Quasi Cosine Similarity Metric Learning

Xiang Wu, Zhi-Guo Shi and Lei Liu

School of Computer and Communication Engineering, University of Science and Technology Beijing, No.30 Xueyuan Road, Haidian District, Beijing, China

Abstract. It is vital to select an appropriate distance metric for many learning algorithm. Cosine distance is an efficient metric for measuring the similarity of descriptors in classification task. However, the cosine similarity metric learning (CSML)[1] is not widely used due to the complexity of its formulation and time consuming. In this paper, a Quasi Cosine Similarity Metric Learning (QCSML) is proposed to make it easy. The normalization and Lagrange multipliers are employed to convert cosine distance into simple formulation, which is convex and its derivation is easy to calculate. The complexity of the QCSML algorithm is $O(t \times p \times d)$[1], while the complexity of CSML is $O(r \times b \times g \times s \times d \times m)$[2]. The experimental results of our method on UCI datasets for classification task and LFW dataset for face verification problem are better than the state-of-the-art methods. For classification task, the proposed approach is employed on Iris, Ionosphere and Wine dataset and the classification accuracy and the time consuming are much better than the compared methods. Moreover, our approach obtains 92.33% accuracy for face verification on unrestricted setting of LFW dataset, which outperforms the state-of-the-art algorithms.

1 Introduction

An appropriate distance measure (or metric) is fundamental to many supervised and unsupervised learning algorithm such as k-means, kernel method, the nearest neighborhood classification and so on. Besides, it is important for varieties of application such as image retrieval or face recognition to choose a proper distance metric to measure the similarity or dissimilarity between different images. Therefore, to apply an appropriate distance metric for practical applications, lots of distance metric learning algorithm methods are proposed to find the special latent relevance between different samples.

However, choosing a proper distance metric is highly problem-specific and ultimately dictates the success of the actual learning algorithm. Many existing

[1] The parameters t, p, d represent the number of iterations, the dimensionality of descriptors and the compressed features.

[2] From the paper[1], r is the number of iterations used to optimize the projection matrix, b is the number of values tested in cross validation process, g is the number of steps in the Conjugate Gradient method, s is the number of training data, d and m are the dimensions of projection matrix.

algorithms for metric learning have been shown to perform well in different application, but most of them do not perform well in high dimensional input. The high dimensional descriptor exists in a wide range of application such as image retrieval[2], face recognition[1, 3] and natural language processing. In these occasions, it is necessary to compress the high dimensional descriptors into low dimensional ones due to high computation and storage. So the distance metric learning is not only used to make data separately but also to compress the high dimensional vectors.

Recently, Mahalanobis distance metric has been widely applied in many aspects as a metric learning measure[4–6]. Xing et al. [4] applied semidefinite programming (SDP) objective function to learn a Mahalanobis distance metric for clustering. They minimize the sum of Euclidean distance between similarity labeled inputs and maintained a lower bound on the distance between different ones. Davis et al. [5] use information-theoretic regularization term for Euclidean distance. Moreover, Qi et al. [6] formulate a sparse Mahalanobis matrix which reflects the intrinsic nature of sparsity. They impose a sparse prior and show the obtained l_1-penalized Log-Determinant optimization problem for sparse metric can be minimized by a block coordinate descent algorithm [7], which is faster than SDP method widely used in metric learning.

Moreover, cosine similarity is an efficient distance metric to comparing the difference between vectors and it is an effective alternative to Euclidean distance in metric learning problem. Nguyen proposed a cosine similarity metric learning (CSML)[1] which can improve the generalization ability of an existing metric significantly in most cases. But it is not useful for high dimensional descriptors because of the highly memory used and computing burden for gradient descent method. In paper [3], Cao proposed a similarity metric learning (SML) which combines Euclidean distance and cosine similarity as the metric learning objective function. The formulation of similarity metric learning is convex and Cao optimized the dual formulation to obtain the global solution instead.

In this paper, due to the high computation and memory used, we proposed a Quasi Cosine Similarity Metric Learning (QCSML) for high dimensional vectors. There are two main contributions of our method. One is that we have introduced a novel solution for cosine similarity metric learning problem. The other is that QCSML is efficient for high dimensional vectors which are usually as the descriptors for face recognition, image classification and image retrieval. QCSML is not only discriminative for classification tasks but also used for dimensionality reduction.

The paper is organized as following. In section 2, we review distance metric and similarity metric for classification. In section 3, we formulate the detail of Quasi Cosine Similarity Metric Learning. The learning algorithm and gradient descent optimization method is introduced. Section 4 evaluates the proposed Quasi Cosine Similarity Metric Learning algorithm on UCI datasets for classification and LFW datasets for face recognition. Finally, the conclusion is given in section 5.

2 Preliminary

In this section, we briefly review general distance metric learning[4], cosine similarity metric learning[1] and similarity metric learning[3].

2.1 Metric Learning

Given a set of points $X = \{x_1, \ldots, x_n\}, x_i \in \mathbb{R}^d$, we can define a positive definite matrix $A \in \mathbb{R}^{d \times d}$ which represents the Mahalanobis distance.

$$d_A^2(x_i, x_j) = (x_i - x_j)^T A (x_i - x_j) \tag{1}$$

The goal of metric learning is to adapt the metric function to the problem using information from the training datasets. Because of positive definite characteristics, the matrix A can be decomposed as $A = W^T W$, $W \in \mathbb{R}^{p \times d}$, therefore, the metric function can be shown as

$$d_W^2(x_i, x_j) = (x_i - x_j)^T W^T W (x_i - x_j) = \|Wx_i - Wx_j\|_2^2 \tag{2}$$

Here, we assume we have known the prior knowledge about the relationship constraining the similarity or dissimilarity between pairs of points.

$$\begin{array}{ll} S : (x_i, x_j) \in S & \text{if } x_i \text{ and } x_j \text{ are similar} \\ D : (x_i, x_j) \in D & \text{if } x_i \text{ and } x_j \text{ are dissimilar} \end{array} \tag{3}$$

This gives the optimization problem

$$\begin{array}{l} \min \sum_{(x_i, x_j) \in S} \|Wx_i - Wx_j\|_2^2 \\ \text{s.t.} \quad \sum_{(x_i, x_j) \in D} \|Wx_i - Wx_j\|_2^2 \geq 1 \end{array} \tag{4}$$

Then in paper [4], Xing introduced the efficient algorithm using the Newton-Raphson method to optimize the objective function.

2.2 Cosine Similarity Metric Learning

Compared with distance metric, cosine similarity between two vectors can be defined as

$$d_W^2(x_i, x_j) = \frac{(Wx_i)^T Wx_j}{\|Wx_i\| \|Wx_j\|} \tag{5}$$

Given the similar sets S and dissimilar sets D, the objective function can be shown as

$$\max \sum_{(x_i, x_j) \in S} d_W^2(x_i, x_j) - \alpha \sum_{(x_i, x_j) \in D} d_W^2(x_i, x_j) \tag{6}$$

where d_W^2 is defined as Eq.(5)

In this optimization problem, it is difficult to calculate the derivation of objective function which is used for gradient descent method. In paper [1], Nguyen gives the gradient as

$$\begin{aligned}\frac{\partial d_W^2(x_i,x_j)}{\partial W} &= \frac{\partial(\frac{u(W)}{v(W)})}{\partial W} \\ &= \frac{1}{v(W)}\frac{\partial u(W)}{\partial W} - \frac{u(W)}{v(W)^2}\frac{\partial v(W)}{\partial W}\end{aligned} \tag{7}$$

where

$$\begin{cases}\frac{\partial u(W)}{\partial W} = W(x_i x_j^T + x_j x_i^T) \\ \frac{\partial v(W)}{\partial W} = \frac{\|Wx_j\|}{\|Wx_i\|}Wx_i x_i^T - \frac{\|Wx_i\|}{\|Wx_j\|}Wx_j x_j^T\end{cases} \tag{8}$$

As is shown in Eq.(8), the complexity of the gradient is too high to compute if the dimensionality of descriptors x_i is large. For example, if the descriptors are high dimensional Fisher Vector (FV) which is about 67586-d in the paper [8], the cosine similarity metric learning will be inefficient and ineffective for dimensionality reduction and data classification.

3 Quasi Cosine Similarity Metric Learning

In this section, we first give the objective function of Quasi Cosine Similarity Metric Learning (QCSML), and introduce the hinge-loss to represent the objective function. The stochastic gradient descent (SGD) is used to optimize the objective function.

3.1 Problem Formulation

We begin this section with some notation definitions. Our goal is to learn the cosine similarity in Eq.(5) from a set of feature space $X = \{x_1, \ldots, x_n\}, x_i \in \mathbb{R}^d$. Obviously, the gradient of the cosine similarity function in Eq.(7)(8) is complex and if the dimension of feaature d is large, the optimization processing will take up lots of memory which personal computer cannot afford.

As is shown in Eq.(5), the cosine similarity metric can be written as

$$d_W^2(x_i,x_j) = \frac{(Wx_i)^T Wx_j}{\|Wx_i\|\|Wx_j\|} \geq \frac{(Wx_i)^T Wx_j}{(\|W\|\|x_i\|)(\|W\|\|x_j\|)} \tag{9}$$

because of the inequality $\|ab\| \leq \|a\|\|b\|$. Therefore, the cosine similarity metric can be written as

$$d_W^2(x_i,x_j) \geq \frac{(Wx_i)^T Wx_j}{\|W\|^2\|x_i\|\|x_j\|} \tag{10}$$

Here, to simplify the cosine similarity metric in Eq.(10), we can give the prior knowledge about the projection matrix W which is represented as

$$\|W\|_F^2 = \mathrm{Tr}(WW^T) = \sum_{i=1}^{p}\sum_{j=1}^{d} W_{ij}^2 = 1 \tag{11}$$

where $\|\|_F$ is the Frobenius norm of matrix. And then we normalize the descriptors $\{x_i\}$ in feature space which means $\|x_i\|_2 = 1, x_i \in \mathbb{R}^d, i = 1, \ldots, n$.

With these constrains, the cosine similarity metric can be written as

$$d_W^2(x_i, x_j) \geq \tilde{d}_W^2(x_i, x_j) = (Wx_i)^T Wx_j \tag{12}$$

Then we define the label y_{ij} represents the similarity or dissimilarity between a pair of two vectors (x_i, x_j). Therefore, we can denote a threshold $b \in \mathbb{R}$ that the pair is similar or different if the cosine similarity $d_W^2(x_i, x_j)$ is upon or below the threshold b. Therefore, these constrains can be defined as

$$y_{ij}(\tilde{d}_W^2(x_i, x_j) - b) > 1 \tag{13}$$

where $y_{ij} = 1$ if x_i and x_j are similar, which means $(x_i, x_j) \in S$, and $y_{ij} = -1$ otherwise.

According the constrains in Eq.(13), the quasi cosine similarity metric learning problem can be defined as

$$\begin{array}{c} \min \sum_{i,j} \max\left[1 - y_{ij}(\tilde{d}_W^2(x_i, x_j) - b), 0\right] \\ \text{s.t.} \quad \|x_i\|_2 = 1, x_i \in S \cup D, i = 1, \ldots, n \\ \mathrm{Tr}(WW^T) = 1 \end{array} \tag{14}$$

The objective function is to make the margin between the positive and negative pairs to be large, since it is hinge-loss. The hinge loss function is used for max-margin classification problem, mostly notably for support vector machine (SVM).

Due to the only equality constraint and the normalization of the input vectors $\{x_i\}$, the Lagrange multipliers method can be used to convert the QCSML problem into an unconstrained problem and the converted objective function can be written as

$$\min \sum_{i,j} \max\left[1 - y_{ij}(\tilde{d}_W^2(x_i, x_j) - b), 0\right] + \lambda(\mathrm{Tr}(WW^T) - 1) \tag{15}$$

where λ is the Lagrange multipliers and $\lambda > 0$. Because of the $W^T W$ is positive definite matrix, the objective function is convex. The constrain $\mathrm{Tr}(WW^T) = \|W\|_F^2 = 1$ can also be treated as the regularization to prevent overfitting and in this aspect, λ can be considered as the trade-off parameters.

3.2 Algorithm and Complexity

Due to the convexity of objective function, we can use the gradient descent method to get the global value of Quasi Cosine Similarity Metric Learning.

Instead of Conjugate Gradient method[1], we employ stochastic gradient descent (SGD) method[9] to optimize the objective function due to large dataset. The projection matrix W can be updated as following

$$W_{t+1} = \begin{cases} W_t & \text{if} \quad y_{ij}(d_W^2(x_i, x_j) - b) > 1) \\ W_t - \alpha \frac{\partial f(W)}{\partial W} & \text{otherwise} \end{cases} \tag{16}$$

Algorithm 1: Quasi Cosine Similarity Metric Learning Optimization

Input : Train data: $(X, y), X = \{x_i\} \subseteq \mathbb{R}^d, \|x_i\| = 1, y_i \in \{-1, 1\}$
Output: The parameters:$\Theta = \{W, b\}$

begin
  Parameters initialization;
  `/* Initialize` W `*/`
  Set W_{init} =PCA_whitening(X);
  `/* Initialize` b `*/`
  (X_s, y_s) =sample(X,y);
  $\phi_s = WX_s$;
  score$_s = \tilde{d}_w^2(x_i, x_j) \quad \forall x_i, x_j \in X_s$;
  b_{init} = accuracy_best(score$_s, y_s$);
  `/* SGD iteration */`
  for $t = 1$ **to** n **do**
    switch y_t **do**
      case *positive*
        score $= \tilde{d}_W^2(x_i, x_j) \quad \forall x_i, x_j \in X_t$;
        if *score* $< b + 1$ **then**
          $W_{t+1} = W_t - \alpha \frac{\partial \tilde{d}_W^2}{\partial W}$;
          $b_{t+1} = b_t - \gamma_b$;
        break;
      case *negative*
        score $= \tilde{d}_W^2(x_i, x_j) \quad \forall x_i, x_j \in X_t$;
        if *score* $> b - 1$ **then**
          $W_{t+1} = W_t + \alpha \frac{\partial \tilde{d}_W^2}{\partial W}$;
          $b_{t+1} = b_t + \gamma_b$;
        break;
  return $\Theta = \{W, b\}$;

where α is learning rate, t is the number of iterations and $f(W)$ represents the objective function. The gradient of objective function is written as

$$\frac{\partial f(W)}{\partial W} = y_{ij} W (x_i x_j^T + x_j x_i^T) + \lambda W \tag{17}$$

where λ is the parameter which is set by us. The threshold b can be updated by

$$b_{t+1} = \begin{cases} b_t & \text{if} \quad y_{ij}(\tilde{d}_W^2(x_i, x_j) - b) > 1) \\ b_t - \gamma_b y_{ij} & \text{otherwise} \end{cases} \tag{18}$$

where γ_b is the threshold bias.

We can use SGD to update the parameters $\Theta = (W, b)$. The detail of Quasi Cosine Similarity Metric Learning is given in Algorithm.1. The choice of parameters α and λ is important for the optimization. The learning rate α controls the

speed of gradient descent for the optimization processing. With the SGD method for optimization, the learning rate α should be smaller than L-BFGS[10] or conjugate gradient[1] because we calculate the gradient only use one sample (x_i, x_j) from data set at every iteration. Moreover, the trade-off parameter λ is also important for optimization if it is suitable or not. Although the objective function is convex, we also need to give a good initialization for projection matrix W due to the SGD method. K-means and PCA-whitening are both choices for initialization and the effectiveness will be performed in section 4.

According to the Algorithm.1, the complexity of computing the gradients of objective function is $O(p \times d)$ where d is the dimensionality of descriptors x_i and p is the dimension of reduction by projection matrix W. Therefore, the complexity of QCSML algorithm is $O(t \times p \times d)$ where t is the number of iterations to update the projection matrix W by SGD. It is faster than CSML which the complexity is $O(r \times b \times g \times s \times d \times m)$, where r is the number of iterations used to optimize the projection matrix, b is the number of values tested in cross validation process, g is the number of steps in the Conjugate Gradient method, s is the number of training data, d and m are the dimensions of projection matrix. And the experiments of QCSML for time consuming are in next section.

4 Experiments

In this section, we employ the proposed Quasi Cosine Similarity Metric Learning (QCSML) on various benchmark UCI datasets and LFW datasets[11] for face recognition.

4.1 UCI Datasets Classification

We evaluate the algorithm on three UCI datasets: Iris[3], Ionosphere[4] and Wine[5]. And we deal with three UCI benchmark as following:

1. Iris dataset: This dataset has 150 instances for 3 classes (50 in each of 3 classes). The dimension of descriptors is 4 and we use 120 instances for training (40 in each class) and 30 instances for testing.
2. Ionosphere dataset: This dataset has 351 instances for binary classification task, which dimension of each feature is 34. We use 200 instances for training data and the rest for testing.
3. Wine dataset: This dataset has 178 instances for 3 classes and the number of attributes is 13. We consider 150 instances as training data and the rest as testing data.

The proposed QCSML method is compared with the following algorithm for two aspects: classification performance and computational costs.

[3] https://archive.ics.uci.edu/ml/datasets/Iris
[4] https://archive.ics.uci.edu/ml/datasets/Ionosphere
[5] https://archive.ics.uci.edu/ml/datasets/Wine

Table 1. Classification error rates(%) for different distances across various UCI benchmark datasets.

Algorithm	Iris	Ionosphere	Wine
Euclidean	4.00	14.86	4.5
InvCov	8.67	17.71	43.82
LMNN[12]	3.34	14.29	2.25
ITML[5]	3.00	17.14	3.94
SDML(Identity Matrix)[6]	2.00	13.71	0.5618
SDML(Inverse Covariance)[6]	2.00	12	**0**
QCSML(Random Projection)	4.34	13.49	5.81
QCSML(K-means)	**1.04**	7.99	**0**
QCSML(PCA-whitening)	2.22	**6.93**	**0**

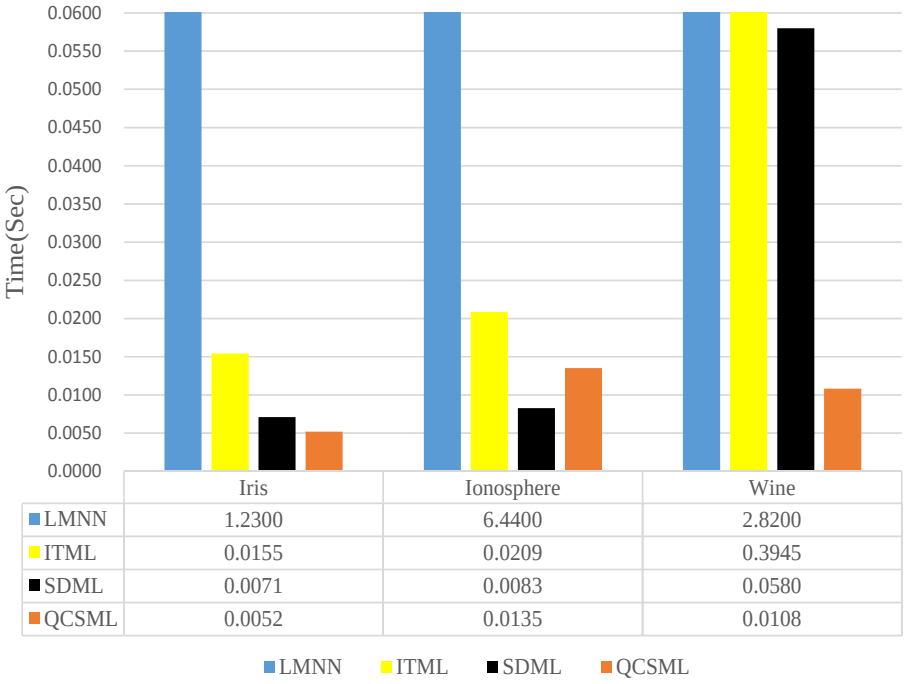

Fig. 1. Training time used different distance metric learning on different datasets

1. **Euclidean**: The squared Euclidean distance $\|x_i - x_j\|_2^2$ as a baseline algorithm for classification.
2. **InvCov**: A Mahalanobis distance parameterized by the inverse of sample covariance. It is equivalent to performing PCA over the input data and then computing the squared Euclidean distance in the transformed space.
3. **LMNN**: Large margin nearest neighbor method is proposed by [12]. This method trains the classifier to separate different classes by a large margin.
4. **ITML**: Information-theoretic metric learning is proposed by [5]. It formulates to learn the Mahalanobis matrix by optimizing the differential relative entropy loss function.
5. **SDML**: Sparse distance metric learning is proposed by [6]. It formulates the loss funtion by log-determinant divergence with a prior knowledge M_0 and the L1-norm regularization for sparsity.

The experimental result is illustrated in Table 1. We can see that the proposed QCSML has the smaller error rates across the datasets compared with other distance metric learning. On the other hand, the PCA-whitening initialization method performs better than other initialization.

Finally, we compare the computational costs of these metric learning algorithms. Figure 1proves the computing efficiency of the proposed QCSML algorithm. We find our method is faster than LMNN, ITML and SDML in most cases. Moreover, with the different gradient descent method employing, our algorithm will be faster if the input data is large and high dimensional because SDML and ITML algorithm need to compute the gradient by all the training data every iteration, while QCSML only need one sample per each iteration.

4.2 LFW Dataset Face Recognition

In this section, we show the performance of the proposed QCSML metric learning on LFW dataset[6] in detail.

LFW dataset contains 13233 images of 5749 people for face verification. For evaluation, the face data is divided in 10 folds which contain different identities and 600 face pairs for evaluation. There are two evaluation setting about LFW training and testing: restricted and unrestricted. In restricted setting, the predefine image pairs is fixed by author (each fold contains 5400 pairs for training and 600 pairs for testing). And in unrestricted setting, the identities of people within each fold for training is allowed to be much larger.

For face verification, it is also important to extract robust descriptors for representing the images. In this paper, we employ Fisher Vector (FV)[13] which is widely used in image classification[14], image retrieval[15] and face recognition[8]. We extract dense SIFT for each aligned image and learn Gaussian Mixture Model (GMM) parameters by EM algorithm. Then the local descriptors are encoded into Fisher Vectors via GMM parameters.

[6] http://vis-www.cs.umass.edu/lfw/

Table 2. Comparison of QCSML method with other state-of-the-art methods in restricted setting of LFW.

Method	Dimension	Accuracy(%)
Combined B/G sample based methods, aligned[16]	-	86.83 ± 0.34
LDML, funneled[17]	-	79.27 ± 0.60
DML-eig combined, funneled & aligned[18]	-	85.65 ± 0.56
LBP+CSML, aligned[1]	200	85.57 ± 0.52
Sub-SML, funneled & aligned[3]	300	86.73 ± 0.53
FV+PCA-Whitening funneled & aligned[8]	128	78.60± N/A
Fisher Vector Faces, funneled & aligned[8]	128	$\mathbf{87.47 \pm 1.49}$
FV+QCSML, aligned	256	87.10 ± 1.25
FV+QCSML, aligned	128	$\mathbf{87.47 \pm 1.99}$
FV+QCSML, aligned	64	85.20 ± 1.39
FV+QCSML, aligned	32	84.53 ± 1.74

The Receiving Operating Characteristic Equal Error Rate (ROC-EER) measure is used for evaluations. In the restricted setting, we compare the proposed QCSML method with Combined B/G sample based method[16], LDML[17], DML-eig combined method[18], LBP-CSML[1], SML[3] and Fisher Vector Face[8]. The face verification results are shown in Table 2. Compared with the compressed FV after PCA-whitening, our QCSML method improve the accuracy about 9% and it is the same performance as the Large Margin Dimensionality Reduction (LMDR) which employed metric similarity distance[8]. Besides, the proposed QCSML obtains 87.47% verification rate, which mostly outperforms other state-of-the-art method in the restricted setting.

Table 3. Comparison of QCSML method with other state-of-the-art methods in unrestricted setting of LFW.

Method	Accuracy(%)
LDML-MKNN, funneled[17]	87.50 ± 0.40
PLDA combined, funneled & aligned[19]	90.07 ± 0.51
Joint Bayesian combined[20]	90.90 ± 1.48
Sub-SML combined, funneled & aligned[3]	90.75 ± 0.64
Fisher Vector Faces, funneled & aligned[8]	$\mathbf{93.03 \pm 1.05}$
FV+QCSML, aligned	$\mathbf{92.33 \pm 1.12}$

Moreover, we evaluate the proposed QCSML method in unrestricted setting of LFW. The results of our method performance are shown in Table 3 and Figure 2. Our method achieves 92.33% accuracy, closely matching the Fisher Vector Face[8], which achieves 93.03%. According to the Table 3, it is obvious that our method obtains 92.33% verification rate and outperforms most state-of-the-art methods such as LDML-MKNN[17], PLDA[19], joint Baysian[20] and Sub-SML[3]. Although our method cannot obtain higher verification rate than

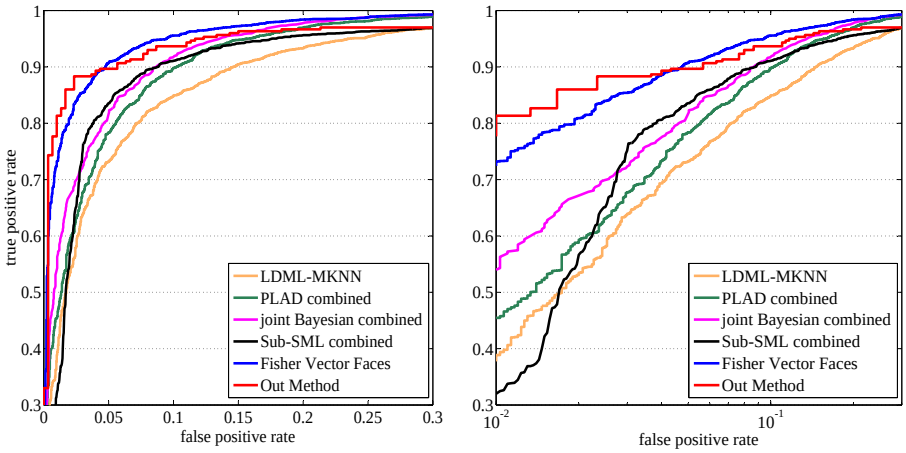

Fig. 2. ROC curves of our method and the state-of-the-art techniques in LFW-unrestricted setting. The left is shown in Linear-Axis and the right is in Log-Axis.

Fisher Vector Faces[8], we can find our method performs better at false accept rate (false positive rate) 1% point than them in Figure 2(b), which means our method has more valuable for practical systems because the threshold is often selected when the false accept rate is at 0.1% or 1% point instead of equal error rate.

5 Conclusion

In this paper, we proposed the Quasi Cosine Similarity Metric Learning (QCSML) method for classification and face verification tasks. We employ normalization and Lagrange multipliers to convert the cosine similarity metric into a new formulation and it makes the computation faster for high dimensional features and the complexity of QCSML is $O(t \times p \times d)$ which precedes CSML method. In practice, our QCSML performs considerably better on both UCI classification datasets and LFW dataset. In the future, we plan to investigate the optimization processing to make the method more effective and efficient and extend our QCSML to other applications.

Acknowledgement. This work was jointly supported by Beijing Natural Science Foundation under Grant No.4122049, Beijing Higher Education Young Elite Teacher (No.YETP0381), and the Fundamental Research Funds for the Central Universities(FRF-JX-12-002).

References

1. Nguyen, H.V., Bai, L.: Cosine similarity metric learning for face verification. In: Computer Vision–ACCV 2010. Springer (2011) 709–720
2. Guillaumin, M., Mensink, T., Verbeek, J., Schmid, C.: Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation. In: Computer Vision, 2009 IEEE 12th International Conference on, IEEE (2009) 309–316
3. Cao, Q., Ying, Y., Li, P.: Similarity metric learning for face recognition. In: Computer Vision (ICCV), 2013 IEEE International Conference on, IEEE (2013) 2408–2415
4. Xing, E.P., Ng, A.Y., Jordan, M.I., Russell, S.: Distance metric learning with application to clustering with side-information. Advances in neural information processing systems (2003) 521–528
5. 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, ACM (2007) 209–216
6. Qi, G.J., Tang, J., Zha, Z.J., Chua, T.S., Zhang, H.J.: An efficient sparse metric learning in high-dimensional space via l 1-penalized log-determinant regularization. In: Proceedings of the 26th Annual International Conference on Machine Learning, ACM (2009) 841–848
7. Friedman, J., Hastie, T., Tibshirani, R.: Sparse inverse covariance estimation with the graphical lasso. Biostatistics **9** (2008) 432–441
8. Simonyan, K., Parkhi, O.M., Vedaldi, A., Zisserman, A.: Fisher vector faces in the wild. In: Proc. BMVC. Volume 1. (2013) 7
9. Bottou, L., Bousquet, O.: The tradeoffs of large scale learning. In: NIPS. Volume 4. (2007) 2
10. Zhu, C., Byrd, R.H., Lu, P., Nocedal, J.: Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. ACM Transactions on Mathematical Software (TOMS) **23** (1997) 550–560
11. Huang, G.B., Ramesh, M., Berg, T., Learned-Miller, E.: Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, Technical Report 07-49, University of Massachusetts, Amherst (2007)
12. Weinberger, K., Blitzer, J., Saul, L.: Distance metric learning for large margin nearest neighbor classification. Advances in neural information processing systems **18** (2006) 1473
13. Perronnin, F., Dance, C.: Fisher kernels on visual vocabularies for image categorization. In: Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on, IEEE (2007) 1–8
14. Perronnin, F., Sánchez, J., Mensink, T.: Improving the fisher kernel for large-scale image classification. In: Computer Vision–ECCV 2010. Springer (2010) 143–156
15. Perronnin, F., Liu, Y., Sánchez, J., Poirier, H.: Large-scale image retrieval with compressed fisher vectors. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE (2010) 3384–3391
16. Wolf, L., Hassner, T., Taigman, Y.: Similarity scores based on background samples. In: Computer Vision–ACCV 2009. Springer (2010) 88–97
17. Guillaumin, M., Verbeek, J., Schmid, C.: Is that you? metric learning approaches for face identification. In: Computer Vision, 2009 IEEE 12th International Conference on, IEEE (2009) 498–505
18. Ying, Y., Li, P.: Distance metric learniying2012distanceng with eigenvalue optimization. The Journal of Machine Learning Research **13** (2012) 1–26

19. Li, P., Fu, Y., Mohammed, U., Elder, J.H., Prince, S.J.: Probabilistic models for inference about identity. Pattern Analysis and Machine Intelligence, IEEE Transactions on **34** (2012) 144–157
20. Chen, D., Cao, X., Wang, L., Wen, F., Sun, J.: Bayesian face revisited: A joint formulation. In: Computer Vision–ECCV 2012. Springer (2012) 566–579