) \right) \tag{47}$$

$$s.t.\ \ M \succeq 0 \tag{48}$$

Similarly, the above problem can be rewritten as

$$\min_{w} \sum_{k=1}^{p} w_k^\top (H - \eta G) w_k \tag{49}$$

$$s.t. \quad w_k^\top w_k = 1, k = 1, \ldots, p \tag{50}$$

where

$$H = \sum_{x_l} \sum_{x_i \in S_l^k} (x_l - x_i)(x_l - x_i)^\top \tag{51}$$

$$G = \sum_{x_l} \sum_{x_j \in D_l^k} (x_l - x_j)(x_l - x_j)^\top \tag{52}$$

The dual problems can be obtained by constructing Lagrange function

$$\max_{w,\beta} \sum_{k=1}^{p} \beta_k \tag{53}$$

$$s.t. \quad (H - \eta G) w_k = \beta_k w_k, \tag{54}$$

$$w_k^\top w_k = 1, k = 1, \ldots, p \tag{55}$$

The above problem can be solved similarly as MLEV-G. It should be noted that MLEV-L need to look for $2k-1$ nearest neighbors using Euclidean distance. But after metric learning, the neighbors may change because of the new distance metric. To deal with such problem, L is iteratively learned until convergence in the learning procedure. Similar as MLEV-G, the local method can also make dimension reduction when $p < n$ is set.

3.3. Model comparison and analysis

We will analyze the two algorithms in an unified framework. Both aim to maximize the difference between intra-class distance and inter-class distance, but in global view and local view respectively. They extract different data information to realize the same goal, similar points are pulled closer and different labeled points are pushed further. MLEV-G and MLEV-L are inspired by MLSI and LMNN respectively, but there are essential difference between them, including the objective functions, the constraints on the inter-class distance and intra-class distance, the solution for the optimization(distance metric or linear transformation):

- In light of the decision rule in kNN classification, our approaches are proposed base on simple ideas, similar points(with the same label) should be as near as possible and dissimilar points(different labels) should have large distance in global view or local region. In this paper, we just hope to maximize the difference between intra-class distance and inter-class distance. In MLSI and LMNN, both expect to minimize the inter-class distance in the objective functions, but put the intra-class distance in the constraints, which may underestimate the effect of maximizing intra-class distance and ignore balancing inter-class and intra-class distance.
- MLSI and LMNN put different constraints on distance. MLSI enforces that the between-class distance should larger than one in macroscopic view, which can not ensure that different classes are far from each other. LMNN builds a relative distance relationship in microscopic view, namely, the distance between dissimilar points in neighborhood is one unit larger than the distance of similar points. This two kinds of constraints are both difficult to be implemented, leading to SDP problems, which are solved with high complexity. In our methods, the objective function is only subjected to the constraint that ensure the property of semi-definite positive. The constraint can be further transformed into an equality related with linear transformation, facilitating the construction of eigenvalue problem.
- Learning the original metric M with the character of semi-definite positive are hard to achieved in both MLSI and LMNN. Since any semi-definite positive matrix can be decomposed into $M = L^\top L (L \in R^{p \times n})$, making the new distance be the Euclidean distance in transformed space $x \to Lx$. Then we can establish optimization problem in terms of L and cast off the constraints on M. More importantly, our methods can make dimension reduction when $p < n$ is set. It can make it easily to deal with high-dimensional datasets and depress the adverse impact of redundant features.

The computational complexities of the two approaches both contain two parts, calculation of the intra-class distance and inter-class distance and eigenvalue decomposition. The time complexity of MLEV-G and MLEV-L is $O(m^2n^2) + O(n^3)$ and $O(Sm^2n^2) + O(Smn^3)$ respectively, where S is the iteration number in MLEV-L. It should be noted that the computational cost of MLEV-L is nearly S times larger than that of MLEV-G.

In a word, the two metric learning methods with eigenvalue optimization are different from the previous ones and they have the following advantages: (1) The formulations of MLEV-G and MLEV-L are simple and can be solved only by eigenvalue-decomposition ; (2) The two proposed methods are both optimized with respect to L but not M, making them be able to implement dimension reduction when L is not square. The mechanism of reducing dimension can depress the negative impact of noise sometimes. (3) MLEV-G and MLEV-L are constructed on the basis of global and local view respectively. They are complementary to each other, suitable for more kinds of data distribution.

4. Experiments

In this section, comprehensive numerical experiments on Benchmark datasets, Letter recognition, and dimensional reduction will be made to verify that the new proposed approaches can both improve kNN performance and make dimension reduction effectively. All the experiments are made on MATLAB 2015a(Lenovo PC, Intel Core i5, 2 cores with 3.10GHz, 8GB RAM).

4.1. Benchmark datasets

We select 17 benchmark datasets from the UCI Machine Learning Repository to evaluate the global and local version of our new algorithm, **MLEV-G** and **MLEV-L**. The selected datasets are all low dimensional with different sizes and classes, listed in Table 1. Every dataset is randomly splitted with 70% and 30% of the instances, used for training and test severally. We compare our new methods with five previous classical methods, including ***k*NN** with the standard Euclidean distance as the baseline method and **MLSI, ITML, LMNN, BoostMetric**. All the methods have the same partitions on every dataset.

For all the methods, k is set to be 3 in kNN classification. And the test error is used as the index to assess the performance of these methods:

$$\text{Test error} = \sum_{i=1}^{n_e} [\![y_i \neq y_i^*]\!] / n_e$$

where $[\![A]\!]$ is an indicator function, its value is 1 if A is true, otherwise 0. n_e is the number of test points. The classification results are summarized in Table 1. All the errors are obtained from the mean of the results on 10 times runs. The experimental settings for all the methods, except the parameter-free algorithm MLSI, are as following: In ITML, γ is selected from searching the set

Table 1
Error rates of different metric learning methods on Benchmark Datasets.

Dataset	inst. × attr.	Eculidean	MLSI	ITML	LMNN	BoostMetric	MLEV-G	MLEV-L
WPBC	198 × 33	26.27 ± 3.77	28.47 ± 5.10	28.64 ± 3.61	27.46 ± 4.65	27.80 ± 5.07	24.58 ± 3.12	**24.24** ± 3.75
Sonar	208 × 60	18.87 ± 6.23	25.97 ± 5.51	21.61 ± 7.65	17.10 ± 5.55	**14.52** ± 4.93	15.48 ± 4.82	18.71 ± 6.46
Spectf	267 × 44	27.50 ± 4.71	22.38 ± 3.88	28.38 ± 4.08	**22.13** ± 4.13	23.50 ± 3.90	26.00 ± 4.36	26.25 ± 4.41
Heart	270 × 13	20.99 ± 2.85	21.11 ± 3.47	22.35 ± 2.94	20.86 ± 3.21	21.73 ± 5.15	20.12 ± 2.61	**18.64** ± 3.21
Hungarian	294 × 13	21.25 ± 2.84	**19.77** ± 3.83	22.50 ± 3.07	20.68 ± 3.63	20.57 ± 2.96	21.48 ± 3.23	20.57 ± 2.54
Heartc	303 × 13	21.33 ± 2.61	22.67 ± 3.36	23.22 ± 3.33	21.11 ± 3.05	20.67 ± 2.04	21.33 ± 3.39	**18.11** ± 2.67
Dermatology	366 × 34	**2.57** ± 0.58	9.45 ± 3.41	2.75 ± 0.86	3.67 ± 1.22	3.49 ± 1.61	**2.57** ± 0.72	2.75 ± 0.75
WDBC	569 × 30	**3.88** ± 1.12	3.94 ± 1.21	4.47 ± 1.80	4.06 ± 0.70	3.94 ± 1.11	**3.88** ± 1.12	**3.88** ± 1.12
Blood	748 × 4	26.52 ± 2.31	31.47 ± 16.5	26.70 ± 1.83	27.01 ± 2.21	25.98 ± 2.67	**24.02** ± 2.25	24.87 ± 1.44
Pima	768 × 8	26.13 ± 1.97	27.57 ± 2.42	28.30 ± 2.79	25.57 ± 1.58	26.35 ± 2.18	**25.48** ± 1.52	26.65 ± 2.29
German	1000 × 20	27.67 ± 2.50	28.10 ± 2.20	27.37 ± 2.51	27.37 ± 2.66	27.87 ± 2.37	27.67 ± 2.65	**26.93** ± 2.28
Parkinson	1040 × 25	34.97 ± 2.19	37.79 ± 2.49	34.97 ± 2.67	34.29 ± 1.93	**33.37** ± 2.93	34.78 ± 2.03	34.55 ± 1.86
Iris	150 × 4	4.67 ± 2.21	6.00 ± 4.45	4.67 ± 2.66	**3.33** ± 2.16	4.00 ± 2.04	4.44 ± 2.34	4.00 ± 2.30
Thyroid	215 × 5	5.78 ± 2.34	**4.06** ± 1.68	4.38 ± 2.42	6.41 ± 2.80	4.22 ± 2.45	5.63 ± 2.35	6.09 ± 2.60
Glass	214 × 9	32.03 ± 7.19	33.75 ± 6.47	39.53 ± 10.83	**30.63** ± 4.90	32.19 ± 4.55	31.41 ± 7.23	31.09 ± 6.89
Vowel	528 × 10	8.16 ± 2.98	10.63 ± 2.60	10.76 ± 3.43	9.56 ± 3.67	7.47 ± 3.68	7.47 ± 2.58	**7.34** ± 2.18
Segment	2310 × 19	3.95 ± 0.59	5.61 ± 0.85	3.71 ± 1.13	2.94 ± 0.53	**2.90** ± 0.70	3.81 ± 0.64	3.95 ± 0.59

Table 2
Average CPU time(seconds) of different metric learning methods.

Dataset	MLSI	ITML	LMNN	BoostMetric	MLEV-G	MLEV-L
WPBC	0.57	0.05	0.06	0.96	0.02	0.23
Sonar	17.91	0.08	0.06	0.84	0.03	0.44
Spectf	0.92	0.06	0.07	0.86	0.03	0.52
Heart	0.96	0.04	0.10	0.01	0.02	0.31
Hungarian	1.06	0.04	0.10	0.22	0.02	0.34
Heartc	1.20	0.04	0.11	0.01	0.02	0.36
Dermatology	17.09	0.05	0.03	0.62	0.04	0.73
WDBC	4.73	0.05	0.12	0.93	0.06	0.77
Blood	23.40	0.04	0.44	0.01	0.05	0.90
Pima	7.80	0.04	0.52	0.01	0.06	1.18
German	12.23	0.05	0.62	0.05	0.10	2.38
Parkinson	77.02	0.05	0.84	1.49	0.12	1.54
Iris	0.43	0.05	0.02	0.01	0.01	0.14
Thyroid	0.65	0.05	0.06	0.04	0.02	0.20
Glass	0.97	0.07	0.04	0.18	0.02	0.13
Vowel	5.94	0.25	0.80	0.02	0.04	0.66
Segment	204.22	0.15	2.70	1.30	0.30	5.26

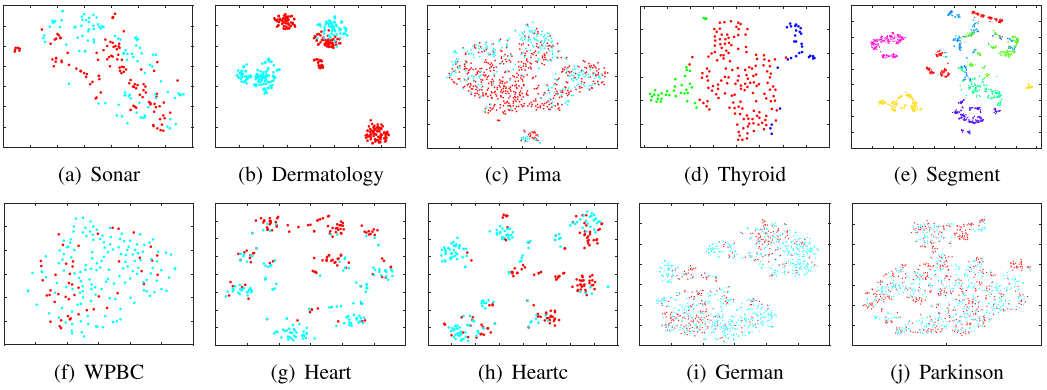
$\{10^{-4}, \ldots, 10^{4}\}$; The settings for LMNN follow [14]; The trade-off $v = 10^{-7}$ and the maximum iterations is 500 in BoostMetric. For MLEV-G and MLEV-L, the penalty parameter λ, η are both chosen from $\{10^{-6}, \ldots, 10^{3}\}$. It is claimed that MLEV-G and MLEV-L with non-square L is of benefit to denoising. The dimensional parameter p is assigned to the multiplication of 0.9 and the feature numbers, which means that the top 90% significant data information is used to make classification. In Table 1, MLEV-L obtains the lowest error on 6 out of 17 datasets. And MLEV-G performs only next to MLEV-L. Table 2 shows the average training time of these methods. It is can be seen that MLEV-G is the fastest algorithm since it only need to solve eigenvalue problems using global information. ITML and LMNN are a little slower than MLEV-G. MLEV-L need search k nearest neighbors, resulting in more training time than the global version method.

Next, we make Wilcoxon test [41,42] to examine the statistical significance, with a significance level of 0.1. Each pair of methods is selected to make comparison on each dataset. For two methods, A and B, the significantly better one will get 1 point and the other gets 0. If there is no significant difference between A and B, each gets 0.5 point. We report the total points of every method in the Table 3. MLEV-L gets the highest score with 59 points, followed by BoostMetric and MLEV-G. It is verified that our framework can obtain consistent performance in classification.

Comparing of MLEV-G and MLEV-L, they perform different on different datasets. It is inferred that global and local view methods are fit for distinct data distribution. To prove our conjecture, we use t-SNE [43] to visualize data distribution in a two-dimensional map. Ten datasets are selected, including Sonar, Dermatology, Pima, Thyroid, Segment, WPBC, Heart, Heartc, German, Parkinson. MLEV-G performs better on the former five datasets and MLEV-L obtains lower error rates on the latter five. The maps of data distribution are shown in the Fig. 3. In the former five maps, data points in the same class cluster together into one or few clusters. Then in a local view, every point has nearly zero dissimilar neighbors, so MLEV-L cannot make most use of the data information. In the latter five maps, the examples scatter loosely. In a global view, it may be very hard to find a metric to transform the data into disjoint clusters. MLEV-L is more suitable to deal with such problems. We give instructions on how to use our model: (1)Use t-SNE to map data into a 2D plane; (2) If the data points distribute in a disor-

Table 3
Total score of Wilcoxon significance test.

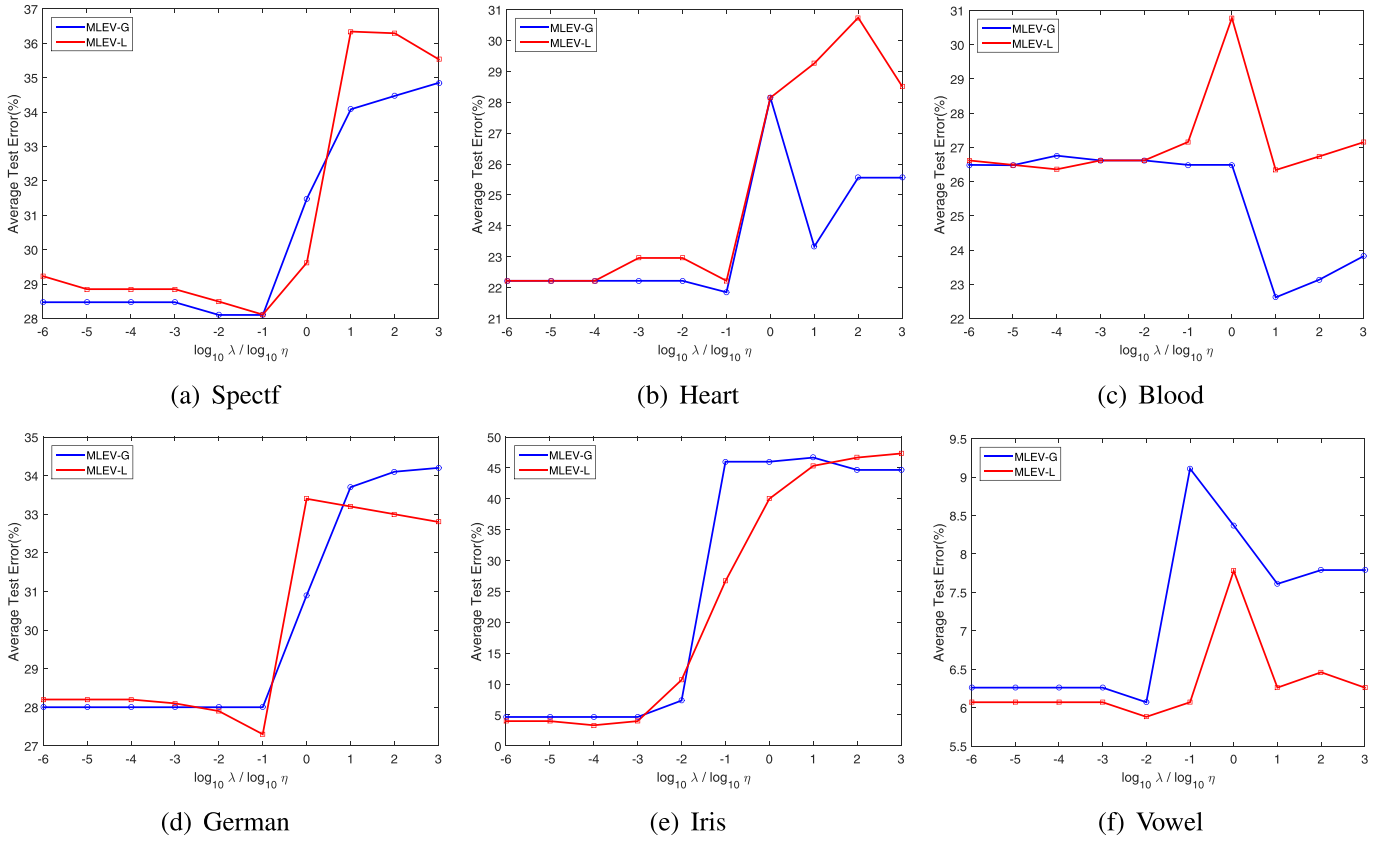
Methods	Score
Eculidean	51.0
MLSI	37.0
ITML	40.0
LMNN	54.5
BoostMetric	58.5
MLEV-G	57.0
MLEV-L	59.0



(a) Sonar (b) Dermatology (c) Pima (d) Thyroid (e) Segment

(f) WPBC (g) Heart (h) Heartc (i) German (j) Parkinson

Fig. 3. Data distribution in 2D plane.



(a) Spectf (b) Heart (c) Blood

(d) German (e) Iris (f) Vowel

Fig. 4. Influence of penalty parameters.

derly condition, MLEV-L is recommended to implement, otherwise MLEV-G.

We also investigate the influence of the penalty parameters in the method. Five-fold cross validation and 90% feature selection are made on 6 datasets. The parameters are varying from 10^{-6} to 10^{3} and the corresponding test errors are plotted in Fig. 4. MLEV-G and MLEV-L have similar trends with the variation of the parameters. The best performance is always achieved when the penalties near 1. From 10^{-6} to 10^{0}, the test error declines slightly. When the parameter is larger than 10, the error surges.

The above results prove that: (1) MLEV-G and MLEV-L can improve kNN classification with competitive performance, which verify that contracting inter-class distance and expanding intra-class distance are both important to form favorable neighborhood for the projected data points; (2) Metric learning with eigenvalue optimization is much faster than semi-definite programming; (3) Learning with Non-square linear transformation L is helpful in denoising; (3)Metric learning method constructed on global or local view is suitable for different data distribution.

4.2. Letter recognition

Identifying a number or a letter, formed by a lot of black-and-white rectangular pixel, is a common problem in machine learning. The distance between different letters or numbers can be designed by metric learning. A letter recognition dataset(download from the UCI machine learning datasets) is selected to test the ability of our new methods. The task is to identify each of the pixel set as one of the 26 capital letters in the English alphabet. The images are displayed in 20 different fonts and every letter within these 20 fonts is randomly distorted to produce a file of 20,000 unique data points. The features of each point are transformed into 16 numeri-

Table 4
Error rates on letter recognition.

Dataset	Eculidean	ITML	LMNN	BoostMetric	MLEV-G	MLEV-L
letter-5000	11.24 ± 0.55	11.24 ± 1.14	10.15 ± 0.38	7.37 ± 0.86	**7.28** ± 0.62	8.56 ± 0.21
letter-10000	7.28 ± 0.30	8.33 ± 0.61	6.06 ± 0.48	**4.23** ± 0.17	4.58 ± 0.50	5.36 ± 0.30
letter-15000	5.49 ± 0.37	6.37 ± 0.64	4.62 ± 0.21	3.60 ± 0.39	**3.60** ± 0.21	4.23 ± 0.28
letter-20000	4.70 ± 0.25	5.15 ± 0.26	3.86 ± 0.18	3.06 ± 0.26	**2.99** ± 0.18	3.60 ± 0.27

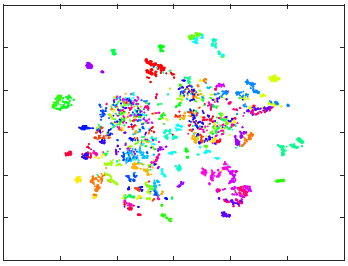


Fig. 5. The data distribution of **letter1**.

cal attributes which are then scaled into a range of integer values from 0 to 15. We randomly select 5000, 10,000, 15,000 instances from the original datasets, respectively. Since MLSI can not deal with large scale datasets, ITML, LMNN and BoostMetric are selected to make comparison with our methods. The parameter settings and the datasets partitions are the same as the Section 4.1. The test errors are got from the average values of 5 times random runs. From the Table 4, we can see that MLEV-G and MLEV-L consistently improve *k*NN classification performance on letter recognition. MLEV-G obtains best results on 3 datasets. The data distribution on 2D plane is shown in the Fig. 5. In fact, each class distributes in several clusters and these clusters are favorable to MLEV-G since the neighborhood of every point contains nearly no dissimilar points.

4.3. Dimension reduction

In the following, we will further explore the capability of our methods in dimension reduction. We compare our algorithms with three metric learning related dimensional reduction methods, **PCA, LDA** and **NCA**. First, to clearly and directly understand the performance of dimension reduction, four low dimensional datasets are chosen to be projected into 2D space by the five methods. The datasets are randomly partitioned with 70% and 30% patterns. The former part of the datasets is used for training to learn the linear transformation L, and then all the instances will be projected into 2D plane. 3−NN classification is adopted to get the test errors. Figs. 6, 7, 8, 9 shows the data distribution and test errors after projection. MLEV-L makes the best transformation on three datasets.

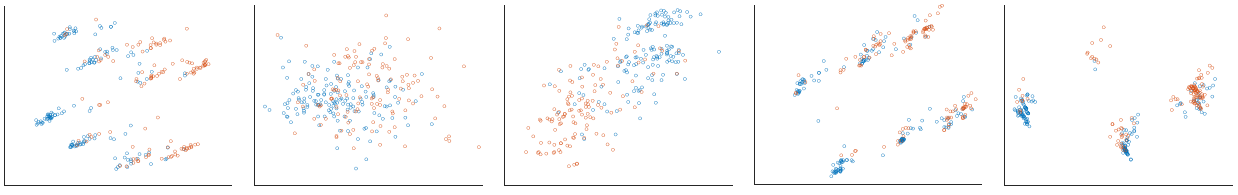
However, in real applications, the datasets are always appeared with high dimensional, which will bring the trouble of curse of dimensionality or memory overflow. For example, face verification and image recognition raise great challenges for metric learning since they are characterized by large variations including gender, expressions, pixels, etc. Meanwhile, such problems need a data-dependent metric to improve the performance of classification or clustering. Next, we use our methods to make dimension reduction for high-dimensional datasets(NCA rans out of memory easily on large datasets). The classification error of *k*NN on transformed data is the index of comparison. Four datasets from UCI , **Yaleface, Isolet, Ads, Mnist**, are selected to make experiments. The charactertics are listed in the Table 5. We reduce the dimension of all the datasets to 20, 50, 100, 150, 200, respectively. The *k*NN classification errors are reported in the Table 6 and Fig. 10. The black horizontal solid line in Fig. 10 denotes the *k*NN classification error on the primary datasets. Our methods obtain consistent performance in dimension reduction, comparable to or even better than PCA and LDA on most target dimension(In Minst, LDA cannot make

Table 5
Information of large datastes.

Dataset	Instance	Attribute	Class	Feature pattern
Yaleface	165	1024	15	Face image
Isolet	6238	617	26	Spoken letter
Ads	3279	1555	2	Image, text
Mnist	10,000	778	10	Handwritten digits

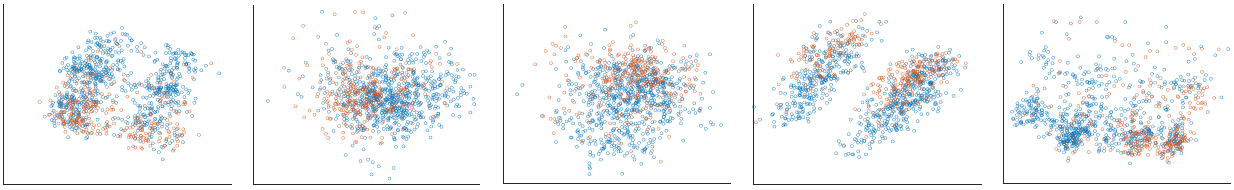
Table 6
Error rates after dimension reduction.

Dataset	Target Dim.	PCA	LDA	MLEV-G	MLEV-L
Yaleface	20	**26.53**	79.59	44.90	38.78
	50	**26.53**	77.55	30.61	32.65
	100	30.61	77.55	**28.57**	36.73
	150	30.61	77.55	**28.57**	36.73
	200	30.61	77.55	**28.57**	36.73
Isolet	20	**27.19**	91.65	29.98	32.76
	50	22.70	91.65	**19.70**	21.20
	100	20.99	87.37	**17.13**	18.84
	150	21.41	85.87	**14.78**	15.85
	200	21.63	85.01	**16.06**	**15.85**
Ads	20	**4.18**	9.97	5.49	4.27
	50	**4.17**	9.97	5.19	**4.27**
	100	**3.56**	9.97	4.58	**3.97**
	150	**3.97**	9.97	**3.97**	4.78
	200	4.48	9.97	**4.37**	5.29
Mnist	20	**4.6**		5.43	6.00
	50	**4.33**		5.37	**4.37**
	100	**4.43**		7.77	**4.37**
	150	**4.70**		10.97	**4.77**
	200	**4.83**		14.40	**4.87**



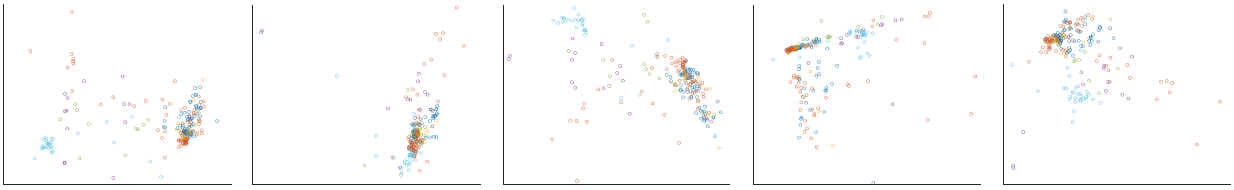
(a) PCA(28.4%) (b) LDA(45.7%) (c) NCA(30.9%) (d) MLEV-G(29.6%) (e) MLEV-L(21.0%)

Fig. 6. Heart(dim=13,error=24.7%).



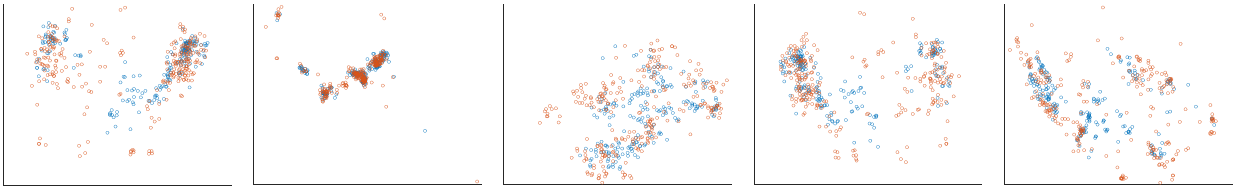
(a) PCA(36.0%) (b) LDA(35.7%) (c) NCA(32%) (d) MLEV-G(29.0%) (e) MLEV-L(27.3%)

Fig. 7. German(dim=20,error=26.0%).



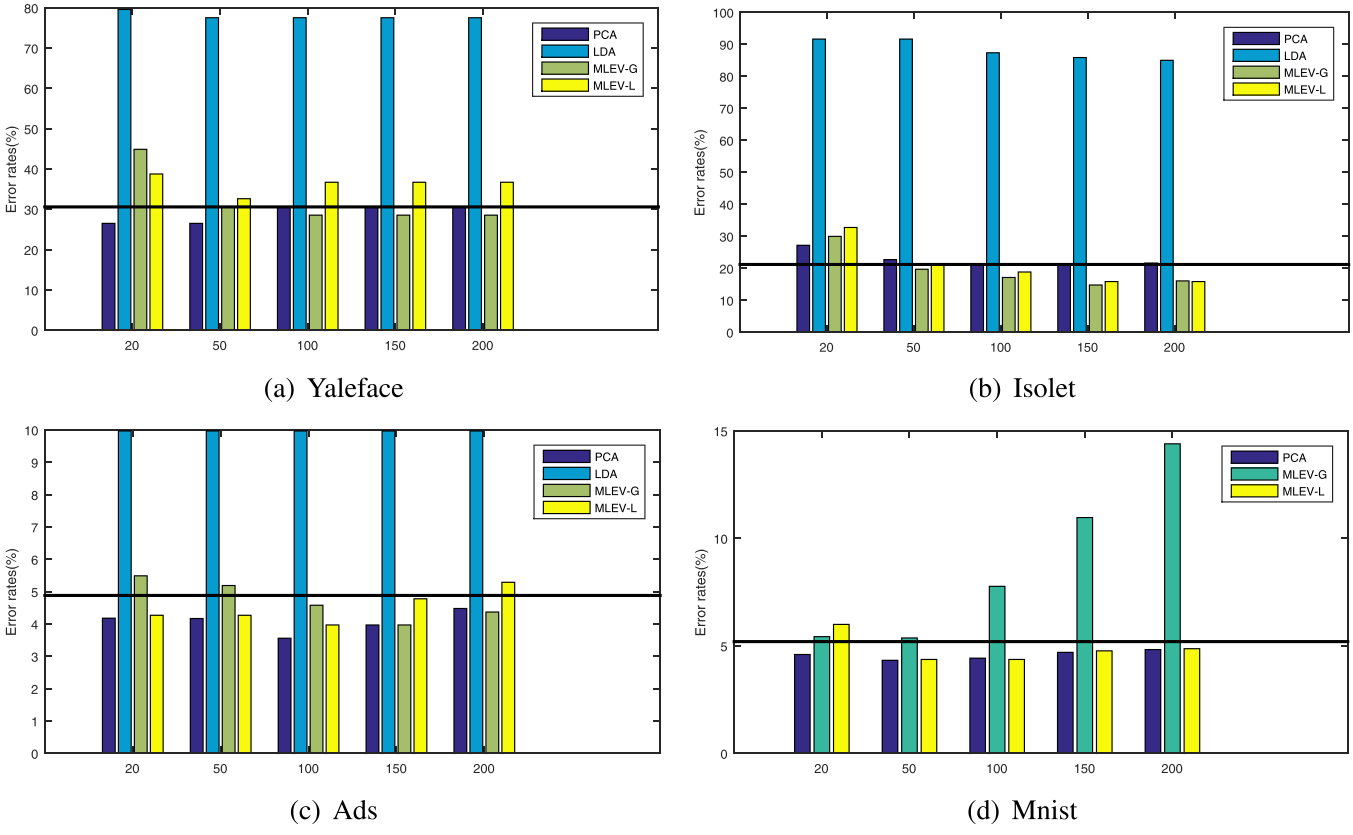
(a) PCA(34.4%) (b) LDA(43.8%) (c) NCA(34.4%) (d) MLEV-G(29.7%) (e) MLEV-L(32.8%)

Fig. 8. Glass(dim=9,error=29.7%).



(a) PCA(43.7%) (b) LDA(43.0%) (c) NCA(34.5%) (d) MLEV-G(40.8%) (e) MLEV-L(31.7%)

Fig. 9. Clean1(dim=166, error=25.4%).



(a) Yaleface (b) Isolet

(c) Ads (d) Mnist

Fig. 10. Dimension reduction for high-dimensional datasets.

dimension reduction due to the sparsity of Mnist). LDA performs much worse than other methods.

5. Conclusions

In this paper, we propose two metric learning methods, MLEV-G and MLEV-L, extracting global and local information respectively, to overcome the drawback of Euclidean distance computed in traditional *k*NN classification that overlooked the statistical information. The global version method aims to maximize the difference between global intra-class distance and inter-class distance, looking for a tradeoff between shrinking the distance of similar points and expanding the distance of dissimilar points. In local view, the neighborhood-level difference between intra-class distance and inter-class distance is optimized. The two methods are both constructed with respect to the linear transformation but not the original metric, resulting in learning metric and reducing dimension synchronously. The optimization problems can be solved by eigenvalue decomposition directly with much faster speed than SDP. Numerical experiments demonstrates that our methods can obtain competitive results in both improving the performance of *k*NN classification and reducing dimension.

Acknowledgement

This work has been partially supported by grants from National Natural Science Foundation of China (Nos.61472390, 11271361, 71331005, and 11226089), Major International (Regional) Joint Research Project (No. 71110107026) and the Beijing Natural Science Foundation (No.1162005).

References

[1] T.M. Cover, P.E. Hart, Nearest neighbor pattern classification, Inf. Theory IEEE Trans. 13 (1) (1967) 21–27.
[2] L. Yang, R. Jin, Distance metric learning: a comprehensive survey, Michigan State Universiy 2 (2006).
[3] F. Wang, J. Sun, Survey on distance metric learning and dimensionality reduction in data mining, Data Min. Knowl. Discov. 29 (2) (2014) 534–564.
[4] A. Bellet, A. Habrard, M. Sebban, A survey on metric learning for feature vectors and structured data, Technical report, Available: https://arxiv.org/pdf/1306.6709.pdf, 2013.
[5] B. Kulis, Metric learning: a survey, Found. Trends Mach. Learn. 5 (4) (2012) 287–364.
[6] P. Moutafis, M. Leng, I.A. Kakadiaris, An overview and empirical comparison of distance metric learning methods (2016).
[7] M. Guillaumin, J. Verbeek, C. Schmid, Is that you? metric learning approaches for face identification, in: Computer Vision, 2009 IEEE 12th International Conference on, IEEE, 2009, pp. 498–505.
[8] Y. Verma, C. Jawahar, Image annotation using metric learning in semantic neighbourhoods, in: Computer Vision–ECCV 2012, Springer, 2012, pp. 836–849.
[9] G. Lebanon, Metric learning for text documents, Pattern Anal. Mach. Intell. IEEE Trans. 28 (4) (2006) 497–508.
[10] H.L. Royden, P. Fitzpatrick, Real Analysis, volume 198, Macmillan New York, 1988.
[11] S. Wold, K. Esbensen, P. Geladi, Principal component analysis, Chemometrics Intell. Lab. Syst. 2 (1–3) (1987) 37–52.
[12] S. Balakrishnama, A. Ganapathiraju, Linear discriminant analysis-a brief tutorial, Inst. Sig. Inf. Process. 18 (1998).
[13] J. Shlens, A tutorial on principal component analysis, http://www.cs.cmu.edu/~elaw/papers/pca.pdf.
[14] K.Q. Weinberger, L.K. Saul, Distance metric learning for large margin nearest neighbor classification, J. Mach. Learn. Res. 10 (2009) 207–244.
[15] J. Yang, A.F. Frangi, J.-y. Yang, D. Zhang, Z. Jin, Kpca plus lda: a complete kernel fisher discriminant framework for feature extraction and recognition, IEEE Trans. Pattern Anal. Mach. Intell. 27 (2) (2005) 230–244.
[16] E.P. Xing, M.I. Jordan, S. Russell, A.Y. Ng, Distance metric learning with application to clustering with side-information, in: Advances in Neural Information Processing Systems, 2002, pp. 505–512.
[17] A. Globerson, S.T. Roweis, Metric learning by collapsing classes, in: Advances in Neural Information Processing Systems, 2005, pp. 451–458.
[18] R. Jin, S. Wang, Y. Zhou, Regularized distance metric learning: theory and algorithm, in: Advances in Neural Information Processing Systems, 2009, pp. 862–870.
[19] Y. Ying, P. Li, Distance metric learning with eigenvalue optimization, J. Mach. Learn. Res. 13 (Jan) (2012) 1–26.
[20] J.V. Davis, B. Kulis, P. Jain, S. Sra, I.S. Dhillon, Information-theoretic metric learning, in: Proceedings of the 24th International Conference on Machine learning, ACM, 2007, pp. 209–216.
[21] C. Shen, J. Kim, L. Wang, A. Hengel, Positive semidefinite metric learning with boosting, in: Advances in Neural Information Processing Systems, 2009, pp. 1651–1659.
[22] C. Shen, J. Kim, L. Wang, A. Van Den Hengel, Positive semidefinite metric learning using boosting-like algorithms, J. Mach. Learn. Res. 13 (1) (2012) 1007–1036.
[23] N. Nguyen, Y. Guo, Metric learning: a support vector approach, in: Machine Learning and Knowledge Discovery in Databases, Springer, 2008, pp. 125–136.
[24] N. Zaidi, D. Squire, Svms and data dependent distance metric, in: Image and Vision Computing New Zealand (IVCNZ), 2010 25th International Conference of, IEEE, 2010, pp. 1–7.
[25] W. Zuo, F. Wang, D. Zhang, L. Lin, Y. Huang, D. Meng, L. Zhang, Iterated support vector machines for distance metric learning, https://arxiv.org/pdf/1502.00363v1.pdf.
[26] C. Luo, M. Li, H. Zhang, F. Wang, D. Zhang, W. Zuo, Metric learning with relative distance constraints: a modified SVM approach, in: Intelligent Computation in Big Data Era, Springer, 2015, pp. 242–249.
[27] M. Koestinger, M. Hirzer, P. Wohlhart, P.M. Roth, H. Bischof, Large scale metric learning from equivalence constraints, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE, 2012, pp. 2288–2295.
[28] R. Rosales, G. Fung, Learning sparse metrics via linear programming, in: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2006, pp. 367–373.
[29] Y. Ying, K. Huang, C. Campbell, Sparse metric learning via smooth optimization, in: Advances in Neural Information Processing Systems, 2009, pp. 2214–2222.
[30] Y. Hong, Q. Li, J. Jiang, Z. Tu, Learning a mixture of sparse distance metrics for classification and dimensionality reduction, in: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE, 2011, pp. 906–913.
[31] B. Shaw, B. Huang, T. Jebara, Learning a distance metric from a network, in: Advances in Neural Information Processing Systems, 2011, pp. 1899–1907.
[32] X. Cai, C. Wang, B. Xiao, X. Chen, J. Zhou, Deep nonlinear metric learning with independent subspace analysis for face verification, in: Proceedings of the 20th ACM international conference on Multimedia, ACM, 2012, pp. 749–752.
[33] F. Wang, W. Zuo, L. Zhang, D. Meng, D. Zhang, A kernel classification framework for metric learning, Neural Netw. Learn. Syst. IEEE Trans. 26 (9) (2015) 1950–1962.
[34] J. Goldberger, G.E. Hinton, S.T. Roweis, R. Salakhutdinov, Neighbourhood components analysis, in: Advances in Neural Information Processing Systems, 2004, pp. 513–520.
[35] K. Park, C. Shen, Z. Hao, J. Kim, Efficiently learning a distance metric for large margin nearest neighbor classification, in: Twenty-Fifth AAAI Conference on Artificial Intelligence, 2011.
[36] K.Q. Weinberger, L.K. Saul, Fast solvers and efficient implementations for distance metric learning, in: Proceedings of the 25th International Conference on Machine Learning, ACM, 2008, pp. 1160–1167.
[37] L. Torresani, K.-c. Lee, Large margin component analysis, in: Advances in Neural Information Processing Systems, 2006, pp. 1385–1392.
[38] S. Parameswaran, K.Q. Weinberger, Large margin multi-task metric learning, in: Advances in Neural Information Processing Systems, 2010, pp. 1867–1875.
[39] J. Lu, X. Zhou, Y.-P. Tan, Y. Shang, J. Zhou, Neighborhood repulsed metric learning for kinship verification, Pattern Anal. Mach. Intell. IEEE Trans. 36 (2) (2014) 331–345.
[40] J. Hu, J. Lu, J. Yuan, Y.-P. Tan, Large margin multi-metric learning for face and kinship verification in the wild, in: Computer Vision–ACCV 2014, Springer, 2014, pp. 252–267.
[41] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (Jan) (2006) 1–30.
[42] S. Garcia, F. Herrera, An extension on "statistical comparisons of classifiers over multiple data sets"for all pairwise comparisons, J. Mach. Learn. Res. 9 (Dec) (2008) 2677–2694.
[43] M.L. van der, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (Nov) (2008) 2579–2605.