Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 352849, 12 pages
http://dx.doi.org/10.1155/2015/352849

Research Article

Chi-Squared Distance Metric Learning for Histogram Data

Wei Yang,[1] Luhui Xu,[2] Xiaopan Chen,[1] Fengbin Zheng,[1] and Yang Liu[1]

[1]*Laboratory of Spatial Information Processing, School of Computer and Information Engineering, Henan University, Kaifeng 475004, China*
[2]*Department of Information Engineering, Shengda Trade Economics and Management College of Zhengzhou, Zhengzhou 451191, China*

Correspondence should be addressed to Yang Liu; ly.sci.art@gmail.com

Received 11 December 2014; Revised 25 March 2015; Accepted 27 March 2015

Academic Editor: Davide Spinello

Learning a proper distance metric for histogram data plays a crucial role in many computer vision tasks. The chi-squared distance is a nonlinear metric and is widely used to compare histograms. In this paper, we show how to learn a general form of chi-squared distance based on the nearest neighbor model. In our method, the margin of sample is first defined with respect to the nearest hits (nearest neighbors from the same class) and the nearest misses (nearest neighbors from the different classes), and then the simplex-preserving linear transformation is trained by maximizing the margin while minimizing the distance between each sample and its nearest hits. With the iterative projected gradient method for optimization, we naturally introduce the $\ell_{2,1}$ norm regularization into the proposed method for sparse metric learning. Comparative studies with the state-of-the-art approaches on five real-world datasets verify the effectiveness of the proposed method.

1. Introduction

Histograms are frequently used tools in natural language processing and various computer vision tasks, including image retrieval, image classification, shape matching, and object recognition, to represent texture and color features or to characterize rich information in local/global regions of objects. In particular, a histogram in the statistics is the frequency distribution of a set of specific measurements over discrete intervals. For many computer vision tasks, each object of interest can be presented as a histogram by using visual descriptors, such as SIFT [1], SURF [2], GIST [3], and HOG [4]. As a result, the resulting histogram obtains some merits of the descriptors, for example, rotation-invariant, scale-invariant, and translation-invariant. These make it an excellent representation method for performing classification and recognition of objects.

When the histogram representations are adopted, the choice of histogram distance metric has a great impact on the classification performance or recognition accuracy of the specific task. Since a histogram can be considered as a vector of probability, many metrics such as ℓ_2 distance, chi-squared distance, and Kullback-Leibler (KL) divergence can be used directly. These metrics, however, only account for the difference between the corresponding bins and are hence sensitive to distortions in visual descriptors as well as quantization effects [5]. To mitigate these problems, many cross-bin distances have been proposed. Rubner et al. [6] propose the Earth Movers Distance (EMD), which is defined as the minimal cost that must be paid to transform one histogram into the other, by considering the cross-bin information. Diffusion distance [5] exploits the idea of diffusion process to define the difference between two histograms as a temperature field. The Quadratic-Chi distances (QCS and QCN) [7] take into account cross-bin relationships and meanwhile reduce the effect of large bins. In particular, for the cross-bin distance, most of the work mainly focuses on how to improve the EMD and hence many variants have been proposed. EMD-ℓ_1 [8] uses the ℓ_1 distance as the ground distance and significantly simplifies the original linear programming formulation of the EMD. Pele and Werman [9] propose a different formulation of the EMD with a linear-time algorithm for nonnormalized histograms. FastEMD [10] adopts a robust thresholded ground distance

and was shown to outperform the EMD in both accuracy and speed. TEMD [11] uses a tangent vector to represent each global transformation. For the methods mentioned above, the determinations of metrics are all based on a priori knowledge of features or handcraft. However, distance metric is problem-specific and designing a good distance metric manually is extremely difficult. Aiming at this problem, some researchers have attempted to learn a proper distance metric from histogram training data. Considering that the ground distance, which is the unique variable of the EMD, should be chosen according to the problem at hand, Cuturi and Avis [12] propose a ground metric learning algorithm to learn the ground metric adaptively by using the training data. Subsequently, EMDL [13] formulates the ground metric learning as an optimization problem in which a ground distance matrix and a flow-network for the EMD are learned jointly based on a partial ordering of histogram distances. Noh [14] uses a convex optimization method to perform chi-squared metric learning with relaxation. χ^2-LMNN [15] employs a large-margin framework to learn a generalized chi-squared distance for histogram data and obtains a significant improvement compared to standard histogram metrics and the state-of-the-art metric learning algorithms. Le and Cuturi [16] adopt the generalized Aitchison embedding to compare histograms by mapping the probability simplex onto a suitable Euclidean space.

In this paper, we present a novel nearest neighbor-based nonlinear metric learning method, chi-squared distance metric learning (CDML), for normalized histogram data. CDML learns a simplex-preserving linear transformation by maximizing the margin while minimizing the distance between each sample and its k-nearest hits. In the original space, the learned metric can be considered as a cross-bin metric. For sparse metric learning, the $\ell_{2,1}$ norm regularization term is further introduced to enforce row sparsity on the learned linear transformation matrix. Two solving strategies, the iterative projected gradient and the soft-max method, are used to induce the linear transformation. We demonstrate that our algorithms perform better than the state-of-the-art ones in terms of classification performance.

The remainder of this paper is organized as follows. Section 2 provides a review of supervised metric learning algorithms. Section 3 describes the proposed distance metric learning method. The experimental results on five real-world datasets are given in Section 4. Meanwhile, we discuss the difference between our method and χ^2-LMNN in detail. Section 5 concludes the paper.

2. Related Work

In this section, we review the related work on supervised distance metric learning. Due to the seminal work of Xing et al. [17], which formulates metric learning as an optimization problem, supervised metric learning has been extensively studied in machine learning area and various algorithms have been proposed. In general, the proposed methods can be roughly cast into three different categories: Mahalanobis metric learning, local metric learning, and nonlinear metric learning. For the Mahalanobis metric learning, its main characteristic is to learn a linear transformation or a positive semidefinite matrix from training data under the Mahalanobis distance metric. The representative methods include neighborhood component analysis [18], large-margin nearest neighbor [19], and information-theoretic metric learning [20]. Neighborhood component analysis [18] learns a linear transformation by directly maximizing the stochastic variant of the expected leave-one-out classification accuracy on the training set. Large-margin nearest neighbor (LMNN) [19] formulates distance metric learning into a semidefinite programming problem by forcing that the k-nearest neighbors of each training sample belong to the same class while examples from different classes are separated by a large margin. Information-theoretic metric learning (ITML) [20] formulates distance metric learning as a particular Bregman optimization problem by minimizing the differential relative entropy between two multivariate Gaussians under constraints on the distance function. Bian and Tao [21] formulate metric learning as a constrained empirical risk minimization problem. Wang et al. [22] propose a general kernel classification framework, which can unify many representative and state-of-the-art Mahalanobis metric learning algorithms such as LMNN and ITML. Chang [23] uses boosting algorithm to learn a Mahalanobis distance metric. Shen et al. [24] propose an efficient and scalable approach to the Mahalanobis metric learning problem based on the Lagrange dual formulation. Yang et al. [25] propose a novel multitask framework for metric learning by using common subspace. For the local metric learning, its motivation is to increase the expressiveness of learned metrics so that more complex problems, such as heterogeneous data, can be better handled. In virtue of involving more learning parameters compared to its global counterpart, local metric learning is prone to overfitting. One of early local metric algorithms is discriminant adaptive nearest neighbor classification (DANN) [26], which estimates local metrics by shrinking neighborhoods in directions orthogonal to the local decision boundaries and enlarging the neighborhoods parallel to the boundaries. Multiple metrics LMNN [19] learns multiple locally linear transformations in different parts of the sample space under the large-margin framework. By using an approximation error bound of the metric matrix function, Wang et al. [27] formulate local metric learning as linear combinations of basis metrics defined on anchor points over different regions of the instance space. Mu et al. [28] propose a new local discriminative distance metrics algorithm to learn multiple distance metrics. For nonlinear metric learning, there are two ways to conduct metric learning. One strategy is to use kernel trick to learn a linear metric in the high-dimensional nonlinear feature space induced by a kernel function. The kernelized variants of many Mahalanobis metric learning methods, such as KLFDA [29] and large-margin component analysis [30], have been shown to be efficient in capturing complicated nonlinear relationships between data. Soleymani Baghshah and Bagheri Shouraki [31] formulate nonlinear metric learning as constrained trace ratio problems by using both positive and negative constraints. By combining metric learning and multiple kernel learning, Wang et al. [32]

propose a general framework for learning a linear combination of a number of predefined kernels. Another strategy is to learn nonlinear forms of metrics directly. Based on convolutional neural network, Chopra et al. [33] propose learning a nonlinear function such that the ℓ_1 norm in the target space approximates the semantic distance in the input space. GB-LMNN [15] learns a nonlinear mapping directly in function space with gradient boosted regression trees. Support vector metric learning [34] learns a metric for radial basis function kernel by minimizing the validation error of the SVM prediction at the same time as it trains the SVM classifier. For a comprehensive review of metric learning and its applications we refer the readers to [35–37] for details.

Although metric learning about Mahalanobis distance has been widely studied, metric learning for chi-squared distance is largely unexplored. Unlike Mahalanobis distance, chi-squared distance is a nonlinear metric and its general form requires the learned linear transformation to be simplex-preserving. Therefore, the existing linear metric learning algorithms cannot naturally apply to chi-squared distance. χ^2-LMNN adopts the LMNN model to learn chi-squared distance, but its additional margin hyperparameter is sensitive to the used data and needs to be evaluated on a hold-out set. In addition, it exploits the soft-max method to optimize the objective function, which makes the regularizers unable to be introduced naturally. The proposed method utilizes the margin of sample to construct the objective function and adopts the iterative projected gradient method for optimization and hence overcomes the weaknesses of the χ^2-LMNN. The regularizers can be incorporated into our model naturally and no additional parameter needs to be evaluated compared to the χ^2-LMNN.

3. Chi-Squared Distance Metric Learning

In this section, we will propose a metric learning algorithm termed as chi-squared distance metric learning (CDML). This algorithm uses the margin of sample to construct the objective function. It is more suitable to metric learning for histogram data.

In the following, we will first introduce the definition of the margin of sample. Then the motivation and the objective function of CDML will be proposed. Finally, the optimization method of the algorithm will be discussed.

3.1. The Margin of Sample. Let training data be $\{\mathbf{x}_i, y_i\}_{i=1}^N$, where $\mathbf{x}_i$ is sampled from a probability simplex $S^d = \{\mathbf{x} \in \Re^d \mid \mathbf{x} \succeq 0, \mathbf{1}^T\mathbf{x} = 1\}$ and let $y_i \in \{1, 2, \ldots, c\}$ be the associated class label; the symbol $\mathbf{1}$ denotes a d-dimensional column vector whose all components are one. The chi-squared distance between two samples $\mathbf{x}_i$ and $\mathbf{x}_j$ can be computed by

$$\chi^2\left(\mathbf{x}_i, \mathbf{x}_j\right) = \frac{1}{2}\sum_{l=1}^{d}\frac{\left(x_{il} - x_{jl}\right)^2}{x_{il} + x_{jl}}, \tag{1}$$

where x_{il} indicates the lth feature of the sample $\mathbf{x}_i$.

For each instance in the original input space, we can map it into an r-dimensional probability simplex space by performing a simplex-preserving linear transformation $\mathbf{x}' = \mathbf{L}\mathbf{x}$, where $\mathbf{L}$ is an element-wise nonnegative matrix of size $r \times d$ ($r \leq d$) and the sum of each column element is one. In particular, the set of such simplex-preserving linear transformations can be defined as $\Theta = \{\mathbf{L} \in \Re^{r\times d} : \forall i, \forall j, L_{ij} \geq 0 \text{ and } \forall j, \sum_i L_{ij} = 1\}$. With the linear transform matrix $\mathbf{L}$, the chi-squared distance between two instances $\mathbf{x}_i$ and $\mathbf{x}_j$ under the transformed space can be written as

$$\chi^2_{\mathbf{L}}\left(\mathbf{x}_i, \mathbf{x}_j\right) = \chi^2\left(\mathbf{L}\mathbf{x}_i, \mathbf{L}\mathbf{x}_j\right). \tag{2}$$

For each sample $\mathbf{x}_i$, we call $\mathbf{x}_j$ a hit if $\mathbf{x}_j$ ($j \neq i$) has the same class label with $\mathbf{x}_i$, and the nearest hit $\mathbf{x}_j$ ($j \neq i$) is defined as the hit which has the minimum distance with the sample $\mathbf{x}_i$. Similarly, we call $\mathbf{x}_j$ a miss if the class label of $\mathbf{x}_j$ is different from $\mathbf{x}_i$, and the nearest miss $\mathbf{x}_j$ is defined as the miss which has the minimum distance with the sample $\mathbf{x}_i$. Let $\mathbf{NH}_l(\mathbf{x}_i)$ and $\mathbf{NM}_l(\mathbf{x}_i)$ be the lth nearest hit and miss of $\mathbf{x}_i$, respectively. The margin of sample [38] $\mathbf{x}_i$ with respect to its jth nearest hit and lth nearest miss is defined as

$$\rho_{ijl} = \chi^2_{\mathbf{L}}\left(\mathbf{x}_i, \mathbf{NM}_l\left(\mathbf{x}_i\right)\right) - \chi^2_{\mathbf{L}}\left(\mathbf{x}_i, \mathbf{NH}_j\left(\mathbf{x}_i\right)\right), \tag{3}$$

where $1 \leq j, l \leq k$. Note that $\mathbf{NH}_j(\mathbf{x}_i)$ and $\mathbf{NM}_l(\mathbf{x}_i)$ are determined by the generalized chi-squared distance and the transformation matrix $\mathbf{L}$ affects the margin through the distance metric.

3.2. The Objective Function. Similar to many metric learning algorithms about Mahalanobis distance, the goal of our algorithm is to learn a simplex-preserving linear transformation optimizing kNN classification. Given an unclassified sample point $\mathbf{x}$, kNN first finds its k-nearest neighbors in the training set and then assigns the label by the class that appears most frequently in the k-nearest neighbors. Therefore, for robust kNN classification, each training sample $\mathbf{x}_i$ should have the same label with its k-nearest neighbors. Obviously, if the margins of all the samples in the training set are bigger than zero, then the robust kNN classification can be obtained. By maximizing the margins of all training samples, our distance metric learning problem can be formulated as follows:

$$\min_{\mathbf{L}\in\Theta}\sum_{i=1}^{N}\sum_{j,l}\frac{1}{\beta}\log\left(1 + \exp\left(-\beta\rho_{ijl}\right)\right). \tag{4}$$

Here, the utility function $u(\rho) = \log(1 + \exp(-\beta\rho))/\beta$ is used to control the contribution of each margin term to the objective function. The introduction of constraint $\mathbf{L} \in \Theta$ is to ensure that the chi-squared distance in the transformed space is still a well-defined metric.

Note that in (4) maximizing the margins can also be attained by increasing the distances between each sample and its nearest hits and the distances to its nearest misses simultaneously, where the latter obtain the much larger increase. However, we expect that each training sample and its nearest hits form a compact clustering. Therefore, we further introduce a term to constrain the distances between

each sample and its nearest hits and obtain the following optimization problem:

$$\min_{\mathbf{L}\in\Theta} g(\mathbf{L}) = (1-\mu)\sum_{i=1}^{N}\sum_{j=1}\chi^2_{\mathbf{L}}\left(\mathbf{x}_i, \mathbf{NH}_j(\mathbf{x}_i)\right) + \mu\sum_{i=1}^{N}\sum_{jl}\frac{1}{\beta}\log\left(1+\exp\left(-\beta\rho_{ijl}\right)\right), \tag{5}$$

where $\mu \in [0,1]$ is a balance parameter trading off the effect between two terms.

Moreover, considering the sparseness of some high-dimensional histogram data, the direct transformation matrix learning probably overfits the training data, resulting in poor generalization performance. To address this problem, we introduce the $\ell_{2,1}$ norm regularizer to regularize the model complexity. With the $\ell_{2,1}$ norm regularization, the metric learning problem can be written as

$$\min_{\mathbf{L}\in\Theta} f(\mathbf{L}) = \left\{g(\mathbf{L}) + \lambda\|\mathbf{L}\|_{2,1}\right\}, \tag{6}$$

where the regularization term $\|\mathbf{L}\|_{2,1} = \sum_{i=1}^{r}\sqrt{\sum_{j=1}^{d}\mathbf{L}_{ij}^2}$ guarantees that the parameter matrix $\mathbf{L}$ is sparse in rows and λ is a nonnegative regularization parameter.

3.3. The Optimization Method. For the constrained optimization problem in (5), there are two methods that can be used to solve it. The first strategy is the iterative projected gradient method, which uses a gradient descent step to minimize $g(\mathbf{L})$ followed by the method of iterative projections to ensure that $\mathbf{L}$ is a simplex-preserving linear transformation matrix. Specifically, we will take a gradient step $\mathbf{L} = \mathbf{L} - \alpha\nabla g(\mathbf{L})$ and then project $\mathbf{L}$ into the set Θ on each iteration, where $\alpha > 0$ is a learning rate and $\nabla g(\mathbf{L})$ is the gradient of the objective function $g(\mathbf{L})$ about the matrix parameter $\mathbf{L}$. Note that the constraints on $\mathbf{L}$ can be seen as d separated probabilistic simplex constraints on each column of $\mathbf{L}$. Therefore, the projection onto the set Θ can be done by performing a probabilistic simplex projection, which can be efficiently implemented with a complexity of $\mathcal{O}(r\log(r))$ [39], on each column of $\mathbf{L}$. In addition, in order to compute the gradient $\nabla g(\mathbf{L})$, we need to obtain the partial derivative of the chi-squared distance in (2). Let $\overline{\mathbf{x}}_i = \mathbf{L}\mathbf{x}_i = (\overline{x}_{i1}, \ldots, \overline{x}_{ir})$ and $t_{ijp} = (\overline{x}_{ip} - \overline{x}_{jp})/(\overline{x}_{ip} + \overline{x}_{jp})$; the partial derivative of $\chi^2_{\mathbf{L}}(\mathbf{x}_i, \mathbf{x}_j)$ with respect to the matrix $\mathbf{L}$ can be given by

$$\frac{\partial\chi^2_{\mathbf{L}}\left(\mathbf{x}_i, \mathbf{x}_j\right)}{\partial L_{pq}} = t_{ijp}\left(x_{iq} - x_{jq}\right) - \frac{1}{2}t^2_{ijp}\left(x_{iq} + x_{jq}\right). \tag{7}$$

Generally speaking, the iterative projected gradient method needs a matrix of size $r \times d$ to initialize the linear transformation matrix $\mathbf{L}$. In our work, the rectangle identity matrix is always used to initialize it. When the iterative projected gradient method is used, in particular, various regularizers, such as Frobenius norm regularization and $\ell_{2,1}$ norm regularization, can be naturally incorporated into the objective function in (5) and without influencing the solving of the problem.

Another strategy is that we first transform the constrained optimization problem in (5) into an unconstrained version by introducing a soft-max function, and then the steepest gradient descent method is used for learning. Here the soft-max function is defined as

$$L_{ij} = \frac{e^{A_{ij}}}{\sum_{l=1}^{r} e^{A_{lj}}} \quad \forall i, j, \tag{8}$$

where the matrix $\mathbf{A}$ is an assistant parameter. Obviously, the matrix $\mathbf{L}$ is always in the set Θ for any choice of $\mathbf{A} \in \Re^{r\times d}$. Thus, we can use the gradient of the objective function $g(\mathbf{L})$ with respect to the matrix $\mathbf{A}$ to minimize (5). In particular, the partial derivative of the chi-squared distance in (2) with respect to the matrix $\mathbf{A}$ can be computed by

$$\frac{\partial\chi^2_{\mathbf{L}}\left(\mathbf{x}_i, \mathbf{x}_j\right)}{\partial A_{pq}} = L_{pq}\left(\left(t_{ijp}\left(x_{iq} - x_{jq}\right) - \frac{t^2_{ijp}\left(x_{iq} + x_{jq}\right)}{2}\right) - \sum_{l=1}^{r} L_{lq}\left(t_{ijl}\left(x_{iq} - x_{jq}\right) - \frac{t^2_{ijl}\left(x_{iq} + x_{jq}\right)}{2}\right)\right) \tag{9}$$

which will be used to compute the gradient $\nabla_A g(\mathbf{L})$. The initial value of the matrix $\mathbf{A}$ used for optimization is set to $10\mathbf{I} - 5\mathbf{B}$, where $\mathbf{I}$ is a rectangle identity matrix and $\mathbf{B} \in \Re^{r\times d}$ denotes the matrix of all ones. This solving strategy is named as the soft-max method. In particular, when the soft-max method is used for optimization, it is not easy for us to introduce the regularization directly. For the two solving methods, the proposed algorithm can always perform both metric learning and dimensionality reduction.

4. Experiments

In this section, we perform a number of experiments on five real-world image datasets to evaluate the proposed methods. In the first experiment, two solving strategies, the iterative projected gradient and the soft-max method, are compared according to training time and classification error. In the second experiment, we evaluate the proposed method with the state-of-the-art methods, including four histogram metrics (χ^2, QCN (available at http://www.ariel.ac.il/sites/ofirpele/QC/), QCS (available at http://www.ariel.ac.il/sites/ofirpele/QC/), and FastEMD (available at http://www.ariel.ac.il/sites/ofirpele/FastEMD/code/)) and three metric learning methods (ITML (available at http://www.cs.utexas.edu/~pjain/itml/), LMNN (available at http://www.cse.wustl.edu/~kilian/code/files/mLMNN2.4.zip), and GB-LMNN (available at http://www.cse.wustl.edu/~kilian/code/files/mLMNN2.4.zip)), on the image retrieval dataset corel. As the source code of the closely related method χ^2-LMNN [15] is not publicly available, we further perform the full-rank and low-rank metric learning experiments on the four datasets (dslr, webcam, amazon, and caltech). Since the χ^2-LMNN has

Table 1: Summary of the histogram datasets used in the experiments.

Datasets	Samples	Classes	Features
corel	773	10	384
dslr	157	10	800
webcam	295	10	800
amazon	958	10	800
caltech	1123	10	800

also been tested on the above datasets, we can make a direct comparison. There are several parameters to be set in our model. The parameter k is empirically set to **Max**{3, **Min**{9, 10%#NumberofTrainingSamples/#NumberofClasses}}. We fix the parameters μ and β to 0.5 and 50 in our experiments, respectively. Moreover, the parameter λ is set to 1 if the regularization is used. The proposed methods are implemented in standard C++. In our work, all the experiments are executed in a PC with 8 Intel(R) Xeon(R) E5-1620 CPUs (3.6 GHz) and 8 GB main memory.

4.1. Datasets. Table 1 summarizes the basic information of the five histogram datasets used in our experiments. The dataset corel is often used in the evaluation of histogram distance metric [7, 10, 11], which contains 773 landscape images in 10 different classes: people in Africa, beaches, outdoor buildings, buses, dinosaurs, elephants, flowers, horses, mountains, and food. There are 50 to 100 images in each class. All images have two types of representation: SIFT and CSIFT. For SIFT, Harris-affine detector [1] is used to extract 6×8×8 orientation histogram descriptor. The second representation CSIFT is a SIFT-like descriptor for color image. CSIFT takes into account color edges in time of computing the SIFT and skips the normalization step to preserve more distinctive information. The size of final histogram descriptor is also 6 × 8×8. For more detailed information readers can be referred to [10]. As in [7], for each kind of descriptions we select 5 images (numbers $1, 20, \ldots, 40$) from each class to construct the test data of 50 samples and the remaining image as training data. Moreover, each histogram descriptor of dimension 384 is further normalized to sum to one.

The remaining four datasets are all from 10 common object categories (back_pack, bike, calculator, headphones, keyboard, laptop_computer, monitor, mouse, mug, and projector) and are often used in the study of domain adaptation [40, 41]. Therein, dslr contains high-resolution images captured from a digital SLR camera in an office; webcam consists of low-resolution images taken from a web camera; amazon contains medium-resolution images downloaded from online merchants; caltech's images are all from Caltech-256 database [42]. Figure 1 shows several example images from the category of projector in the four datasets. According to the same protocols in the previous work [40], we first resized all images to the same width and converted them to grayscale. The local scale-invariant descriptor detector SURF [2] with the Hessian threshold of 1000 was then used to extract 64-dimensional SURF descriptor. Subsequently, we use k-means clustering algorithm to construct a codebook of size 800 based on a randomly chosen descriptor subset of the amazon dataset. Finally, each image can be represented by a bag of keypoints, which corresponds to a histogram of the number of occurrences of particular visual codebook entry in it. As in corel, each histogram is further normalized to sum to one.

4.2. Comparison of the Two Solving Strategies. In this subsection, we first evaluate the computational efficiency of the two solving strategies: the iterative projected gradient and the soft-max. For a fair comparison, we adopt the same stop criterion and adaptive step-size adjusting strategy for the implementation of two methods. Figure 2 presents the training time of two solving strategies under different projection dimensions on the corel dataset with two kinds of descriptors, SIFT and CSIFT. It can be observed from the figure that the iterative projected gradient method is always several times faster than the soft-max method. The result should not be amazing considering that the soft-max method requires more complex computation of gradient than the former according to (7) and (9). Although the iterative projected gradient method needs to perform the projection step with a complexity of $\mathcal{O}(dr\log(r))$ on each iteration, the soft-max method also requires calculating the matrix $\mathbf{L}$ based on the matrix $\mathbf{A}$, involving the computation of the exponential function of rd times.

We further compare the kNN classification error based on the distance metrics learned by two solving strategies on the corel dataset. The experimental results are given in Figure 3. For the results in the figure, the number of nearest neighbors of kNN is set to 3. From Figure 3, it can be found that the classification error of the iterative projected gradient is lower than that of the soft-max in most cases. One possible reason is that the matrices in the set Θ have less restriction than the $\mathbf{L}$ in (8). Considering training time and classification error, hereafter we use the iterative projected gradient method as the default solving strategy of CDML.

4.3. Image Retrieval Results. In the image retrieval task, we compare the performance of the proposed method with four histogram metrics (χ^2, QCN, QCS, and FastEMD) and three metric learning methods (ITML, LMNN, and GB-LMNN) on the corel dataset. As in [10], we make the images in the test set of the corel as the query images. The 50 nearest neighbors of each query image are searched based on different metrics. For four metric learning methods, CDML, GB-LMNN, LMNN, and ITML, we use the defined training dataset to train the metrics. Specially, for LMNN we utilize the PCA matrix to initialize it and GB-LMNN is initialized by the output matrix of LMNN. The regularization parameter λ of CDML is set to 1. The retrieval results are given in Figure 4. We can see that CDML achieves better performance compared with the competing methods, which performs best on SIFT and ranks second on CSIFT. One key observation is that the retrieval results of GB-LMNN are significantly better than those of other methods on the CSIFT descriptor, which shows the effectiveness of nonlinear transformation method. Moreover, it should be noted that χ^2 metric always performs better than QCN, QCS, and FastEMD; one important reason is that

Figure 1: Example images of the projector in four datasets: dslr, webcam, amazon, and caltech.

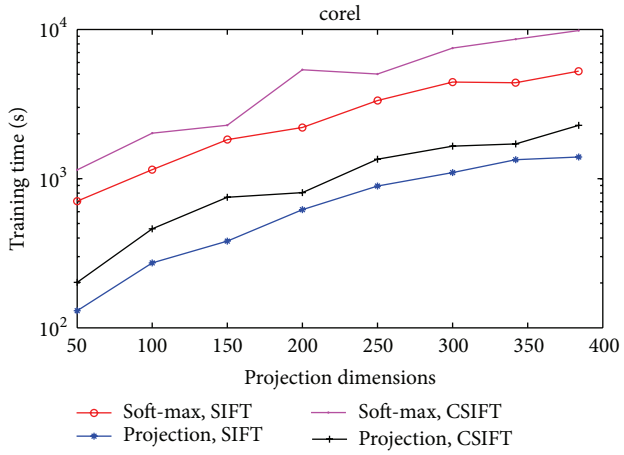

Figure 2: Training time versus projection dimensions on the corel dataset with two types of representation of descriptor. We use projection and soft-max to denote the iterative projected gradient and soft-max method, respectively.

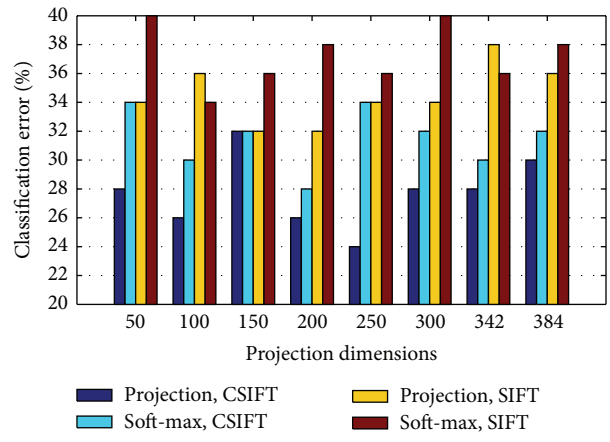

Figure 3: Classification error versus projection dimensions on the corel dataset with two types of representation of descriptor. We use projection and soft-max to denote the iterative projected gradient and soft-max method, respectively.

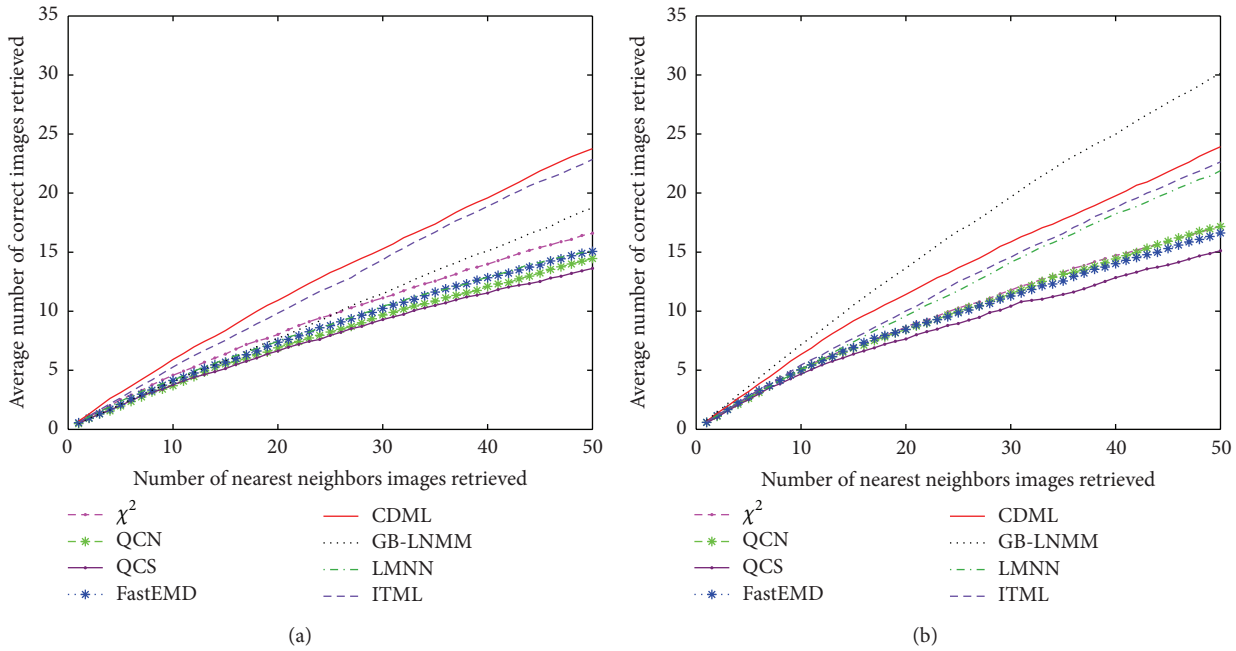

Figure 4: Results for image retrieval under different metrics on the corel dataset. (a) SIFT descriptor and (b) CSIFT descriptor.

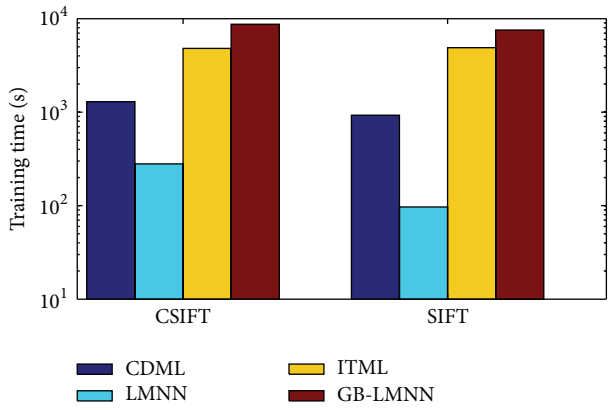

Figure 5: Training time(s) of CDML, LMNN, ITML, and GB-LMNN on two descriptors: CSIFT and SIFT.

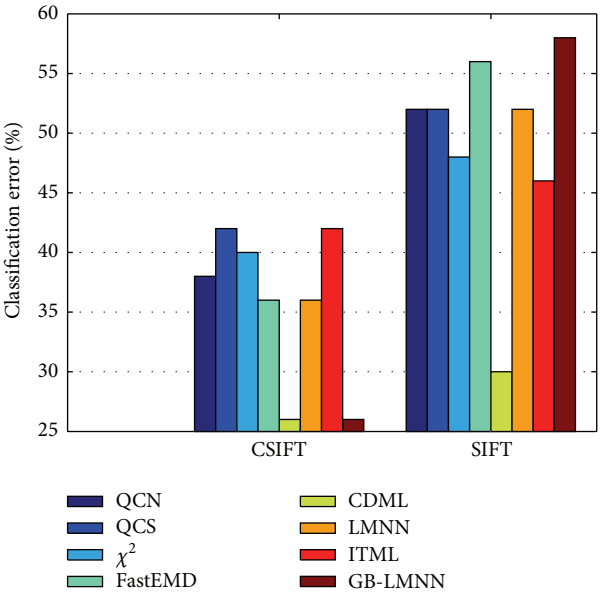

Figure 6: Classification error on the corel dataset.

the latter three methods are mainly to address unnormalized histogram.

Figure 5 compares the training times of the four metric learning methods, that is, CDML, GB-LMNN, LMNN, and ITML. It can be seen that the computational efficiency of CDML ranks second in the four methods. Specially, in average CDML is 9 times faster than the nonlinear metric learning method, GB-LMNN. Actually, the implementation of LMNN and GB-LMNN adopts OpenMP parallel mechanism, while that of CDML does not. Figure 6 compares the kNN ($k = 3$) classification error of χ^2, QCN, QCS, FastEMD, CDML, GB-LMNN, LMNN, and ITML on the test set of the corel. Clearly, CDML always achieves the lowest classification error, and the classification performance of GB-LMNN is unstable.

4.4. Object Classification Results. To investigate the ability of the proposed method under the full-rank and low-rank metric learning cases, we further performed the experiments to compare it with seven different algorithms χ^2, QCS, QCN, ITML, LMNN, GB-LMNN, and χ^2-LMNN on the four object classification datasets, including dslr, webcam, amazon, and caltech. For each dataset, we adopt exactly the same experimental setup as used in [15]: The results of CDML were obtained by averaging over 5 runs on randomly generated 80%/20% splits for training and test. Therefore,

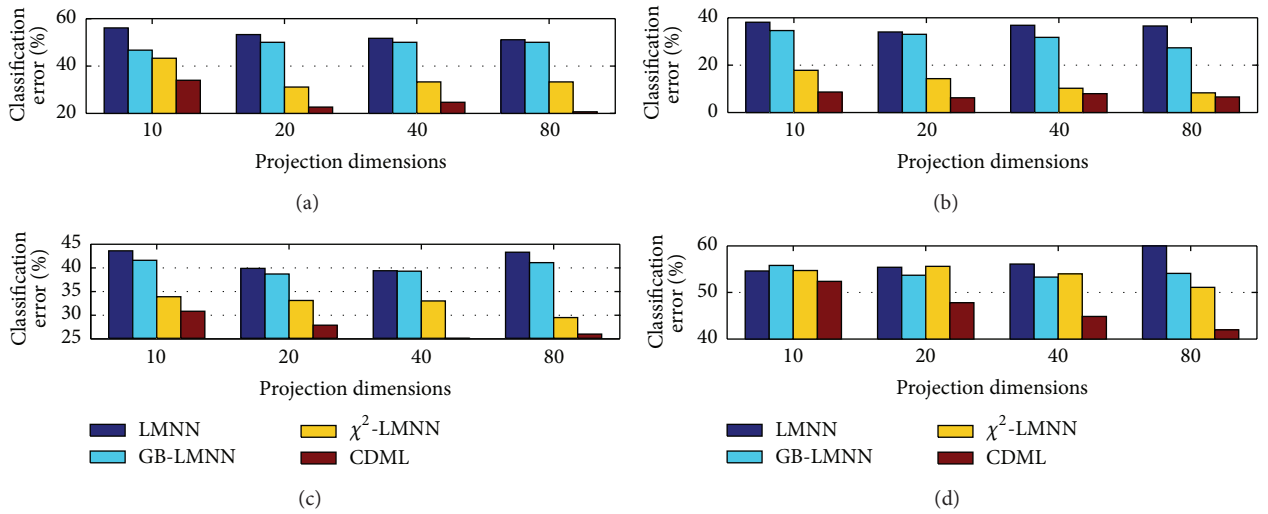

Figure 7: Comparison of four algorithms via the classification error under different projections on four histogram datasets: (a) dslr; (b) webcam; (c) amazon; and (d) caltech.

Table 2: Classification errors (%) of kNN ($k = 3$) using different metrics on the four histogram datasets. Each reported term is an average classification error and a standard deviation over the five runs of cross validation. The minimum classification error of each column is highlighted in bold.

Method	dslr	webcam	amazon	caltech
χ^2 [15]	22.2 ± 1.8	13.0 ± 1.2	34.3 ± 1.0	58.8 ± 1.1
QCS [15]	25.6 ± 2.7	19.4 ± 1.1	33.9 ± 2.0	57.2 ± 1.2
QCN [15]	27.8 ± 4.1	17.5 ± 2.1	34.5 ± 1.5	56.1 ± 1.2
ITML [15]	25.0 ± 3.0	12.4 ± 1.6	31.6 ± 1.2	52.2 ± 2.1
LMNN [15]	28.9 ± 1.6	15.8 ± 3.0	31.8 ± 1.4	50.9 ± 1.4
GB-LMNN [15]	22.9 ± 2.7	12.4 ± 0.9	29.6 ± 1.7	49.8 ± 1.0
χ^2-LMNN [15]	20.6 ± 1.1	8.3 ± 0.9	23.7 ± 0.8	46.5 ± 1.1
CDML	**16.7 ± 4.2**	**5.9 ± 2.3**	**20.8 ± 3.2**	**42.2 ± 2.4**

the direct comparison of CDML with other methods can be made. In what follows, the reported results of the seven algorithms χ^2, QCS, QCN, ITML, LMNN, GB-LMNN, and χ^2-LMNN all come from the literature [15]. Table 2 shows the performance comparison of our method against the methods mentioned above under the full-rank case. Considering χ^2-LMNN without introducing regularizer, we set the regularization parameter of CDML to 0 for a fair comparison. From the table, it can be observed that CDML is the clear winner compared to χ^2, QCS, QCN, ITML, LMNN, GB-LMNN, and χ^2-LMNN according to the classification error. In particular, although CDML and χ^2-LMNN are very similar in the learning model, the former shows significant performance boost on the three datasets (dslr, webcam, and caltech) compared with the latter. In particular, for each dataset χ^2-LMNN needs to perform additional evaluation on a hold-out set so as to determine the adaptive margin parameter l, while CDML does not.

Figure 7 compares classification performance of four metric learning methods LMNN, GB-LMNN, χ^2-LMNN, and CDML under low-rank metric learning case. One can see that for all datasets CDML shows best performance consistently under different projection dimensions among the four metric learning algorithms. The results verify the effectiveness of the proposed method. Moreover, the low classification error of CDML under the projection dimensions 10, 20, 40, and 80 also demonstrates that dimensionality reduction is absolutely effective for histogram data.

4.5. Comparison with the χ^2-LMNN. Since the proposed method is very similar to χ^2-LMNN, in this section we discuss the difference between them. CDML differs from χ^2-LMNN in three major aspects. First, χ^2-LMNN adopts the hinge-loss $u(\rho) = \mathbf{max}(0, l - \rho)$ to construct the objective function, while CDML uses the logistic-loss $u(\rho) = \log(1 + \exp(-50\rho))/50$. Second, χ^2-LMNN uses the soft-max method as the solving strategy, while CDML adopts the iterative projected gradient method. Third, in χ^2-LMNN, the target neighbors of each training sample are determined by the k-nearest neighbors in original metric space and do not change during the learning process. However, when we consider the nearest hits of CDML as the target neighbors, which are dynamically updated according to new distance metric on each iteration, thus, it is interesting to investigate the performance of χ^2-LMNN under the target neighbors being dynamically changed.

In order to evaluate the difference between χ^2-LMNN and CDML, we implement the following four algorithms in standard C++.

(i) *χ^2-LMNN (Soft-Max).* This is the original χ^2-LMNN that uses the soft-max method to solve the simplex-preserving transformation matrix. The number of target neighbors is set to 3.

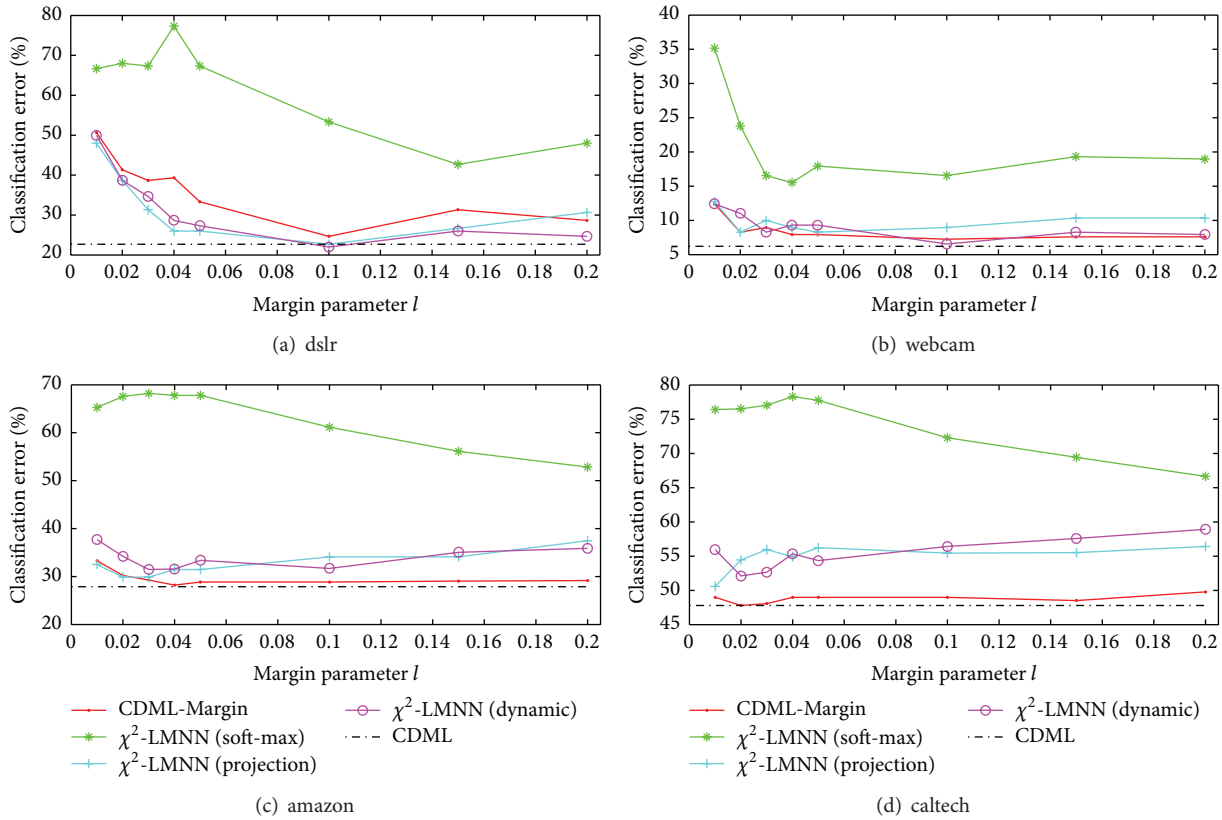

FIGURE 8: The effect of margin parameter on classification error. The margin parameter l is set to 0.01, 0.02, 0.03, 0.04, 0.05, 0.1, 0.15, and 0.2.

(ii) *χ^2-LMNN (Projection).* This is the χ^2-LMNN using the iterative projected gradient method as the solving strategy. The number of target neighbors is set to 3.

(iii) *χ^2-LMNN (Dynamic).* This is the χ^2-LMNN using the iterative projected gradient method as the solving strategy. The target neighbors of each sample are dynamically updated after obtaining the novel simplex-preserving transformation matrix on each iteration. They are the k-nearest neighbors of each sample under the new chi-squared distance metric. The number of target neighbors is set to 3.

(iv) *CDML-Margin.* This is the CDML using the hinge-loss $u(\rho) = \mathbf{max}(0, l - \rho)$ as the utility function instead of $u(\rho) = \log(1 + \exp(-50\rho))/50$, where l is an additional margin parameter as in [15]. The setting about the parameters of nearest neighbor is the same as that of the CDML.

On the four histogram datasets, dslr, webcam, amazon, and caltech, we conducted low-rank fivefold cross validation experiment to evaluate the methods mentioned above. For the four algorithms χ^2-LMNN (soft-max), χ^2-LMNN (projection), χ^2-LMNN (dynamic), and CDML-Margin, they all require specifying a margin parameter l. In our experiment, the value of margin parameter l is set to 0.01, 0.02, 0.03, 0.04, 0.05, 0.1, 0.15, and 0.2. The projection dimension is set to 20. Figure 8 shows the effect of margin parameter on classification error. The reported results are obtained by averaging over 5 runs. In order to make a comparison, the classification errors of CDML are also given. From the figure, we can see that the margin parameter has a significant effect on the performance of the margin-based methods. Different methods and datasets require distinct margin parameters. It can be observed that CDML is the clear winner compared to four margin-based methods χ^2-LMNN (soft-max), χ^2-LMNN (projection), χ^2-LMNN (dynamic), and CDML-Margin. This indicates that the logistic-loss is better than the hinge-loss in metric learning for histogram data. In order to explain it, we further compare the logistic-loss and the hinge-loss in Figure 9. Evidently, the logistic-loss is more suitable to histogram data since the chi-squared distance margin between histogram data is often very small. Moreover, χ^2-LMNN (soft-max) shows the worst performance on all datasets and CDML-Margin outperforms χ^2-LMNN (projection) in most cases. χ^2-LMNN (dynamic) performs better than χ^2-LMNN (projection) in some cases, while it does not in other cases, which implies that the introduction of dynamic target neighbor cannot ensure boosting the performance of χ^2-LMNN (projection). One possible reason is that the used data is insensitive to noise in it. From Figure 8, we summarize that the promising performance of CDML against χ^2-LMNN should be attributed to the following three reasons: (1) Maintaining the same margin for all histogram data is unsuitable. (2) The iterative projected gradient method

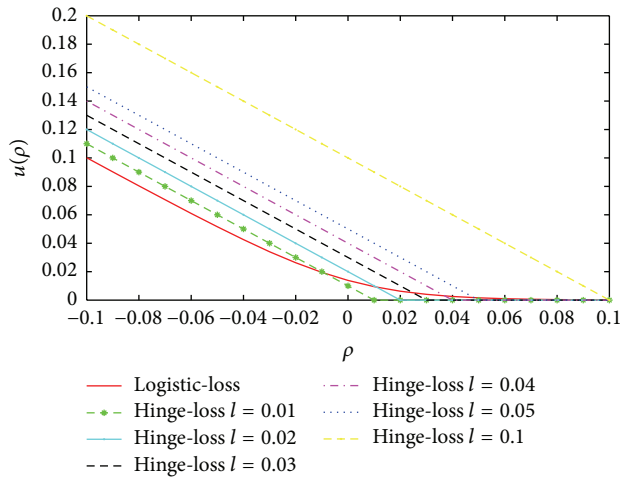

Figure 9: The comparison between the logistic-loss $u(\rho) = \log(1 + \exp(-50\rho))/50$ and the hinge-loss $u(\rho) = \mathbf{max}(0, l-\rho)$ in the interval $[-0.1, 0.1]$.

is more reasonable compared with the soft-max method. (3) In CDML, the nearest hits and misses are adopted as the target neighbors. Thus, even the same hinge-loss and dynamic adjustments on neighbors are adopted; CDML-Margin can outperform χ^2-LMNN (dynamic) in most cases, which indicates that the margins defined based on nearest hits and misses generally result in lower classification error for object classification.

In order to compare the difference between CDML and χ^2-LMNN in the training set with the varying size, we further perform experiment on the webcam dataset. A random subset with n (=5, 10, 15, 20) samples per class was taken to form the training set. The rest of the dataset was considered to be the testing set. For each given n, we average the results over 5 random splits. In particular, we use Euclidean distance and chi-squared distance as the benchmarks. The projection dimension of CDML and χ^2-LMNN is set to 80. Table 3 shows the classification errors. As can be seen, the Euclidean distance performed the worst. The classification performance of CDML is significantly better than that of the χ^2-LMNN, which means that the latter is more sensible than CDML to the size of the training set.

5. Conclusion

To address the matching of histogram data, we propose a novel nearest neighbor-based algorithm to efficiently learn chi-squared distance based on maximizing the margin while maintaining the compactness between each training sample and its nearest hits. The proposed method could obtain a simplex-preserving linear transformation, which makes the learned metric a chi-squared distance in the transformed space. The two solving strategies, the iterative projected gradient and the soft-max method, can be used to solve our method. Experimental results show that the former is more efficient. With the iterative projected gradient method, the regularizer can be introduced naturally. In the comparative experiments on five real-world histogram datasets, the proposed method demonstrates very promising performance in both classification error and efficiency in comparison with the state-of-the-art methods. In the future, we will investigate the other choices of the objective function [18, 43] and consider the robustness against cross-bin distortion to design proper regularization terms. The C++ source code of CDML is freely available from the website https://sites.google.com/site/codeofcdml/.

Table 3: Performance comparisons on the webcam dataset. The number in the first row indicates the number of samples per class being used to construct the training set. Each reported term is an average classification error and a standard deviation. The minimum classification error of each column is highlighted in bold.

Method	5-train	10-train	15-train	20-train
Euclidean distance	52.8 ± 1.6	38.4 ± 3.0	31.2 ± 2.8	27.4 ± 1.5
χ^2 distance	37.6 ± 3.2	23.3 ± 3.1	14.1 ± 3.4	10.3 ± 3.6
χ^2-LMNN	31.9 ± 2.4	23.2 ± 1.4	19.7 ± 3.2	19.0 ± 1.9
CDML	**24.8 ± 2.2**	**14.5 ± 2.1**	**10.1 ± 2.6**	**6.3 ± 3.2**

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The authors would like to reveal that this research was supported partially by Foundation of Henan Educational Committee of China under Grant nos. 14A520027 and 14A520041.

References

[1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[2] H. Bay, A. Ess, T. Tuytelaars, and L. van Gool, "Speeded-up robust features (surf)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[3] A. Oliva and A. Torralba, "Modeling the shape of the scene: a holistic representation of the spatial envelope," *International Journal of Computer Vision*, vol. 42, no. 3, pp. 145–175, 2001.

[4] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, vol. 1, pp. 886–893, San Diego, Calif, USA, June 2005.

[5] H. Ling and K. Okada, "Diffusion distance for histogram comparison," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06)*, vol. 1, pp. 246–253, June 2006.

[6] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *International Journal of Computer Vision*, vol. 40, no. 2, pp. 99–121, 2000.

[7] O. Pele and M. Werman, "The quadratic-chi histogram distance family," in *Computer Vision—ECCV 2010*, vol. 6312 of *Lecture*

Notes in Computer Science, pp. 749–762, Springer, Berlin, Germany, 2010.

[8] H. Ling and K. Okada, "An efficient earth mover's distance algorithm for robust histogram comparison," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 5, pp. 840–853, 2007.

[9] O. Pele and M. Werman, "A linear time histogram metric for improved SIFT matching," in *Computer Vision—ECCV 2008*, vol. 5304 of *Lecture Notes in Computer Science*, pp. 495–508, Springer, Berlin, Germany, 2008.

[10] O. Pele and M. Werman, "Fast and robust earth mover's distances," in *Proceedings of the 12th International Conference on Computer Vision (ICCV '09)*, pp. 460–467, October 2009.

[11] O. Pele and B. Taskar, "The tangent earth mover's distance," in *Geometric Science of Information*, F. Nielsen and F. Barbaresco, Eds., vol. 8085 of *Lecture Notes in Computer Science*, pp. 397–404, Springer, Berlin, Germany, 2013.

[12] M. Cuturi and D. Avis, "Ground metric learning," *Journal of Machine Learning Research*, vol. 15, pp. 533–564, 2014.

[13] F. Wang and L. J. Guibas, "Supervised earth mover's distance learning and its computer vision applications," in *Computer Vision—ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7–13, 2012, Proceedings, Part I*, vol. 7572 of *Lecture Notes in Computer Science*, pp. 442–455, Springer, Berlin, Germany, 2012.

[14] S. Noh, "χ^2 metric learning for nearest neighbor classification and its analysis," in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR '12)*, pp. 991–995, 2012.

[15] K. Dor, T. Stephen, W. Kilian, S. Fei, and L. Gert, "Nonlinear metric learning," *Advances in Neural Information Processing Systems 25*, pp. 2582–2590, 2012.

[16] T. Le and M. Cuturi, "Generalized aitchison embeddings for histograms," *JMLR: Workshop and Conference Proceedings*, vol. 29, pp. 293–308, 2013.

[17] E. P. Xing, M. I. Jordan, S. Russell, and A. Ng, "Distance metric learning with application to clustering with sideinformation," in *Advances in Neural Information Processing Systems*, vol. 15, pp. 505–512, 2002.

[18] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov, "Neighbourhood components analysis," in *Advances in Neural Information Processing Systems*, vol. 17, pp. 513–520, 2004.

[19] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *Journal of Machine Learning Research*, vol. 10, pp. 207–244, 2009.

[20] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, "Information-theoretic metric learning," in *Proceedings of the 24th International Conference on Machine Learning (ICML '07)*, pp. 209–216, June 2007.

[21] W. Bian and D. Tao, "Constrained empirical risk minimization framework for distance metric learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 8, pp. 1194–1205, 2012.

[22] F. Wang, W. Zuo, L. Zhang, D. Meng, and D. Zhang, "A Kernel classification framework for metric learning," *IEEE Transactions on Neural Networks and Learning Systems*, 2014.

[23] C.-C. Chang, "A boosting approach for supervised mahalanobis distance metric learning," *Pattern Recognition*, vol. 45, no. 2, pp. 844–862, 2012.

[24] C. Shen, J. Kim, F. Liu, L. Wang, and A. van den Hengel, "Efficient dual approach to distance metric learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 2, pp. 394–406, 2014.

[25] P. Yang, K. Huang, and C.-L. Liu, "A multi-task framework for metric learning with common subspace," *Neural Computing and Applications*, vol. 22, no. 7-8, pp. 1337–1347, 2013.

[26] T. Hastie and R. Tibshirani, "Discriminant adaptive nearest neighbor classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 6, pp. 607–616, 1996.

[27] J. Wang, A. Kalousis, and A. Woznica, "Parametric local metric learning for nearest neighbor classification," in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., vol. 25, pp. 1601–1609, Curran Associates, 2012.

[28] Y. Mu, W. Ding, and D. Tao, "Local discriminative distance metrics ensemble learning," *Pattern Recognition*, vol. 46, no. 8, pp. 2337–2349, 2013.

[29] M. Sugiyama, "Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis-squared distance metric learning for histogram data 11 sis," *Journal of Machine Learning Research*, vol. 8, pp. 1027–1061, 2007.

[30] L. Torresani and K. C. Lee, "Large margin component analysis," in *Advances in Neural Information Processing Systems*, pp. 1385–1392, 2006.

[31] M. Soleymani Baghshah and S. Bagheri Shouraki, "Non-linear metric learning using pairwise similarity and dissimilarity constraints and the geometrical structure of data," *Pattern Recognition*, vol. 43, no. 8, pp. 2982–2992, 2010.

[32] J. Wang, H. T. Do, A. Woznica, and A. Kalousis, "Metric learning with multiple kernels," in *Advances in Neural Information Processing Systems*, J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, Eds., vol. 24, pp. 1170–1178, Curran Associates, 2011.

[33] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, vol. 1, pp. 539–546, June 2005.

[34] Z. Xu, K. Q. Weinberger, and O. Chapelle, "Distance metric learning for kernel machines," http://arxiv.org/abs/1208.3422.

[35] B. Aur&*apos*;elien, H. Amaury, and S. Marc, "A survey on metric learning for feature vectors and structured data," http://arxiv.org/abs/1306.6709.

[36] B. Kulis, "Metric learning: a survey," *Foundations and Trends in Machine Learning*, vol. 5, no. 4, pp. 287–364, 2012.

[37] L. Yang and R. Jin, *Distance Metric Learning: A Comprehensive Survey*, Michigan State Universiy, 2006.

[38] R. Gilad-Bachrach, A. Navot, and N. Tishby, "Margin based feature selection—theory and algorithms," in *Proceedings of the 21st International Conference on Machine Learning (ICML '04)*, pp. 43–50, ACM, 2004.

[39] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, "Efficient projections onto the *l*1-ball for learning in high dimensions," in *Proceedings of the 25th International Conference on Machine Learning*, pp. 272–279, ACM, 2008.

[40] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *Computer Vision—ECCV 2010*, vol. 6314 of *Lecture Notes in Computer Science*, pp. 213–226, Springer, Berlin, Germany, 2010.

[41] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 2066–2073, June 2012.

[42] G. Griffin, A. Holub, and P. Perona, *Caltech-256 Object Category Dataset*, 2007.

[43] A. Globerson and S. T. Roweis, "Metric learning by collapsing classes," in *Advances in Neural Information Processing Systems*, pp. 451–458, 2005.