Semi-supervised Coefficient-Based Distance Metric Learning

Zhangcheng Wang, Ya Li, and Xinmei Tian(✉)

CAS Key Laboratory of Technology in Geo-spatial Information Processing and Application System, University of Science and Technology of China, Hefei 230027, China
{wzc1,muziyiye}@mail.ustc.edu.cn, xinmei@ustc.edu.cn

Abstract. Distance metric learning plays an important role in real-world applications, such as image classification and clustering. Previous works mainly learn a distance metric through learning a Mahalanobis metric or learning a linear transformation. In this paper, we propose to learn a distance metric from a new perspective. We first randomly generate a set of base vectors and then learn a linear combination of these vectors to approximate the target metric. Compared with previous distance metric learning methods, we only need to learn the coefficients of these base vectors instead of learning the target metric or the linear transformation. Consequently, the number of variables needed to be determined is the same as the number of base vectors, which is irrelevant to the dimension of the data. Furthermore, considering the situation that labeled samples are insufficient in some cases, we extend our proposed distance metric learning method into a semi-supervised learning framework. Additionally, an optimization algorithm is proposed to accelerate training of our proposed methods. Experiments are conducted on several datasets and the results demonstrate the effectiveness of our proposed methods.

Keywords: Distance metric learning · Semi-supervised learning · Non-smoothed function optimization

1 Introduction

Recent years has witnessed the rapid development of machine learning. As one of the most important branches of machine learning, distance metric learning has been widely used in various real-world applications, such as clustering [1], classification [8,12] and retrieval [6]. In traditional KNN classification problem, Euclidean distance is used to evaluate the similarity between different samples. However, Euclidean distance is hard to explore the intrinsic statistical features which might be estimated from the training data. Considering this drawback of Euclidean distance, distance metric learning methods are proposed to better measure the distribution of the training data.

D. Liu et al. (Eds.): ICONIP 2017, Part I, LNCS 10634, pp. 586–596, 2017.
https://doi.org/10.1007/978-3-319-70087-8_61

Previous distance metric learning can be conducted through learning a linear transformation $\boldsymbol{x}_i \rightarrow \mathbf{L}\boldsymbol{x}_i$ or equally learning a Mahalanobis metric $\mathbf{M} = \mathbf{L}\mathbf{L}^T$. Most of the previous works learn a Mahalanobis metric directly and significantly improve the performance of KNN classification, such as relevant component analysis (RCA) [2], principal component analysis (PCA) [9], linear discriminant analysis (LDA) [5], discriminative component analysis(DCA) [7], information-theoretic metric learning (ITML) [4], regularized distance metric learning (RDML) [8], distance metric learning of large margin nearest neighbor (LMNN) [12] and regularized large margin distance metric learning (RLMM) [10]. However, a Mahalanobis metric is usually time consuming to optimize. There are two main reasons. First, the Mahalanobis metric is a positive semi-definite metric. The optimization of a problem with a positive semi-definite constraint takes much time to project the target metric onto a positive semi-definite cone. Second, supposing the dimension of input feature is d, Mahalanobis metric has d^2 variables to be determined. The amount of variables increases dramatically when facing high dimensional data. These two drawbacks cause much difficulty in controlling the complexity of the method.

In this paper, we propose a novel distance metric learning method to overcome the above two drawbacks. We first generate a set of base vectors randomly and then learn a linear combination of these base vectors to approximate the target metric. Consequently, the number of variables we need to learn is the same as the number of random base vectors. The number of variables is irrelevant to the dimension of data and we can easily control the complexity of our method by adjusting the number of random base vectors. Additionally, we are unnecessary to project the target metric onto a positive semi-definite cone. Simialr idea has also been utilized in decomposition-based transfer distance metric learning (DTDML) [11]. However, DTMDL first learns the source metrics from additional data of other domains and then decomposes the metrics into base vectors which might be used to form the target metric. This process is time consuming and the source domains are usually hard to get in real-world applications. Compared with DTMDL, our proposed coefficient-based distance metric learning (CDML) just needs to randomly generate the base vectors which leads to better generalization ability.

Considering the fact that labeled data are difficult to get in real-world applications [15], some semi-supervised distance metric learning methods [1,6,10,13] have been proposed to handle this situation. Take this into consideration, we propose a novel method to explore valuable information from unlabeled data and extend our proposed method into a novel semi-supervised framework (S-CDML). An optimization algorithm is proposed to accelerate the training process. Additionally, we conduct various experiments on several benchmark datasets and the results demonstrate the effectiveness of our proposed methods.

The rest of the paper is organized as follows. We introduce the details of our proposed CDML and S-CDML methods in Sect. 2. In Sect. 3, an optimization algorithm is proposed to solve our problems. Section 4 shows various

experimental results which demonstrate the effectiveness of our proposed methods. In Sect. 5, we will give a conclusion of our work.

2 Semi-supervised Coefficient-Based Distance Metric Learning

In this section, we first introduce the details of our proposed coefficient-based distance metric learning method. Then, we extend it into a semi-supervised framework. Before we introduce our proposed method, the general framework of semi-supervised distance metric learning is presented. The objective function of semi-supervised distance metric learning can be described as follows:

$$\begin{aligned} &\min_{A} g_l(A) + \beta g_u(A) + \lambda R(A) \\ &s.t.\ A \succeq 0, \end{aligned} \tag{1}$$

where $A \in \mathbb{S}_+^{d\times d}$ is a positive semi-defined metric in a $d \times d$ dimensional space. $g_l(A)$ is a loss function of labeled data, $g_u(A)$ is a loss function of unlabeled data and $R(A)$ is a regularization term of metric A. β and λ are two trade-off parameters. β is used to balance the influence of labeled data and unlabeled data. λ is used to control the complexity of the model.

Notice that a positive semi-defined metric A can be decomposed into a linear combination of a set of base vectors as follows:

$$A = \sum\nolimits_{i=1}^{n} c_i \mathbf{u}_i \mathbf{u}_i^T, \tag{2}$$

where $\mathbf{u}_i \in \mathbb{R}^{d\times 1}$ is the i-th random base vector and c_i is the i-th entry of coefficient vector $\mathbf{c} \in \mathbb{R}^n$. n is the total number of the base vectors. Consequently, the learning of metric A is equal to the learning of the coefficients of these base vectors. The objective formulation (1) can be reformulated by replacing metric A with formulation (2) as:

$$\begin{aligned} &\min_{\mathbf{c}} g_l(\mathbf{c}) + \beta g_u(\mathbf{c}) + \lambda R(\mathbf{c}), \\ &s.t.\ \sum\nolimits_{i=1}^{n} c_i = 1, \end{aligned} \tag{3}$$

In the following sections, we will give a detailed introduction to the construction of $g_l(\mathbf{c})$, $g_u(\mathbf{c})$ and $R(\mathbf{c})$.

2.1 Coefficient-Based Distance Metric Learning

In distance metric learning, pairwise constraint has been widely used [1,11,12]. We use pairwise constraint in our methods to pull the similar pairs close and

push dissimilar pairs apart. For simplicity and clear notations, two sets of pairs are introduced:

$$\begin{aligned} \mathcal{S} &= \{(x_i, x_j) | x_i \; and \; x_j \; are \; similar \; , y_{ij} = 1\}, \\ \mathcal{D} &= \{(x_i, x_j) | x_i \; and \; x_j \; are \; disimilar \; , y_{ij} = -1\}, \end{aligned} \tag{4}$$

where x_i and x_j are two samples in d dimensional feature space, y_{ij} denotes the similarity of the pair (x_i, x_j). The similar pair set $\mathcal{S}$ includes positive pairs which have the same class label. And the dissimilar pair set $\mathcal{D}$ includes negative pairs which have different labels. The distance between pair (x_i, x_j) under the distance metric A is denoted as $d_A(x_i, x_j)$ which can be formulated as:

$$d_A^2(x_i, x_j) = \|x_i - x_j\|_A^2 = (x_i - x_j)^{\mathrm{T}} A (x_i - x_j). \tag{5}$$

Similar to RDML [8], we adopt hinge loss in $g_l(\mathbf{c})$. Additionally, we introduce a L1-norm regularization term $R(\mathbf{c})$ to guarantee the sparsity of the coefficient vector. Consequently, the objective formulation of CDML can be expressed as follows:

$$\begin{aligned} &\min_{\mathbf{c}} \frac{1}{N} \sum\nolimits_{k=1}^{N} max(0, b - y_{ij}^k (1 - \left\|x_i^k - x_j^k\right\|)_A^2)) + \lambda \left\|\mathbf{c}\right\|_1, \\ &s.t. \; \sum\nolimits_{i=1}^{n} c_i = 1, \end{aligned} \tag{6}$$

where (x_i^k, x_j^k) is the k-th sample pair and y_{ij}^k is a pairwise label of (x_i^k, x_j^k). N represents the amount of labeled sample pairs in $\mathcal{S}$ and $\mathcal{D}$. For notation simplicity, we denote $y_{ij}^k = y_k$ and $\delta_k = x_i^k - x_j^k$. So $\left\|x_i^k - x_j^k\right\|_A^2 = \sum_{i=1}^{n} c_i \delta_k^T \mathbf{u}_i \mathbf{u}_i^T \delta_k = \mathbf{c}^T \mathbf{h}_k$ where $\mathbf{h}_k = [h_k^1, ..., h_k^n]^T$ with $h_k^i = \delta_k^T \mathbf{u}_i \mathbf{u}_i^T \delta_k$. $\mathbf{u}_i$ is the i-th random base vector. Consequently, problem (6) can be rewritten as follows:

$$\begin{aligned} &\min_{\mathbf{c}} \frac{1}{N} \sum\nolimits_{k=1}^{N} max(0, b - y_k (1 - \mathbf{c}^T \mathbf{h}_k)) + \lambda \left\|\mathbf{c}\right\|_1, \\ &s.t. \; \sum\nolimits_{i=1}^{n} c_i = 1. \end{aligned} \tag{7}$$

2.2 Semi-supervised Coefficient-Based Distance Metric Learning

Considering the situation we don't have enough labeled samples, we propose a novel method to construct positive pairs and negative pairs from the distribution of unlabeled data. The idea of our semi-supervised learning method can be illustrated in Fig. 1. For each unlabeled sample x_i, we use K-NN(K=1) with Euclidean distance to choose its nearest sample x_j. If x_i is also the nearest sample of x_j, we denote (x_i, x_j) as an positive pair and $y_{ij} = 1$. As for negative pairs, we first find the maximum distance between positive pairs from all classes and denote the maximum distance as $maxdist$. A threshold T is defined as: $T = maxdist + \varepsilon$, where ε is a margin to make our method more robust to the noise. For each unlabeled input x_i, we calculate the Euclidean distance between

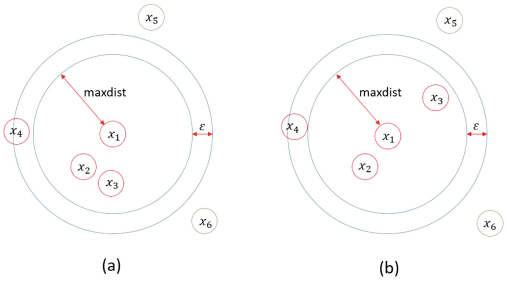

Fig. 1. Two examples of sample pairs in S-CDML. **(a)** For an unlabeled sample x_1, (x_1, x_5) and (x_1, x_6) are two negative pairs while we have no positive pair since the nearest neighbor of x_2 is not x_1. $d(x_1, x_4) \in (maxdist, maxdist + \varepsilon)$, so (x_1, x_4) is not a negative pair. **(b)** (x_1, x_2) is a positive pair; (x_1, x_5) and (x_1, x_6) are two negative pairs.

x_i and x_l which is one sample from the rest of the data. If the Euclidean distance between x_i and x_l is larger than T, (x_i, x_l) is a candidate of negative pairs and $y_{il} = -1$. For each unlabeled positive pair, we randomly select 10 unlabeled negative pairs.

The loss function of semi-supervised part of S-CDML can be formulated as follows:

$$g_u = \sum\nolimits_{k=1}^{N'} w_k max(0, b - y_k(1 - \mathbf{c}^T \mathbf{h}_k)), \tag{8}$$

where y_k and $\mathbf{h}_k$ have the same formulation as those in supervised part except that they utilize the pairs from unlabeled data. N' is the number of sample pairs from unlabeled data. w_k is the weight of the k-th pair. We set $w_k = \frac{1}{d(k)}$, where $d(k)$ is the Euclidean distance of the k-th pair. Then we normalize w_k to make the sum of $\{w_k\}_{k=1}^{N'}$ equals to 1: $\sum_k w_k = 1$. We give larger weights to those pairs which tend to be closer for the following reasons: if they are positive pairs, they are more reliable with smaller distance between them; if they are negative pairs, the closer they approach the threshold, the more attention they should attract.

We now give the objective function of our proposed semi-supervised coefficient-based distance metric learning(S-CDML) as follows:

$$\begin{aligned} &\min_{\mathbf{c}} \frac{1}{N} \sum\nolimits_{k=1}^{N} max(0, b - y_k(1 - \mathbf{c}^T \mathbf{h}_k)) \\ &+ \beta \sum\nolimits_{k=1}^{N'} w_k max(0, b - y_k(1 - \mathbf{c}^T \mathbf{h}_k)) + \lambda \left\| \mathbf{c} \right\|_1 \\ &s.t. \sum\nolimits_{i=1}^{n} c_i = 1 \end{aligned} \tag{9}$$

where we set $b = 0.5$, β and λ are trade-off parameters.

Algorithm 1. The Optimization Algorithm of Semi-supervised Coefficient-based Distance Metric Learning

Input: The lagrangian multipliers $\mathbf{u}_1 = \mathbf{u}_2 = \mathbf{u}_3 = \mathbf{1}$, $\rho_1 = \rho_2 = \rho_3$ can be choose from validation set

Output: $\mathbf{c}$

1. initial $\mathbf{a} = \mathbf{b} = \mathbf{c}$ as zero vectors
2. **while** (not converge) **do**:
3. $\quad \mathbf{a}^{t+1} = \min_{\mathbf{a}} L_\rho(\mathbf{a}^t, \mathbf{b}^t, \mathbf{c}^t, \mathbf{u}^t)$
4. $\quad \mathbf{b}^{t+1} = \min_{\mathbf{b}} L_\rho(\mathbf{a}^{t+1}, \mathbf{b}^t, \mathbf{c}^t, \mathbf{u}^t)$
5. $\quad \mathbf{c}^{t+1} = \min_{\mathbf{c}} L_\rho(\mathbf{a}^{t+1}, \mathbf{b}^{t+1}, \mathbf{c}^t, \mathbf{u}^t)$
6. $\quad \mathbf{u}^{t+1} = \min_{\mathbf{u}} L_\rho(\mathbf{a}^{t+1}, \mathbf{b}^{t+1}, \mathbf{c}^{t+1}, \mathbf{u}^t)$
7. **end while**

3 An Optimization Algorithm

To solve problem (9), we can integrate the losses of labeled data and unlabeled data into one formulation, which is shown as the following:

$$g(\mathbf{c}) = \sum\nolimits_{k=1}^{N+N'} \omega_k max(0, 0.5 - y_k(1 - \mathbf{c}^T \mathbf{h}_k)), \tag{10}$$

where $\omega_k = \frac{1}{N}, k = 1, 2, ..., N$ which means that we give equal weights to all pairs from labeled data. And $\omega_k = \frac{\beta}{d(k)}, k = N+1, N+2, ..., N+N'$ which has been introduced in the above section. Consequently, we should optimize a loss function with the following formulation:

$$\begin{aligned} &\min_{\mathbf{c}} g(\mathbf{c}) + \lambda \left\| \mathbf{c} \right\|_1, \\ &s.t. \sum\nolimits_{i=1}^{n} c_i = 1. \end{aligned} \tag{11}$$

The loss term and the regularization term of the above formulation (11) are both non-smoothed, it is difficult to solve this problem with gradient descent method. Some works solve this problem by replacing the non-smoothed function with a smooth approximation [11,14]. This method can solve the problem but will lose some accuracy according to the performance of approximation. To address this drawback, we propose an optimization algorithm to solve this problem with better performance. We introduce some additional variables and define an equivalence problem, which can be solved using alternating direction method of multipliers (ADMM) [3]. The detailed optimization algorithm is shown in Algorithm 1. The non-smoothed loss function and regularization term need not to be approximated in our optimization algorithm, therefore, we can obtain a better solution. Additionally, each step can be solved efficiently. Consequently, the objective problem can be solved efficiently with better performance using our proposed algorithm.

We first introduce two additional variables $\mathbf{a}, \mathbf{b}$ and define an equivalence problem of the original problem (11) as follows:

$$\begin{aligned} &\min_{\mathbf{a},\mathbf{b},\mathbf{c}} g(\mathbf{a}) + \lambda \left\|\mathbf{b}\right\|_1, \\ &s.t.\ \mathbf{a} = \mathbf{c}, \mathbf{b} = \mathbf{c}, \sum\nolimits_{i=1}^{n} c_i = 1. \end{aligned} \tag{12}$$

Then we use augmented lagrangian method to express this problem as the following:

$$\begin{aligned} &\min_{\mathbf{a},\mathbf{b},\mathbf{c},\mathbf{u}} L_\rho(\mathbf{a},\mathbf{b},\mathbf{c},\mathbf{u}) = g(\mathbf{a}) + \lambda \left\|\mathbf{b}\right\|_1 + \mathbf{u}_1(\mathbf{1}^T\mathbf{c} - 1) \\ &+ \frac{\rho_1}{2}(\mathbf{1}^T\mathbf{c} - 1)^2 + \mathbf{u}_2(\mathbf{a} - \mathbf{c}) + \frac{\rho_2}{2}\left\|\mathbf{a} - \mathbf{c}\right\|_2^2 \\ &+ \mathbf{u}_3(\mathbf{b} - \mathbf{c}) + \frac{\rho_3}{2}\left\|\mathbf{b} - \mathbf{c}\right\|_2^2, \end{aligned} \tag{13}$$

where $L_\rho(\mathbf{a},\mathbf{b},\mathbf{c},\mathbf{u})$ is an augmented lagrangian function and $\mathbf{1}^T$ is a vector whose entries equal to one. The problem can be solved in four steps:

I. Fix other variables, update a:

We need to solve the following problem:

$$\begin{aligned} \mathbf{a}^{t+1} &= \min_{\mathbf{a}} g(\mathbf{a}) + \mathbf{u}_2^t(\mathbf{a} - \mathbf{c}^t) + \frac{\rho_2}{2}\left\|\mathbf{a} - \mathbf{c}^t\right\|_2^2 \\ &= \min_{\mathbf{a}} g(\mathbf{a}) + \frac{\rho_2}{2}\left\|\mathbf{a} - (\mathbf{c}^t - \frac{\mathbf{u}_2^t}{\rho_2})\right\|_2^2. \end{aligned} \tag{14}$$

Then we compute the gradient of the object function with respect to $\mathbf{a}$ and set the gradient to zero. Variable $\mathbf{a}$ can be updated as the following:

$$\mathbf{a}^{t+1} = \mathbf{c}^t - \frac{\mathbf{u}_2^t}{\rho_2} - \frac{1}{\rho_2} \cdot \frac{\partial g(\mathbf{a})}{\partial \mathbf{a}}. \tag{15}$$

The gradient of $g(\mathbf{a})$ can be calculated with the chain rule:

$$\frac{\partial g(\mathbf{a})}{\partial \mathbf{a}} = \sum_k \omega_k \cdot \frac{\partial f_k}{\partial \varphi}\frac{\partial \varphi}{\partial \mathbf{a}}, \quad \frac{\partial f_k}{\partial \varphi} \cdot \frac{\partial \varphi}{\partial \mathbf{a}} = \begin{cases} -y_k\mathbf{h}_k, & \varphi_k < 0.5 \\ 0, & \varphi_k \geq 0.5 \end{cases}, \tag{16}$$

where f_k =max(0,0.5-φ), and $\varphi = y_k(1 - \mathbf{a}^T\mathbf{h}_k)$.

II. Fix other variables, update b:

The optimization goal becomes:

$$\begin{aligned} \mathbf{b}^{t+1} &= \min_{\mathbf{b}} \lambda \left\|\mathbf{b}\right\|_1 + \mathbf{u}_3^t(\mathbf{b} - \mathbf{c}^t) + \frac{\rho_3}{2}\left\|\mathbf{b} - \mathbf{c}^t\right\|_2^2 \\ &= \min_{\mathbf{b}} \lambda \left\|\mathbf{b}\right\|_1 + \frac{\rho_3}{2}\left\|\mathbf{b} - (\mathbf{c}^t - \frac{\mathbf{u}_3^t}{\rho_3})\right\|_2^2. \end{aligned} \tag{17}$$

Similar to the process of updating $\mathbf{a}$, we get the update equation of $\mathbf{b}$ as follows:

$$b_i^{t+1} = \begin{cases} z_i + \frac{\lambda}{\rho_3}, & z_i < -\frac{\lambda}{\rho_3} \\ 0, & else \\ z_i - \frac{\lambda}{\rho_3}, & z_i > \frac{\lambda}{\rho_3} \end{cases}, \tag{18}$$

where $\mathbf{z} = \mathbf{c}^t - \frac{\mathbf{u}_3^t}{\rho_3}$, z_i is the i-th entry in $\mathbf{z}$, b_i is the i-th entry in $\mathbf{b}$.

III. Fix other variables, update c:

We should optimize the following problem:

$$\begin{aligned} \mathbf{c}^{t+1} = & \min_{\mathbf{c}} \mathbf{u}_1^t(\mathbf{1}^T\mathbf{c} - 1) + \frac{\rho_1}{2}(\mathbf{1}^T\mathbf{c} - 1)^2 \\ & + \mathbf{u}_2^t(\mathbf{a}^{t+1} - \mathbf{c}) + \frac{\rho_2}{2}\left\|\mathbf{a}^{t+1} - \mathbf{c}\right\|_2^2 \\ & + \mathbf{u}_3^t(\mathbf{b}^{t+1} - \mathbf{c}) + \frac{\rho_3}{2}\left\|\mathbf{b}^{t+1} - \mathbf{c}\right\|_2^2. \end{aligned} \tag{19}$$

Then we compute the gradient of the object function with respect to $\mathbf{c}$, and we get:

$$\frac{\partial L}{\partial c_i} = u_{1_i}^t + \rho_1(\mathbf{1}^T\mathbf{c} - 1) - u_{2_i}^t - \rho_2(a_i^{t+1} - c_i) - u_{3_i}^t - \rho_3(b_i^{t+1} - c_i), \tag{20}$$

where $u_{1_i}^t$ is the i-th entry in u_1^t. By solving a set of linear equations $\frac{\partial L}{\partial c_i} = 0$, we can get the update equation of $\mathbf{c}$.

IV. Fix other variables, update lagrangian multipliers:

We update the lagrangian multipliers $\mathbf{u}_1, \mathbf{u}_2$ and $\mathbf{u}_3$ using the following equations:

$$\begin{aligned} \mathbf{u}_1^{t+1} &= \mathbf{u}_1^t + \rho_1(\mathbf{1}^T\mathbf{c}^{t+1} - 1), \\ \mathbf{u}_2^{t+1} &= \mathbf{u}_2^t + \rho_2(\mathbf{a}^{t+1} - \mathbf{c}^{t+1}), \\ \mathbf{u}_3^{t+1} &= \mathbf{u}_3^t + \rho_2(\mathbf{b}^{t+1} - \mathbf{c}^{t+1}). \end{aligned} \tag{21}$$

4 Experiment

In this section, we conduct experiments on several landmark datasets from UCI repository. They are Wine dataset, Balance-scale dataset, Breast-cancer dataset and Glass dataset. These datasets have been widely used for evaluating the performance of distance metric learning in previous works [4,8,12]. We compared our proposed methods coefficient-based distance metric learning (CDML) and semi-supervised coefficient-based distance metric learning (S-CDML) with six supervised distance metric learning methods and two semi-supervised distance metric learning methods. The six supervised learning methods are: (1) Regular euclidean distance metric (Euclidean); (2) Relevant component analysis (RCA) [2]; (3) Information-theoretic metric learning (ITML) [4]; (4) Regularized distance metric learning (RDML) [8]; (5) Distance metric learning of large margin

nearest neighbor (LMNN) [12]; (6) Regularized large margin distance metric learning (RLMM) [10]. And the two semi-supervised learning methods are: (1) A semi-supervised distance metric learning (SSmetric) [6]; (2) Semi-supervised regularized large margin distance metric learning (S-RLMM) [10].

For all datasets, we randomly select 10% of the data as the training set and the rest is split into two halves as validation set and test set correspondingly. To avoid randomness, we repeated the random splits for five times and report the average performance. For all methods, we use the same data with the same normalization for a fair comparison. In our methods, the base vectors is generated subject to $\mathcal{N}(0, 1)$ Gaussian distribution. All parameters are chosen on validation set.

4.1 Comparison of Optimization Algorithms

In this section, we compare our proposed optimization algorithm with the optimization algorithm of DTDML. The results are shown in Table 1. From the results, we can conclude that our proposed optimization algorithm outperforms the one of DTDML on all the datasets. This demonstrates the effectiveness of our optimization algorithm.

Table 1. Comparison between our optimization algorithm and the optimization algorithm of DTDML. We evaluate the performance using classification accuracy. Mean accuracy and the standard deviation are reported.

Dataset	Cancer	Scale	Wine	Glass
DTDML	95.15 ± 0.12	82.28 ± 0.41	89.71 ± 0.51	61.24 ± 0.34
CDML	$\mathbf{95.82} \pm 0.54$	$\mathbf{86.24} \pm 0.58$	$\mathbf{91.44} \pm 0.65$	$\mathbf{61.86} \pm 0.17$

4.2 Performance Comparison Between Different Methods

In this section, we compare the performance of our proposed metric learning methods with that of other state-of-the-art methods. Table 2 summarises the performance comparison between our proposed supervised distance metric learning CDML and other six supervised methods. From the results, we can conclude that our supervised method CDML outperforms other supervised methods on all datasets except Wine dataset. Table 3 shows the comparison between S-CDML and other two semi-supervised methods on different datasets. Our semi-supervised method S-CDML outperforms other methods on all datasets except Wine dataset. RLMM and S-RLMM achieve the best performance on wine dataset in Tables 2 and 3 correspondingly. This is mainly because that RLMM and S-RLMM utilize both pairwise constraints and triplet constraints which can provide more information about the distribution of the data. However, our methods only utlize pairwise constraints. Considering the overall performance, our proposed CDML and S-CDML are demonstrated to be effective.

Table 2. Comparison between CDML and other six supervised methods. Mean classification accuracy (%) and the standard deviation are reported.

Dataset	Euclidean	RCA	ITML	RDML	LMNN	RLMM	CDML
Cancer	94.55 ± 0.00	94.29 ± 0.00	94.87 ± 0.18	95.39 ± 0.00	95.23 ± 0.17	95.00 ± 0.00	**95.82** ± 0.54
Scale	76.68 ± 0.00	80.42 ± 0.00	82.26 ± 1.93	78.94 ± 0.00	83.49 ± 0.44	83.68 ± 0.00	**86.24** ± 0.58
Wine	88.64 ± 0.00	64.44 ± 0.00	91.36 ± 0.97	90.12 ± 0.00	91.54 ± 0.56	**91.60** ± 0.00	91.44 ± 0.65
Glass	50.00 ± 0.00	50.00 ± 0.00	57.80 ± 0.84	59.80 ± 0.00	54.80 ± 1.09	60.21 ± 0.00	**61.86** ± 0.17

Table 3. Comparison between S-CDML and other two semi-supervised methods. Mean classification accuracy (%) and the standard deviation are reported.

Dataset	SSmetric	S-RLMM	S-CDML
Cancer	95.32 ± 0.00	95.45 ± 0.13	**96.25** ± 0.26
Scale	82.69 ± 0.00	83.26 ± 0.00	**86.36** ± 0.55
Wine	92.35 ± 0.00	**94.56** ± 0.35	92.84 ± 0.49
Glass	57.80 ± 0.00	61.03 ± 0.00	**63.16** ± 0.68

5 Conclusion

In this paper, we propose a novel distance metric learning method by learning a linear combination of random base vectors to construct the metric. In this way, we can easily control the complexity of our method by adjusting the number of random base vectors. We further extend our proposed distance metric learning method into a semi-supervised learning framework by introducing effective unlabeled pairwise constraints. Additionally, we propose an optimization algorithm to solve this non-smoothed problem efficiently. Many experiments have been conducted on several landmark datasets and the results demonstrate the effectiveness of our proposed methods.

Acknowledgments. This work is supported by the 973 project 2015CB351803, NSFC No. 61572451 and No. 61390514, Youth Innovation Promotion Association CAS CX-2100060016, and Fok Ying Tung Education Foundation WF2100060004.

References

1. Baghshah, M.S., Shouraki, S.B.: Semi-supervised metric learning using pairwise constraints. In: Twenty-First International Joint Conference on Artificial Intelligence (2009)
2. Bar-Hillel, A., Hertz, T., Shental, N., Weinshall, D.: Learning a mahalanobis metric from equivalence constraints. J. Mach. Learn. Res. **6**, 937–965 (2005)
3. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trends® Mach. Learn. **3**(1), 1–122 (2011)
4. Davis, J.V., Kulis, B., Jain, P., Sra, S., Dhillon, I.S.: Information-theoretic metric learning. In: Proceedings of the 24th international conference on Machine learning, pp. 209–216. ACM (2007)

5. Fisher, R.: The use of multiple measures in taxonomic problems. Ann. Eugenics **7**, 179–188 (1936)
6. Hoi, S.C., Liu, W., Chang, S.F.: Semi-supervised distance metric learning for collaborative image retrieval and clustering. ACM Trans. Multimedia Comput. Commun. Appl. (TOMM) **6**(3), 18 (2010)
7. Hoi, S.C., Liu, W., Lyu, M.R., Ma, W.Y.: 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 (2006)
8. Jin, R., Wang, S., Zhou, Y.: Regularized distance metric learning: theory and algorithm. In: Advances in neural information processing systems, pp. 862–870 (2009)
9. Jolliffe, I.: Principal Component Analysis. Wiley Online Library (2002)
10. Li, Y., Tian, X., Tao, D.: Regularized large margin distance metric learning. In: 2016 IEEE 16th International Conference on Data Mining (ICDM), pp. 1015–1022. IEEE (2016)
11. Luo, Y., Liu, T., Tao, D., Xu, C.: Decomposition-based transfer distance metric learning for image classification. IEEE Trans. Image Process. **23**(9), 3789–3801 (2014)
12. Weinberger, K.Q., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. J. Mach. Learn. Res. **10**, 207–244 (2009)
13. Yu, J., Wang, M., Tao, D.: Semisupervised multiview distance metric learning for cartoon synthesis. IEEE Trans. Image Process. **21**(11), 4636–4648 (2012)
14. Zhou, T., Tao, D., Wu, X.: Nesvm: a fast gradient method for support vector machines. In: IEEE 10th International Conference on Data Mining (ICDM), 2010, pp. 679–688. IEEE (2010)
15. Zhu, X., Goldberg, A.B.: Introduction to semi-supervised learning. Synth. Lect. Artif. Intell. Mach. Learn. **3**(1), 1–130 (2009)