Cross-Modal Metric Learning for AUC Optimization

DOI:
10.1109/TNNLS.2017.2769128

Document Version
Accepted author manuscript

Cross-Modal Metric Learning for AUC Optimization

Jing Huo, Yang Gao, *Member, IEEE,* Yinghuan Shi, and Hujun Yin, *Senior Member, IEEE*

***Abstract*—Cross-modal metric learning deals with learning distance functions for cross-modal data matching. The existing methods mostly focus on minimizing a loss defined on sample pairs. However, the numbers of intra-class and inter-class sample pairs can be highly imbalanced in many applications, and this can lead to deteriorating or unsatisfactory performances. The Area Under the ROC curve (AUC) is a more meaningful performance measure for the imbalanced distribution problem. To tackle the problem as well as to make samples from different modalities directly comparable, a cross-modal metric learning method is presented by directly maximizing AUC. The method can be further extended to focus on optimizing partial AUC (pAUC), which is the AUC between two specific false positive rates. This is particularly useful in certain applications where only the performances assessed within predefined false positive ranges are critical. The proposed method is formulated as a Log Determinant (LogDet) regularized semi-definite optimization problem. For efficient optimization, a mini-batch proximal point algorithm is developed. The algorithm is experimentally verified stable with the size of sampled pairs which form a mini-batch at each iteration. Several datasets have been used in evaluation, including three cross-modal datasets on face recognition under various scenarios and a single modal dataset, the Labeled Faces in the Wild (LFW). Results demonstrate the effectiveness of the proposed methods and marked improvements over the existing methods. Specifically, pAUC-optimized cross-modal metric learning proves to be more competitive for performance measures such as Rank-1 and VR@FPR=0.1%.**

***Index Terms*—Cross-modal metric learning, AUC optimization, positive semi-definite optimization, face recognition, multi-modal classification**

I. Introduction

With the rapid development of data acquisition in increasing number of modalities, multi-modal data is becoming a commonality in real-world applications. As a result, there is an increasing demand for cross-modal data matching techniques that can determine whether two samples of different modalities are of the same class. Applications include cross-modal face recognition (e.g. sketches to photos and near infrared to visible light images), cross-media retrieval (images to text), voices to images and so on. To solve such matching problems, cross-modal metric learning has been intensively studied.

Generally, the goals of cross-modal metric learning are of two folds. The first is to remove data heterogeneity so that data of two modalities become comparable. For this objective, a common subspace based cross-modal metric function is adopted as in this paper. The cross-modal metric function, parameterized by two projection matrices, can be interpreted as finding a common subspace to remove modality variations so that distances of different modality are measurable. Previous work that applied such metric function includes [1] and [2]. However, most of these methods have used an alternating optimization technique to optimize the two metric parameters, which leads to non-convex optimization. Differently, the metric function in our approaches is written into a more compact form parameterized by a Positive Semi-Definite (PSD) matrix, so the final optimization problem is convex.

For the second objective, pre-defined must-link (i.e. same labeled or similar) and cannot-link (i.e. differently labeled or dissimilar) cross-modal distance constraints should be satisfied in the common subspace. However, as the numbers of must-link and cannot-link sample pairs constructed on a training (or given) dataset can be highly imbalanced in many applications. For example, in large scale face recognition problem, each subject may only have a small number of samples to construct the must-link constraints. However, all the samples of all the other subjects can be used to construct the cannot-link constraints and are of very large quantity, leading to highly imbalanced distance constraints. For other applications, the cross-modal datasets can also be essentially unbalanced, such as rumor detection (text/image/video) [3], spam detection (network/text) [4], sentiment analysis (image/text) [5]. In these cases, directly minimizing the loss of predicting sample pairs can lead to the loss dominated by the loss of cannot-link constraints. The Area Under the ROC curve (AUC) is a arguably more meaningful performance measure for such imbalanced distribution problems [6]. We therefore propose to learn cross-modal metrics that directly optimize AUC on pre-defined must-link and cannot-link sample pairs. In general, for metric learning with must-link and cannot-link sample pairs given, AUC measures the probability for a randomly selected must-link sample pair to have smaller distance compared with the distance of a randomly selected cannot-link sample pair. In practice, AUC optimization is written as the sum of all the losses of separating the distances of a must-link sample pair and a cannot-link sample pair. Besides, for certain applications, partial AUC (pAUC), which is the AUC for a specific range

This work was supported by the National Science Foundation of China (Grant Nos. 61432008, 61673203), the Young Elite Scientists Sponsorship Program by CAST (YESS 20160035) and the Collaborative Innovation Center of Novel Software Technology and Industrialization. J. Huo was also supported by a scholarship from the China Scholarship Council as a one-year visiting research student at the University of Manchester during 2015-2016. *(Corresponding author: Yang Gao.)*

J. Huo, Y. Gao and Y. Shi are with the State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210046, China. (e-mail: jing.huo@manchester.ac.uk, gaoy@nju.edu.cn, syh@nju.edu.cn).

H. Yin is with the School of Electrical and Electronic Engineering, The University of Manchester, UK. (e-mail: hujun.yin@manchester.ac.uk).

of false positive rate, is of more importance than the full AUC. For example, in face recognition, it is often required that a system should have a high true positive rate (TPR) with the false positive rate (FPR) of a very small scale such as 0.1%. In this case, it is more meaningful to optimize the pAUC with a FPR range of $[0, 0.1]$. To adapt to these situations, the proposed cross-modal metric learning method is also extended for optimizing pAUC. The formulation of AUC and pAUC optimization can be written into a consistent form and optimized in the same fashion.

For the final optimization, as the metric function is parameterized by a PSD matrix, the Log Determinant (LogDet) regularization is adopted to guarantee the metric parameters to remain PSD during the optimization. Besides, as the complexity of the AUC-optimized or pAUC-optimized metric learning is proportional to the number of must-link distance constraints times the number of cannot-link distance constraints, which is quartic of the number of training samples. For efficient optimization, we adopted a combination of the proximal point algorithm [7], [8] and mini-batch stochastic optimization algorithm. Proximal point algorithm is used for LogDet optimization to guarantee the metric parameter PSD. Mini-batch stochastic optimization is to reduce the complexity of the AUC-optimization or pAUC-optimization. Specifically, during the optimization, both must-link and cannot-link sample pairs are sampled and the optimization is done on the sampled subsets, which form the mini-batches. The algorithm is experimentally proven to be stable with the size of sampled must-link and cannot-link pairs. Only under the condition when either the number of must-link or cannot-link pairs used for optimization at each iteration is extremely small (less than 10), the performance may degrade.

The rest of this paper is organized as follows. In Section II, related work is briefly reviewed. Details of the proposed methods together with optimization techniques and implementation details are given in Section III. In Section IV, experiments and results are presented together with discussions. Finally, the study is concluded in Section V.

II. Related Work

A. Single-modal Metric Learning

Metric learning is to learn metric functions that satisfy pairwise must-link/cannot-link distance/similarity constraints or triplet-based relative distance/similarity constraints [9], [10], [11], [12], [13]. It has been widely studied during the past decades. According to the metric functions used, metric learning can be categorized into two categories.

The first is the Mahalanobis metric learning. One drawback is that it is mainly for data of single modality, making it unsuitable for cross-modal data matching and inapplicable if samples of two modalities are of different dimensionality. Xing *et al.* [14] were the first to study the problem of learning Mahalanobis metric functions given pairs of must-link and cannot-link samples. Other representative work includes the Information Theoretic Metric Learning (ITML) [15], which learns metrics subject to both must-link and cannot-link distance constraints together with minimizing the divergence of the learned metric and Euclidean distance metric. Similar to ITML, the proposed method also adopts the LogDet regularization. However, we have used the proximal point method in optimization, while ITML used Bregman projection. In [16], Weinberger *et al.* proposed a Large Margin Nearest Neighbor (LMNN) method. The objective contains two parts, one is to minimize the distances of must-link pairs and the other is to subject to a set of triplet-based relative distance constraints by forcing margins between the distances of must-link and cannot-link pairs encoded in the triplets. The proposed method also uses relative distance constraints and must-link distance constraints for optimization. However, the relative distance constraints are tetrad based. A more detailed comparison of LMNN and the proposed method is given in Section III. In [17], [18], metrics were optimized for various performance measures and AUC was one of them. Differently, the AUC in their methods are defined on ranking results while ours on the must-link and cannot-link constraints. In fact, these existing AUC optimization methods are based on triplets. Besides, our method can be easily extended to optimize partial AUC, and this would not be possible in the previous methods.

The second category is the bilinear similarity metric learning. Because a bilinear similarity metric does not require PSD constraints on metric parameters, it is usually more efficient compared to Mahalanobis metric learning. Early work of this kind was also designed for single-modal data. The Online Algorithm for Scalable Image Similarity learning (OASIS) was proposed [19]. Bilinear similarity function was adopted and an online passive-aggressive based learning algorithm was developed for efficient optimization. In the work of [20], Liu *et al.* proposed to learn low-rank bilinear functions and the algorithm scaled linearly with input dimensionality in both space and time. The work of [21] addresses the problem of efficiently learning similarity measure from high-dimensional sparse data. The similarity measure was modeled as a convex combination of rank-one matrices with specific sparsity structures.

B. Cross-modal Metric Learning

Recently, there is an increase in the number of studies in learning metric functions for cross-modal problems. Again, according to the metric function used, the studies can be put into two categories. The first is the common subspace based and the second is the similarity metric based.

For the first category, a cross-modal metric learning method [1] is proposed to find common subspace in which both the pairwise must-link and cannot-link cross-modal distance constraints are satisfied. Zhou *et al.* [22] proposed to incorporate both homogeneous local information and heterogeneous constraints into a whole framework to learn a cross-modal metric, and the local information was adopted by using an extension of Locally Linear Embedding (LLE). In [23], Quadrianto and Lampert proposed to learn projections to preserve neighbourhood information for multi-view data. Wu *et al.* [2] also proposed to learn a common subspace based heterogeneous distance metric by optimizing distances of related cross-modal data and by using two regularization terms to incorporate prior information.

For the second category, a bilinear cross-modal similarity metric learning method is proposed [24] and the learned metric is optimized to be low rank. Zhen *et al.* [25] proposed a probabilistic framework for learning pairwise similarities. It was capable of learning from both pairwise and relative constraints and suitable for active learning setting.

Our proposed method is also a common subspace based. However, different from the previous methods, which adopt alternating direction methods to optimize the projection parameters, our method is convex, ensuring that the optimization does not suffer from local minimum. Besides, the existing methods optimize metric functions to satisfy either pairwise distance constraints or triplet based relative constraints, while the proposed method optimizes the AUC defined on provided must-link and cannot-link constraints, which are tetrad based.

C. AUC Optimization

AUC optimization has been extensively studied over the past decades [6], [26], [27], [28]. Most of the algorithms were designed for learning classifiers for classification problems. Joachims [26] proposed to use the structural support vector machine (svm) to optimize various performance measures (AUC was one of them). In [29], Narasimhan and Agarwal proposed to use the structural svm framework for efficient optimization of partial AUC. The AUC optimization problem in an online manner was also studied [6], [27], [28]. Kar *et al.* [30] proposed an online and stochastic learning framework for optimizing non-decomposable loss function (the loss function of pAUC was one of them).

In AUC and pAUC optimization, the main problem is the computational cost. For AUC-optimized classifier learning, the complexity is quadratic of the number of training samples. To ease the computational cost, online learning has been developed [6], [27], [28], [30] and another approach is to use cutting plane optimization [26], [29]. The computational cost of AUC-optimized metric learning is higher than classifier learning which is quartic of the training sample number. Perhaps this is the reason why little metric learning work related to AUC optimization has been done. However, as AUC is an important performance measure, it is beneficial to bring AUC optimization to metric learning. The proposed metric learning method is designed to optimize AUC or pAUC. Further more, we have developed a mini-batch proximal point algorithm for efficient optimization, which differs with [17], [18] that adopt cutting plane method.

III. Cross-modal Metric Learning for AUC and pAUC Optimization

A. Notations and Problem Statement

Suppose there are two sets of training samples in two modalities $\mathcal{X} = \{(\mathbf{x}_i, l_i^x)|i = 1, 2, ..., n_x\}$ and $\mathcal{Y} = \{(\mathbf{y}_i, l_i^y)|i = 1, 2, ..., n_y\}$. n_x and n_y are the numbers of samples of the two modalities, respectively. $\mathbf{x}_i \in \mathbb{R}^{d_x}$ is the ith sample of the first modality of dimension d_x with label l_i^x. $\mathbf{y}_i \in \mathbb{R}^{d_y}$ is the ith sample of the second modality of dimension d_y with class label l_i^y. Suppose there are totally c classes, i.e. $l_i^x \in \{1, 2, ..., c\}$ and $l_i^y \in \{1, 2, ..., c\}$. Besides, define $\mathbf{X} \in \mathbb{R}^{d_x \times n_x}$ and $\mathbf{Y} \in \mathbb{R}^{d_y \times n_y}$ the matrices of samples of the two modalities. To learn a cross-modal distance function, two sets of must-link and cannot-link sample pairs are constructed and in some situations, the two sets are pre-given. Define $\mathcal{S} = \{(i, j)|l_i^x = l_j^y\}$ the set of index of the constructed cross-modal must-link sample pairs and $\mathcal{D} = \{(i, j)|l_i^x \neq l_j^y\}$ the constructed cross-modal cannot-link index set. In this work, the aim is to learn a distance function to measure the distance between two samples of different modalities:

$$D_{\mathbf{W}_x, \mathbf{W}_y}(\mathbf{x}_i, \mathbf{y}_j) = \|\mathbf{W}_x^T \mathbf{x}_i - \mathbf{W}_y^T \mathbf{y}_j\|_2^2, \tag{1}$$

where $\|\cdot\|_2$ is the l_2 norm. $D_{\mathbf{W}_x, \mathbf{W}_y}(\mathbf{x}_i, \mathbf{y}_j)$ is the distance function and $\mathbf{W}_x \in \mathbb{R}^{d_x \times d_c}$ and $\mathbf{W}_y \in \mathbb{R}^{d_y \times d_c}$ are its parameters.

The metric function can be interpreted as projecting samples of two modalities into a common sub-space of dimensionality d_c and then distances are measured in the common subspace. However, the optimization of learning such a metric function relies on alternately optimizing $\mathbf{W}_x$ and $\mathbf{W}_y$ and the problem is non-convex. Therefore, the following transformed metric function is adopted instead,

$$\begin{aligned} D_{\mathbf{M}}(\mathbf{x}_i, \mathbf{y}_j) &= \|\mathbf{W}_x^T \mathbf{x}_i - \mathbf{W}_y^T \mathbf{y}_j\|_2^2 \\ &= [\mathbf{x}_i^T, -\mathbf{y}_j^T] \begin{bmatrix} \mathbf{W}_x \mathbf{W}_x^T & \mathbf{W}_x \mathbf{W}_y^T \\ \mathbf{W}_y \mathbf{W}_x^T & \mathbf{W}_y \mathbf{W}_y^T \end{bmatrix} \begin{bmatrix} \mathbf{x}_i \\ -\mathbf{y}_j \end{bmatrix} \\ &= \mathbf{z}_{ij}^T \mathbf{M} \mathbf{z}_{ij} = \langle \mathbf{z}_{ij} \mathbf{z}_{ij}^T, \mathbf{M} \rangle_F, \end{aligned} \tag{2}$$

where $\langle \mathbf{A}, \mathbf{B} \rangle_F = \text{tr}(\mathbf{A}^T \mathbf{B})$ is the Frobenius inner product, $\text{tr}(\cdot)$ is the trace operator. $\mathbf{z}_{ij} \in \mathbb{R}^{d_x + d_y}$ is the concatenation of two feature vectors of the form $[\mathbf{x}_i^T, -\mathbf{y}_j^T]^T$. $\mathbf{M} \in \mathbb{S}_+^{(d_x + d_y)}$ is the parameter of the newly defined metric, $\mathbb{S}_+^d$ denotes the cone of symmetric positive semi-definite (PSD) matrices of size $d \times d$.

The PSD constraint is to ensure that the distance defined in Eq. (2) is non-negative. It is obvious to see that learning parameters of $\mathbf{W}_x$ and $\mathbf{W}_y$ is equivalent to learning parameter $\mathbf{M}$.

B. Objective Functions

Most of the previous metric learning methods try to directly adjust $D_{\mathbf{M}}(\cdot, \cdot)$ by minimizing loss functions defined on training pairs, so that the output of $D_{\mathbf{M}}(\cdot, \cdot)$ for must-link pair is small and the output for cannot-link pair is large. However, the numbers of pairs in the must-link and cannot-link sets are usually imbalanced, making the losses defined on must-link and cannot-link sets also imbalanced. AUC is a more efficient performance measure for this situation. AUC on the defined pairwise constraints is therefore optimized. For a specific metric function, its AUC equals to the proportion of the number of correctly ranked distances of pairs. Simply denote D as $D_{\mathbf{M}}$. Formally, the AUC value of D on the defined pair sets $\mathcal{S}$ and $\mathcal{D}$ is as follows:

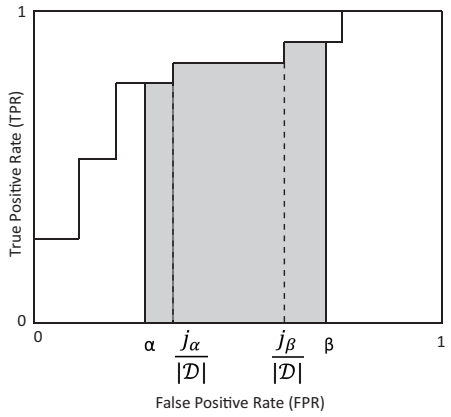


Fig. 1. Illustration of partial AUC between FPRs α and β.

$$\text{AUC}_D = \frac{1}{|\mathcal{S}||\mathcal{D}|}\sum_{(i,j)\in\mathcal{S}}\sum_{(k,l)\in\mathcal{D}}\big(\mathbb{I}[D(\mathbf{x}_i,\mathbf{y}_j) < D(\mathbf{x}_k,\mathbf{y}_l)] + \frac{1}{2}\mathbb{I}[D(\mathbf{x}_i,\mathbf{y}_j) = D(\mathbf{x}_k,\mathbf{y}_l)]\big), \tag{3}$$

where $\mathbb{I}[\cdot]$ is the indicator function which returns 1 if the statement is true and 0 otherwise. $|\mathcal{S}|$ and $|\mathcal{D}|$ are the numbers of index pairs in set $\mathcal{S}$ and $\mathcal{D}$ respectively.

Besides AUC, partial AUC is also an important performance measure for applications that require high true positive rates and low false positive rates. Partial AUC is the AUC in a specific false positive rate (FPR) range. An illustration of partial AUC is given in Fig. 1. The partial AUC for a metric function D on the defined constraints in a false positive rate (FPR) range of $[\alpha, \beta]$ is defined as:

$$\text{pAUC}_D(\alpha,\beta) = \frac{1}{|\mathcal{S}||\mathcal{D}_{j_\alpha,j_\beta}|}\sum_{(i,j)\in\mathcal{S}}\sum_{(k,l)\in\mathcal{D}_{j_\alpha,j_\beta}}\big(\mathbb{I}[D(\mathbf{x}_i,\mathbf{y}_j) < D(\mathbf{x}_k,\mathbf{y}_l)] + \frac{1}{2}\mathbb{I}[D(\mathbf{x}_i,\mathbf{y}_j) = D(\mathbf{x}_k,\mathbf{y}_l)]\big), \tag{4}$$

where $j_\alpha = \lceil|\mathcal{D}|\alpha\rceil + 1$ and $j_\beta = \lfloor|\mathcal{D}|\beta\rfloor$ are two integers[1]. Note that we assume $|\mathcal{D}| \geq \frac{1}{\beta-\alpha}$, so that $j_\alpha \leq j_\beta$. $\mathcal{D}_{j_\alpha,j_\beta}$ is a subset of $\mathcal{D}$. By firstly sorting all the distances of the cannot-link pairs in set $\mathcal{D}$ in an ascending order using the distance function D, then $\mathcal{D}_{j_\alpha,j_\beta}$ are the indices of those pairs that are ranked within the range of j_αth position to j_βth position.

Given the definition of pAUC, the full AUC can be seen as a special case by setting $\alpha = 0$ and $\beta = 1$. Therefore, in the following, only the optimization of the partial AUC is discussed. Directly optimize the formulation in Eq. (4) can lead to an NP-hard problem. We therefore try to optimize an approximated formulation by replacing the indicator function with a convex loss function (hinge loss in this case). The maximization of the correctly ranked pairs is equivalent to the minimization of incorrectly ranked pairs. Then by replacing the indicator function with the loss function resulted in the final objective function [6], the objective of maximizing AUC or partial AUC is thus changed to minimizing the following function:

$$L(\mathbf{M}) = \sum_{(i,j)\in\mathcal{S}}\sum_{(k,l)\in\mathcal{D}_{j_\alpha,j_\beta}}\frac{[1 + D(\mathbf{x}_i,\mathbf{y}_j) - D(\mathbf{x}_k,\mathbf{y}_l)]_+}{|\mathcal{S}||\mathcal{D}_{j_\alpha,j_\beta}|}, \tag{5}$$

where $[a]_+ = \max(0, a)$.

Directly minimizing Eq. (5) can lead to both the distances of must-link pairs and the distances of cannot-link pairs turning unbounded. Therefore regularization terms are needed to avoid such situations. In this paper, the following two regularization terms are used. The first is to force the distances of the must-link cross-modal pair remain small. The second is the Log Determinant (LogDet) divergence function. The LogDet divergence of two metric parameters $\mathbf{M}$ and $\mathbf{M}_0$ is defined as $LD_g(\mathbf{M}||\mathbf{M}_0) = g(\mathbf{M}) - g(\mathbf{M}_0) - \langle\nabla g(\mathbf{M}_0), \mathbf{M} - \mathbf{M}_0\rangle$ with $g(\mathbf{M}) = -\text{logdet}(\mathbf{M})$. By removing the constant term, it is equivalent to $LD(\mathbf{M}||\mathbf{M}_0) = \text{tr}(\mathbf{M}_0^{-1}\mathbf{M}) - \text{logdet}(\mathbf{M})$. In this paper, $\mathbf{M}_0$ is set as the identity matrix. The use of such regularization has several nice properties. The first is that it implicitly guarantees the metric parameter remains PSD during optimization. Besides, by setting $\mathbf{M}_0$ as the identity matrix, it regularizes $\text{tr}(\mathbf{M}) = \|\mathbf{W}_x\|_F^2 + \|\mathbf{W}_y\|_F^2$ which implicitly controls the magnitude of the two projection matrices to avoid overfitting. The last is that as $\mathbf{M}$ is PSD, the nuclear norm of $\mathbf{M}$ is equal to the trace of the matrix. Thus $\text{tr}(\mathbf{M})$ can promote low-rankness of matrix $\mathbf{M}$ during optimization. The final objective of the proposed method is to minimize the following function:

$$F(\mathbf{M}) = \sum_{(i,j)\in\mathcal{S}}\sum_{(k,l)\in\mathcal{D}_{j_\alpha,j_\beta}}\frac{[1 + D(\mathbf{x}_i,\mathbf{y}_j) - D(\mathbf{x}_k,\mathbf{y}_l)]_+}{|\mathcal{S}||\mathcal{D}_{j_\alpha,j_\beta}|} + \gamma\sum_{(i,j)\in\mathcal{S}}\frac{D(\mathbf{x}_i,\mathbf{y}_j)}{|\mathcal{S}|} + \mu\big[\text{tr}(\mathbf{M}_0^{-1}\mathbf{M}) - \text{logdet}(\mathbf{M})\big], \tag{6}$$

where $\mathbf{M} \in \mathbb{S}_+^{(d_x+d_y)}$ and $\mathbf{M}_0$ is set as the identity matrix. γ and μ are two regularization parameters which need to tune. The first term in Eq. (6) relates to AUC or pAUC maximization. The second term controls the distances of the must-link pairs staying small. The last term is the LogDet regularization term.

With the formulation of the proposed method at hand, we give a brief discussion of the differences of the proposed method and LMNN. Both of them use relative distance constraints (distance of a cannot-link pair be greater than that of a must-link pair) and pairwise must-link constraints for optimization. However, the key differences are of two folds. 1) The relative distance constraints used in this paper are tetrad based. While in LMNN, the relative distance constraints are

[1]Note that the formulation of pAUC used in this paper does not take the AUC between FPRs $[\alpha, \frac{j_\alpha}{|\mathcal{D}|}]$ and $[\frac{j_\beta}{|\mathcal{D}|}, \beta]$ into consideration, as can be seen in Fig. 1. It is a very small potion for large scale problem. By using this formulation, the definition of pAUC is consistent with that of full AUC and can be optimized in a compact form.

triplet based. The proposed method is thus more general. One drawback of LMNN using triplets is that, as the construction of triplets relies on data labels, it is not applicable for applications where only must-link and cannot-link pairs are given. 2) LMNN can partly be seen as a special case of our pAUC formulation under the following setting. Suppose LMNN selects one same class nearest neighbor $\mathbf{y}$ for a sample $\mathbf{x}$ to construct a must-link pair $(\mathbf{x}, \mathbf{y})$. Also suppose that for $\mathbf{x}$, there are n different classes samples. LMNN then selects k different classes nearest neighbors from all the n different classes samples to construct k cannot-link pairs. The objective of LMNN is to force the k cannot-link pairs have larger distances than the must-link pair. This process is equivalent to optimizing the pAUC of the must-link pair and the n cannot-link pairs with the FPR range of $[0, \frac{k}{n}]$.

C. Optimization

Being introduced in the previous section, the objective of the proposed AUC and pAUC optimization is to solve the following optimization problem.

$$\begin{aligned} &\min_{\mathbf{M}} F(\mathbf{M}) \\ &\text{s.t. } \mathbf{M} \in \mathbb{S}_+^{(d_x+d_y)}. \end{aligned} \tag{7}$$

Since there is a $\log\det(\cdot)$ term in function $F(\mathbf{M})$, the proximal point algorithm (PPA) together with mini-batch stochastic optimization is adopted for efficient optimization. Before introducing the optimization algorithm, we first transform the objective function to a more compact form without summation terms. The mini-batch PPA optimization problem is then introduced, followed with implementation details.

1) Transform the Objective Function: We first transform the first term in Eq. (6). Define an index matrix $\mathbf{J} \in \{0,1\}^{|\mathcal{S}|\times|\mathcal{D}_{j_\alpha,j_\beta}|}$. The elements of $\mathbf{J}$ are defined as:

$$\mathbf{J}_{p(i,j),q(k,l)} = \begin{cases} 1 & 1 + D_{i,j} > D_{k,l} \\ 0 & 1 + D_{i,j} \le D_{k,l} \end{cases}, \tag{8}$$

where $(i,j) \in \mathcal{S}$ and $(k,l) \in \mathcal{D}_{j_\alpha,j_\beta}$. $D_{i,j}$ and $D_{k,l}$ are short for $D_{\mathbf{M}}(\mathbf{x}_i, \mathbf{y}_j)$ and $D_{\mathbf{M}}(\mathbf{x}_k, \mathbf{y}_l)$ respectively. $p(i,j)$ is defined as the index of pair $(\mathbf{x}_i, \mathbf{y}_j)$ in set $\mathcal{S}$ and $q(k,l)$ the index of pair $(\mathbf{x}_k, \mathbf{y}_l)$ in set $\mathcal{D}_{j_\alpha,j_\beta}$. Thus $\mathbf{J}_{p(i,j),q(k,l)}$ is the element at the $p(i,j)$th row and $q(k,l)$th column of matrix $\mathbf{J}$. It indicates whether the distance of the $p(i,j)$th must-link pair plus one is correctly ranked before the distance of the $q(k,l)$th cannot-link pair using the distance function $D_{\mathbf{M}}$, if it is incorrectly ranked, the value is 1. Define n_j the number of non-zero elements in $\mathbf{J}$. Then the following transformation holds:

$$\begin{aligned} &\sum_{(i,j)\in\mathcal{S}} \sum_{(k,l)\in\mathcal{D}_{j_\alpha,j_\beta}} \frac{[1 + D(\mathbf{x}_i, \mathbf{y}_j) - D(\mathbf{x}_k, \mathbf{y}_l)]_+}{|\mathcal{S}||\mathcal{D}_{j_\alpha,j_\beta}|} \\ &= \langle \sum_{(i,j)\in\mathcal{S}} \sum_{(k,l)\in\mathcal{D}_{j_\alpha,j_\beta}} \frac{\mathbf{J}_{p(i,j),q(k,l)}(\mathbf{z}_{ij}\mathbf{z}_{ij}^T - \mathbf{z}_{kl}\mathbf{z}_{kl}^T)}{|\mathcal{S}||\mathcal{D}_{j_\alpha,j_\beta}|}, \mathbf{M}\rangle_F \\ &+ \frac{n_j}{|\mathcal{S}||\mathcal{D}_{j_\alpha,j_\beta}|} \\ &= \langle \mathbf{P}_A, \mathbf{M}\rangle_F + \frac{n_j}{|\mathcal{S}||\mathcal{D}_{j_\alpha,j_\beta}|}, \end{aligned} \tag{9}$$

where $\mathbf{P}_A$ is defined as:

$$\mathbf{P}_A = \sum_{(i,j)\in\mathcal{S}} \sum_{(k,l)\in\mathcal{D}_{j_\alpha,j_\beta}} \frac{\mathbf{J}_{p(i,j),q(k,l)}(\mathbf{z}_{ij}\mathbf{z}_{ij}^T - \mathbf{z}_{kl}\mathbf{z}_{kl}^T)}{|\mathcal{S}||\mathcal{D}_{j_\alpha,j_\beta}|}. \tag{10}$$

From the above definition, it is apparent that during optimization, for different $\mathbf{M}_t$ at different time step, the index matrix $\mathbf{J}$ changes and thus $\mathbf{P}_A$ also changes. Calculate $\mathbf{P}_A$ on the full set of $\mathcal{S}$ and $\mathcal{D}_{j_\alpha,j_\beta}$ is computationally intensive if both the two sets are large. Stochastic optimization [31] has been proved to be an efficient optimization method for many applications. For our optimization problem, as both must-link pairs and cannot-link pairs are needed for optimization, we perform sampling in sets $\mathcal{S}$ and $\mathcal{D}$ to get two subsets $\hat{\mathcal{S}}^t$ and $\hat{\mathcal{D}}^t$ at the tth time step. Suppose uniform sampling is adopted, the AUC or pAUC defined on sets $\mathcal{S}$ and $\mathcal{D}$ are assumed to be consistent with that on $\hat{\mathcal{S}}^t$ and $\hat{\mathcal{D}}^t$. At the tth time step, $\mathbf{P}_A^t$ which is defined on $\hat{\mathcal{S}}^t$ and $\hat{\mathcal{D}}^t$ is calculated and thus becomes more efficient.

The second term in Eq. (6) is equivalent to the following:

$$\begin{aligned} \sum_{(i,j)\in\mathcal{S}} \frac{D(\mathbf{x}_i, \mathbf{y}_j)}{|\mathcal{S}|} &= \frac{1}{|\mathcal{S}|} \sum_{(i,j)\in\mathcal{S}} \langle \mathbf{z}_{ij}\mathbf{z}_{ij}^T, \mathbf{M}\rangle_F \\ &= \frac{1}{|\mathcal{S}|} \langle \sum_{(i,j)\in\mathcal{S}} \mathbf{z}_{ij}\mathbf{z}_{ij}^T, \mathbf{M}\rangle_F = \langle \mathbf{P}_S, \mathbf{M}\rangle_F, \end{aligned} \tag{11}$$

where $\mathbf{P}_S = \frac{1}{|\mathcal{S}|}\sum_{(i,j)\in\mathcal{S}} \mathbf{z}_{ij}\mathbf{z}_{ij}^T$. Differing with $\mathbf{P}_A$, $\mathbf{P}_S$ does not change during optimization and can be pre-calculated.

Therefore the final optimization problem at the tth time step is equivalent to:

$$\begin{aligned} &\min_{\mathbf{M}} \langle \mathbf{P}_A^t, \mathbf{M}\rangle_F + \gamma\langle \mathbf{P}_S, \mathbf{M}\rangle_F + \mu\big[\langle I, \mathbf{M}\rangle_F - \log\det(\mathbf{M})\big] \\ &\text{s.t. } \mathbf{M} \in \mathbb{S}_+^{(d_x+d_y)}, \end{aligned} \tag{12}$$

where I is the identity matrix. Irrelevant terms are removed from the final objective function.

2) Proximal Point Algorithm: For minimizing the objective function with a current solution $\mathbf{M}_t$ at the tth step, the Proximal Point Algorithm (PPA) generates the next solution $\mathbf{M}_{t+1}$ by (approximately) solving a perturbed problem of the following form:

$$\mathbf{M}_{t+1} = \arg\min_{\mathbf{M}\in\mathbb{S}_+^{(d_x+d_y)}} F_t(\mathbf{M}) + \frac{1}{2\eta_t}\|\mathbf{M} - \mathbf{M}_t\|_F^2, \tag{13}$$

where η_t is a parameter and $F_t(\mathbf{M}) = \langle \mathbf{P}_A^t, \mathbf{M}\rangle_F + \gamma\langle \mathbf{P}_S, \mathbf{M}\rangle_F + \mu\big[\langle I, \mathbf{M}\rangle_F - \log\det(\mathbf{M})\big]$. The solution to the optimization problem in Eq. (13) is:

$$\mathbf{M}_{t+1} = \phi_{\lambda_t}^+\big(\mathbf{M}_t - \eta_t(\mathbf{P}_A^t + \gamma\mathbf{P}_S + \mu I)\big), \tag{14}$$

where $\lambda_t = \eta_t\mu$. The proof of this solution is given in Lemma 3. And the whole algorithm is summarized in Algorithm 1.

To proof Lemma 3, a few preliminary information is first presented.

Lemma 1. *Given a symmetry matrix $\mathbf{X} \in \mathbb{S}^n$ of size $n \times n$ with its eigenvalue decomposition $\mathbf{X} = \mathbf{U}\mathbf{V}\mathbf{U}^T$, where $\mathbf{V} = \mathrm{diag}([v_1, v_2, ..., v_n])$ and assume that $v_1 \geq v_2 \geq ... \geq v_n$. Let $\lambda > 0$ be given. For the two following scalar functions $\phi_\lambda^+(x) := \left(\sqrt{x^2 + 4\lambda} + x\right)/2$ and $\phi_\lambda^-(x) := \left(\sqrt{x^2 + 4\lambda} - x\right)/2$ for all $x \in \mathbb{R}$. We define their matrix counterparts as:*

$$\begin{aligned}\mathbf{X}_1 &= \phi_\lambda^+(\mathbf{X}) := \mathbf{U}\mathrm{diag}([\phi_\lambda^+(v_1), ..., \phi_\lambda^+(v_n)])\mathbf{U}^T \\ \mathbf{X}_2 &= \phi_\lambda^-(\mathbf{X}) := \mathbf{U}\mathrm{diag}([\phi_\lambda^-(v_1), ..., \phi_\lambda^-(v_n)])\mathbf{U}^T, \mathbf{X} \in \mathbb{S}^n.\end{aligned} \tag{15}$$

Then the following holds $\mathbf{X} = \mathbf{X}_1 - \mathbf{X}_2$. Both $\mathbf{X}_1 \in \mathbb{S}_{++}^n$ and $\mathbf{X}_2 \in \mathbb{S}_{++}^n$ are positive definite and $\mathbf{X}_1\mathbf{X}_2 = \lambda I$.

Proof. The proof can be easily obtained from the definition of ϕ_λ^+ and ϕ_λ^-. □

Lemma 2. *Given $\mathbf{X} \in \mathbb{S}^n$ and $\eta > 0$, then the following holds:*

$$\begin{aligned}&\min_{\mathbf{Y} \in \mathbb{S}_{++}^n} \left\{ \frac{1}{2\eta}\|\mathbf{Y} - \mathbf{X}\|_F^2 - \mu \mathrm{logdet}(\mathbf{Y}) \right\} \\ &= \frac{1}{2\eta}\|\phi_\lambda^-(\mathbf{X})\|_F^2 - \mu \mathrm{logdet}(\phi_\lambda^+(\mathbf{X})),\end{aligned} \tag{16}$$

where $\lambda = \eta\mu$.

Proof. The objective function in Eq. (16) is strictly convex and continuously differentiable. Thus its optimal solution $\mathbf{Y}^*$ is unique and if exists it must satisfy the equation:

$$\mathbf{X} = \mathbf{Y}^* - \lambda(\mathbf{Y}^*)^{-1} \tag{17}$$

By Lemma 1, $\mathbf{Y}^* = \phi_\lambda^+(\mathbf{X})$ is a solution to Eq. (17). By plugging $\mathbf{Y}^* = \phi_\lambda^+(\mathbf{X})$ back into the objective function, one gets the lemma. □

Lemma 3. *The solution to the optimization problem in Eq. (13) is:*

$$\mathbf{M}_{t+1} = \phi_{\lambda_t}^+\big(\mathbf{M}_t - \eta_t(\mathbf{P}_A^t + \gamma\mathbf{P}_S + \mu I)\big), \tag{18}$$

where $\lambda_t = \eta_t\mu$.

Proof. As the optimization problem in Eq. (13) is equivalent to the following:

$$\min_{\mathbf{M} \in \mathbb{S}_+^{(d_x+d_y)}} \frac{1}{2\eta_t}\|\mathbf{M} - \mathbf{M}_t + \eta_t(\mathbf{P}_A^t + \gamma\mathbf{P}_S + \mu I)\|_F^2 - \mu \mathrm{logdet}(\mathbf{M}), \tag{19}$$

By Lemma 2, the solution to the above optimization problem is $\mathbf{M}^* = \phi_{\lambda_t}^+\big(\mathbf{M}_t - \eta_t(\mathbf{P}_A^t + \gamma\mathbf{P}_S + \mu I)\big)$, with $\lambda_t = \eta_t\mu$. □

3) Implementation Details: To further improve the efficiency of updating $\mathbf{P}_A^t$, the following techniques are used. If mini-batch sampling is not adopted, then the update of $\mathbf{P}_A^t$ can be done in the following form:

$$\begin{aligned}\mathbf{P}_A^t &= \sum_{(i,j)\in\mathcal{S}} \sum_{(k,l)\in\mathcal{D}_{j_\alpha,j_\beta}^t} \frac{\mathbf{J}_{p(i,j),q(k,l)}^t(\mathbf{z}_{ij}\mathbf{z}_{ij}^T - \mathbf{z}_{kl}\mathbf{z}_{kl}^T)}{|\mathcal{S}||\mathcal{D}_{j_\alpha,j_\beta}|} \\ &= \sum_{(i,j)\in\mathcal{S}} \psi_t(\mathbf{z}_{ij}).\end{aligned} \tag{20}$$

We use $\psi_t(\mathbf{z}_{ij})$ to denote the feature map of a must-link pair $(\mathbf{x}_i, \mathbf{y}_j)$ at the time step of t. As $\psi_t(\mathbf{z}_{ij})$ can be decomposed into the following form:

$$\psi_t(\mathbf{z}_{ij}) = \begin{bmatrix} \mathbf{X}\mathbf{A}_{ij}\mathbf{X}^T & \mathbf{X}\mathbf{B}_{ij}\mathbf{Y}^T \\ \mathbf{Y}\mathbf{B}_{ij}^T\mathbf{X}^T & \mathbf{Y}\mathbf{C}_{ij}\mathbf{Y}^T \end{bmatrix}, \tag{21}$$

where $\mathbf{A}_{ij} \in \mathbb{R}^{n_x \times n_x}$, $\mathbf{B}_{ij} \in \mathbb{R}^{n_x \times n_y}$ and $\mathbf{C}_{ij} \in \mathbb{R}^{n_y \times n_y}$ are three matrices of the following form:

$$\begin{aligned}\mathbf{A}_{ij} &= \sum_{(k,l)\in\mathcal{D}_{j_\alpha,j_\beta}^t} \frac{\mathbf{J}_{p(i,j),q(k,l)}^t(e_i e_i^T - e_k e_k^T)}{|\mathcal{S}||\mathcal{D}_{j_\alpha,j_\beta}^t|} \\ \mathbf{B}_{ij} &= \sum_{(k,l)\in\mathcal{D}_{j_\alpha,j_\beta}^t} \frac{\mathbf{J}_{p(i,j),q(k,l)}^t(-e_i \bar{e}_j^T + e_k \bar{e}_l^T)}{|\mathcal{S}||\mathcal{D}_{j_\alpha,j_\beta}^t|} \\ \mathbf{C}_{ij} &= \sum_{(k,l)\in\mathcal{D}_{j_\alpha,j_\beta}^t} \frac{\mathbf{J}_{p(i,j),q(k,l)}^t(\bar{e}_j \bar{e}_j^T - \bar{e}_l \bar{e}_l^T)}{|\mathcal{S}||\mathcal{D}_{j_\alpha,j_\beta}^t|}\end{aligned} \tag{22}$$

where $e_i \in \mathbb{R}^{n_x}$ is the standard basis vector with the ith element equal to 1. $\bar{e}_i \in \mathbb{R}^{n_y}$ is also the standard basis vector with the ith element equal to 1. By using the above decomposition and firstly computing all the $\mathbf{A}_{ij}$, $\mathbf{B}_{ij}$ and $\mathbf{C}_{ij}$, the matrix multiplication number in Eq. (20) can be reduced from $|\mathcal{S}||\mathcal{D}_{j_\alpha,j_\beta}|$ to 6 plus a matrix transpose operation.

The calculation of the three matrices can be done efficiently in the following way. Take the calculation of $\mathbf{A}_{ij}$ as an example. Initialize $\mathbf{A}_{ij}$ as an all zero matrix. By firstly calculate all the distances of pairs in set $\mathcal{D}_{j_\alpha,j_\beta}$ and then compare those distances with $1 + D_{ij}$. The ith diagonal element of $\mathbf{A}_{ij}$ is set to the number of cannot-link pairs that have smaller distances than $1 + D_{ij}$ divide $|\mathcal{S}||\mathcal{D}_{j_\alpha,j_\beta}|$. For the cannot-link pair $(\mathbf{x}_k, \mathbf{y}_l)$ that triggers the hinge loss, the kth diagonal element of $\mathbf{A}_{ij}$ subtract a number of one divided by $|\mathcal{S}||\mathcal{D}_{j_\alpha,j_\beta}|$.

Therefore, the complexity of updating $\mathbf{P}_A^t$ mainly consists of the following four parts. 1) Compute the distances of pairs in $\mathcal{D}_{j_\alpha,j_\beta}^t$ and $\mathcal{S}$ which is of $O(|\mathcal{S}| + |\mathcal{D}_{j_\alpha,j_\beta}^t|)$. 2) Compare the distances of must-link pairs plus 1 with the distances of cannot-link pairs where the compare operation is of $O(|\mathcal{S}| \times |\mathcal{D}_{j_\alpha,j_\beta}^t|)$. 3) Set the values of all the $\mathbf{A}_{ij}$, $\mathbf{B}_{ij}$ and $\mathbf{C}_{ij}$ which is of complexity $O(|\mathcal{S}| \times |\mathcal{D}_{j_\alpha,j_\beta}^t|)$. 4) 6 matrix multiplications. Therefore by combining the above $\mathbf{P}_A^t$ update procedure with mini-batch sampling described in the previous section, the complexity of all the four steps can be reduced. Specifically, as the sampling is done on pairs, not all samples in sets $\mathcal{X}$ and $\mathcal{Y}$ are used. Therefore the complexity of the 6 matrix multiplications is also reduced.

An illustration of the full optimization algorithm is given in Algorithm 1, where the initialization of $\mathbf{M}_0$ is done by first performing principal component analysis (PCA) on data of two modality to get to projection matrices $\mathbf{W}_{x0}$ and $\mathbf{W}_{y0}$. Then $\mathbf{M}_0$ is set to $\begin{bmatrix} \mathbf{W}_{x0}\mathbf{W}_{x0}^T & \mathbf{W}_{x0}\mathbf{W}_{y0}^T \\ \mathbf{W}_{y0}\mathbf{W}_{x0}^T & \mathbf{W}_{y0}\mathbf{W}_{y0}^T \end{bmatrix}$. η_t is updated in a form of $\eta_{t+1} = \min(\eta_t \times \rho, \tau)$, where τ is the maximum value of η_t to set and ρ is set to a value slightly greater than 1.

Algorithm 1 Cross-modal Metric Learning for AUC/pAUC Optimization by Mini-batch Proximal Point Algorithm

Input:

Two sets of constructed (or pre-given) indices $\mathcal{S}$ and $\mathcal{D}$ and training sample sets $\mathcal{X}$ and $\mathcal{Y}$

Regularization parameters $\gamma > 0$ and $\mu > 0$

Parameter $\eta_0 > 0$

Two false positive rates α and β

Sampling ratio s_1 on set $\mathcal{S}$ and ratio s_2 on set $\mathcal{D}$

Precalculate $\mathbf{P}_S$

Initialize $\mathbf{M}_0$

Output:

Metric parameter $\mathbf{M}_t \in \mathbb{S}_+^{(d_x+d_y)}$

Initialize $t = 0$

while not converged **do**

Perform uniform sampling according to sampling ratios s_1 and s_2 on sets $\mathcal{S}$ and $\mathcal{D}$ to get $\hat{\mathcal{S}}^t$ and $\hat{\mathcal{D}}^t$, compute $\hat{\mathcal{D}}^t_{j_\alpha, j_\beta}$

Calculate $\mathbf{P}_A^t$ in Eq. (20) on sets $\hat{\mathcal{S}}^t$ and $\hat{\mathcal{D}}^t_{j_\alpha, j_\beta}$

Update $\mathbf{M}_{t+1} = \phi^+_{\lambda_t}\left(\mathbf{M}_t - \eta_t(\mathbf{P}_A^t + \gamma\mathbf{P}_S + \mu I)\right)$, where $\lambda_t = \eta_t\mu$

Update η_{t+1}

$t = t + 1$

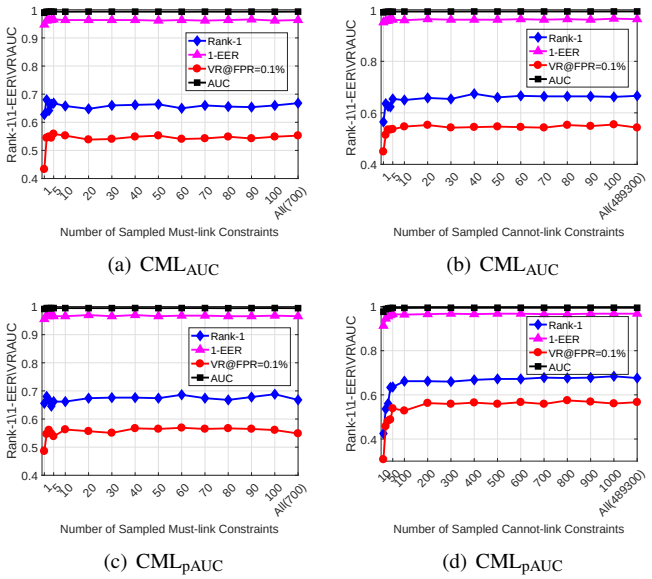
end while

IV. Experimental Results

The proposed method has been evaluated on several datasets for cross-modal matching. The first three are cross-modal face datasets including matching sketches to photos, matching near-infrared (NIR) to visible light (VIS) face images and matching low resolution to high resolution face images. As the proposed method is also applicable for single modality, the Labeled Faces in the Wild (LFW) [32], [33] face image dataset was also adopted in testing.

A. Dataset Information and Evaluation Protocols

CUHK Face Sketch FERET Dataset (CUFSF) [34]: The CUFSF dataset was used for photo to sketch face recognition. It includes 1,194 persons from the FERET dataset [35]. For each person, there are one photo and one sketch drawn by an artist after viewing the photo. To evaluate on this dataset, we randomly split the dataset into two parts for ten times and each time the first part was used for training and the other part for testing. The first part contained images of 700 persons and the second part contained images of 494 persons. For face feature extraction, after face alignment and image filtering, a face region of 160×160 was cropped out. The 160×160 face images were also down sampled to size 96×96 and 32×32. Then SIFT features [36] were extracted on the images of three scales in a fixed cell size of 32×32 with a separation of 16 pixels. All the extracted features were concatenated to serve as the final features of a face image. As this feature vector was of very high dimension, 13,696 ($128 \times 81 + 128 \times 25 + 128$), the extracted face features were reduced by PCA before applying metric learning. Results with various reduced dimensions were presented.



(a) CML_{AUC} (b) CML_{AUC}

(c) CML_{pAUC} (d) CML_{pAUC}

Fig. 2. Results of CML_{AUC} and CML_{pAUC} with different number of sampled must-link and cannot-link constraints on CUFSF dataset.

CASIA NIR-VIS 2.0 Dataset [37]: The CASIA NIR-VIS 2.0 dataset was used for evaluating the VIS-NIR face recognition. It contains 725 subjects. We followed the same evaluation protocol on this dataset by [37]. The dataset was divided into two views. View 1 was used for parameter tuning and view 2 for testing. On this dataset, the feature extraction strategy was the same as that on CUFSF dataset.

NJU-ID Dataset[2]: NJU-ID dataset contains images of 256 persons. For each person, there are one card image and fifty images collected from a high resolution camera. The ID card image is of resolution 102×126 and the camera image is of resolution 640×480. To evaluate on this dataset, we randomly selected one of the fifty images for each person and then divided the dataset into 10 folds according to identity information. The 10 folds were fully independent without overlap. The same feature extraction method on CUFSF dataset was also adopted.

Labeled Faces in the Wild (LFW) Dataset: LFW dataset contains 13,233 face images of 5,749 people. We followed the restricted setting of this dataset for testing. For comparison with related methods on this dataset, we adopted the features provided in [38] for experiments. They were SIFT features extracted at 9 fixed points of three scales. Before testing our methods, the features were also reduced to a lower dimension using PCA.

Besides the SIFT features, on all four datasets, VGG-Face [39] was also used for extracting deep features for comparison. Face alignment method was adjusted to be consistent with the one used in [39]. The resulting face feature was of 4,096 dimensions. PCA was also applied before testing.

On the three cross-modal datasets, we used evaluation protocols of AUC, 1-EER, VR@FPR=0.1% and Rank-1. EER denotes Equal Error Rate, which is the rate where false positive rate equals to false negative rate. 1-EER can be obtained

[2]Available from: http://cs.nju.edu.cn/rl/Data.html

by plotting a straight line to connect $(0,1)$ and $(1,0)$, the true positive rate of the intersection point of this line with ROC curve equals to 1-EER. It measures the degree of the ROC curve approaching the point (0,1) with false positive rate 0 and true positive rate 1. The third performance measure, VR@FPR=0.1%, is the verification rate at FPR=0.1%. It equals to the true positive rate at FPR=0.1%. The first three performance measures are directly related to AUC optimization. Rank-1 recognition rate is also provided as it is a commonly used performance measure in face recognition. On the LFW dataset, to compare with the state-of-the-art methods, accuracy was adopted as required in [32].

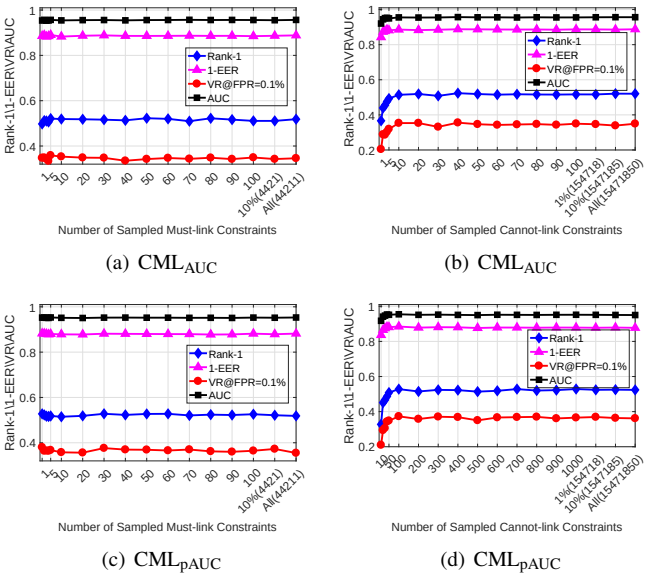
B. Parameter Settings

The proposed cross-modal metric learning for AUC optimization (denoted as CML_{AUC}) and pAUC optimization with FPRs of range $[0, 0.1]$ (CML_{pAUC}) were tested on all the datasets. The gradient step related parameter η_0 was set to 0.05, ρ to 1.01 and τ to 0.5 on the three cross-modal datasets. The setting of η_0 was 0.05, ρ 1.01 and τ 0.3 on the LFW.

γ was firstly tuned by fixing μ to a small value 10^{-7} and selected from $[0.1, 0.5, 1, 1.5, 2, 3, 4, 5]$. Then γ was fixed and μ was selected from $[10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}]$. While tuning these parameters, the sampling ratio s_1 on set $\mathcal{S}$ and sampling ratio s_2 on set $\mathcal{D}$ were both set to 1 on CUFSF, NJU-ID and LFW. On CASIA, s_1 was set to 0.1 and s_2 to 0.01.

To verify the influence of the numbers of sampled must-link and cannot-link constraints, we varied these numbers at each iteration on the CUFSF and CASIA NIR-VIS 2.0 datasets. Results on the CUFSF dataset are given in Fig. 2. The data dimension was set to 400; the influence of data dimension will be illustrated later. For AUC, the numbers of sampled must-link and cannot-link constraints were respectively varied among $[1, 2, ..., 5, 10, 20, ..., 100, 700]$ and $[1, 2, ..., 5, 10, 20, ..., 100, 489300]$. When testing the number of must-link constraints, the number of cannot-link constraints was set to 20. The number of must-link constraints was set to 10 when testing the number of cannot-link constraints. As can be seen, if the sampled number was larger than 10, the results were quite stable for different numbers of both sampled must-link and cannot-link constraints. For pAUC optimization, the number of sampled must-link was the same with AUC optimization. The number of sampled cannot-link constraints was varied among $[10, 20, ..., 50, 100, 200, ..., 1000, 489300]$. As for pAUC with FPRs of $[0, 0.1]$, at each iteration, only the top 0-0.1 cannot-link constraints were used for training. Therefore while sampling 10 cannot-link constraints, only the top 1 was used. Similar results have been observed for varying the number of sampled must-link constraints. But for number of sampled cannot-link constraints, the results were stable if the number of sampled cannot-link constraints was greater than 100 where the number of actually used constraints was still 10.

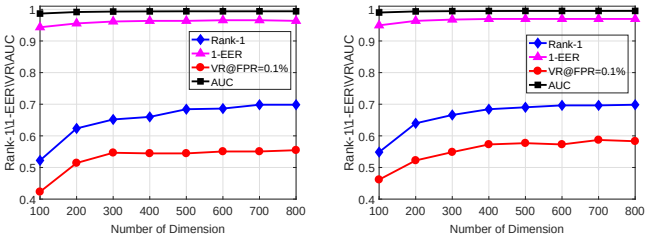
Since the CUFSF dataset is relatively small, we have also tested on CASIA NIR-VIS 2.0 dataset. As can be seen in Fig. 3, the results were similar to that on CUFSF dataset.



(a) CML_{AUC} (b) CML_{AUC}

(c) CML_{pAUC} (d) CML_{pAUC}

Fig. 3. Results of CML_{AUC} and CML_{pAUC} with different number of sampled must-link and cannot-link constraints on CASIA NIR-VIS 2.0 dataset.



(a) CML_{AUC} (b) CML_{pAUC}

Fig. 4. Results of CML_{AUC} and CML_{pAUC} with different number of dimensions on CUFSF dataset.

For both AUC and pAUC optimization, when the sampled number of must-link constraints was greater than 10, the results were stable for all the performance measures. If the sampled numbers of cannot-link constraints were greater than 10 for AUC and 100 for pAUC, the results of AUC and pAUC were stable.

C. Results on CUFSF Dataset

1) Influence of Dimension: We first present results of the influence of the dimension on CUFSF dataset. The dimensions tested on this dataset were $[100, 200, ..., 800]$. For this experiment, the sampling ratios for both must-link and cannot-link constraints were set to 1. Using a smaller sampling ratio would not influence much the results as it has been shown in the previous part. All the four performance measures were evaluated. Results are provided in Fig. 4. As can be seen, for both AUC and pAUC optimization, all four performance measures improved with the increase in dimension. Specifically, the improvements tended to be small when the dimension reached 800. Thus for both CML_{AUC} and CML_{pAUC}, we set the dimension to 800 for reporting results.

2) Comparison with Single Modal Methods: Table I presents the results of the proposed methods on CUFSF

TABLE I
COMPARISON WITH SINGLE MODAL METHODS ON CUFSF DATASET, VR DENOTES THE VERIFICATION RATE AT FPR=0.1%

Methods	1-EER(%)	VR(%)	AUC	Rank-1(%)
PCA	85.5 ± 0.7	25.9 ± 4.8	0.936	35.0 ± 2.3
KDA	94.5 ± 0.4	50.2 ± 6.1	0.987	61.0 ± 2.6
LMNN	93.3 ± 0.5	34.7 ± 6.8	0.981	55.0 ± 1.4
NCA	95.1 ± 0.4	42.6 ± 2.4	0.989	59.1 ± 2.0
ITML	91.4 ± 0.8	34.8 ± 3.4	0.974	41.4 ± 2.0
LDML	94.3 ± 0.4	42.9 ± 5.7	0.986	59.8 ± 1.7
Maha_{AUC}	95.8 ± 0.4	55.6 ± 2.8	0.991	67.4 ± 1.7
$\text{Maha}_{\text{pAUC}}$	96.0 ± 0.5	58.3 ± 4.5	0.992	71.7 ± 1.5
PCA-C	93.6 ± 0.3	37.2 ± 8.1	0.983	48.7 ± 1.8
KDA-C	95.5 ± 0.2	37.4 ± 10.2	0.990	57.9 ± 1.3
LMNN-C	94.2 ± 1.9	33.1 ± 9.3	0.985	54.2 ± 7.5
NCA-C	94.6 ± 0.5	46.5 ± 1.5	0.988	52.4 ± 1.4
ITML-C	95.3 ± 0.3	45.1 ± 5.2	0.990	55.1 ± 1.7
LDML-C	93.1 ± 0.7	41.9 ± 3.8	0.981	47.6 ± 4.1
CML_{AUC}	96.3 ± 0.4	58.0 ± 5.4	0.994	73.6 ± 1.3
CML_{pAUC}	$\mathbf{96.5 \pm 0.3}$	$\mathbf{59.7 \pm 6.8}$	**0.995**	$\mathbf{75.5 \pm 1.5}$

TABLE II
COMPARISON WITH MULTI-MODAL METHODS ON CUFSF DATASET, VR DENOTES THE VERIFICATION RATE AT FPR=0.1%

Methods	1-EER(%)	VR(%)	AUC	Rank-1(%)
CSR	95.1 ± 0.3	51.9 ± 7.5	0.989	63.7 ± 2.0
KCSR	96.0 ± 0.3	39.5 ± 13.2	0.991	66.8 ± 1.8
CCA	93.6 ± 0.3	37.2 ± 8.1	0.983	48.7 ± 1.8
KCCA	95.2 ± 0.3	41.3 ± 9.6	0.990	57.4 ± 1.5
CDFE	91.8 ± 0.7	15.9 ± 4.8	0.974	48.6 ± 1.4
MvDA	90.7 ± 0.7	27.7 ± 3.1	0.967	36.0 ± 1.7
CMML	94.1 ± 0.5	43.7 ± 6.7	0.986	61.2 ± 1.9
HMLCR	95.7 ± 0.3	51.0 ± 10.9	0.992	68.7 ± 1.8
CML_{AUC}	96.3 ± 0.4	58.0 ± 5.4	0.994	73.6 ± 1.3
CML_{pAUC}	$\mathbf{96.5 \pm 0.3}$	$\mathbf{59.7 \pm 6.8}$	**0.995**	$\mathbf{75.5 \pm 1.5}$

TABLE III
COMPARISON WITH MULTI-MODAL METHODS ON CUFSF DATASET WITH DEEP FEATURES, VR DENOTES THE VERIFICATION RATE AT FPR=0.1%

Methods	1-EER(%)	VR(%)	AUC	Rank-1(%)
PCA	93.8 ± 0.2	46.0 ± 2.1	0.985	54.3 ± 2.3
CCA	97.4 ± 0.3	72.9 ± 1.6	0.997	76.8 ± 2.1
CSR	97.1 ± 0.3	63.0 ± 1.7	0.996	75.5 ± 2.0
KCSR	97.4 ± 0.3	62.5 ± 2.3	0.997	73.7 ± 1.8
HMLCR	97.2 ± 0.2	71.9 ± 2.1	0.997	77.8 ± 2.4
CML_{AUC}	98.1 ± 0.2	73.2 ± 1.9	**0.998**	79.5 ± 2.0
CML_{pAUC}	$\mathbf{98.1 \pm 0.1}$	$\mathbf{74.7 \pm 2.4}$	**0.998**	$\mathbf{80.6 \pm 1.6}$

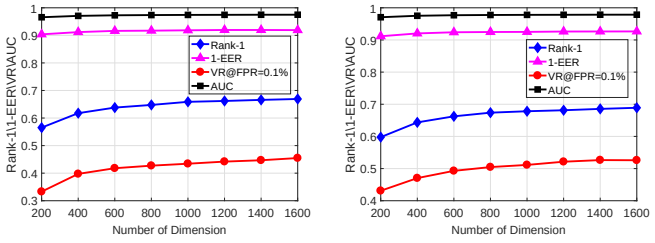
dataset and comparisons with single modal methods. The six compared methods are, PCA [40], Kernel Discriminant Analysis (KDA) [41], LMNN [16] and Neighbourhood Components Analysis (NCA) [42], ITML [15], and Logistic Discriminant-based Metric Learning (LDML) [38].

Since the proposed methods have two terms that may influence the final results: the first is the cross-modal metric to remove modality variations, and the second is AUC or pAUC optimization to learn discriminant metrics. The six methods were compared in two settings. In the first setting, original features were used. Results of the compared methods are in the first six rows of Table I. In the second setting, features processed with Canonical Correlation Analysis (CCA) [43] were used (denoted as the method with a suffix of '-C' in the table). Features under this setting should have less modality variability compared with the first setting. The proposed methods CML_{AUC} and CML_{pAUC} were using original features. Besides, we have also replaced the cross-modal metric in our methods with Mahalanobis metric for comparison, giving in two results of AUC and pAUC (in FPRs of range $[0, 0.1]$) optimization of Mahalanobis metric, denoted as Maha_{AUC} and $\text{Maha}_{\text{pAUC}}$.

As can be seen in the first eight rows of the table, Maha_{AUC} and $\text{Maha}_{\text{pAUC}}$ outperformed almost all of the single modal based methods. Though KDA was comparable with them with respect to 1-EER and AUC. This illustrates the superiority of AUC and pAUC optimization. The second observation is that results of CML_{AUC} and CML_{pAUC} were better than all the compared methods under both settings, with CML_{pAUC} the best. Compared with Maha_{AUC} and $\text{Maha}_{\text{pAUC}}$, CML_{AUC} and CML_{pAUC} should have better ability to remove modality variations, as demonstrated by the results of CML_{AUC} and CML_{pAUC}. Besides, compared with the methods with CCA preprocessing original features, the proposed methods are end to end with both the abilities to remove modality variations and to learn discriminant metric. In this case, the superiority is clear. Lastly, the results of pAUC optimization were better than that of AUC optimization for both Mahalanobis and cross-modal metrics. This is mainly because that the partial AUC of false positive range [0,0.1] is optimized. Compared with AUC optimization, pAUC optimizes with the top 10% most inseparable cannot-link constraints. There may be some redundancy in all the cannot-link constraints used in AUC optimization. Therefore, pAUC optimization can outperform AUC optimizaiton.

3) Comparison with Multi-Modal Methods: Comparison with multi-modal methods are also given in Table II. Compared methods include Coupled Spectral Regression (CSR) [44], Kernel Coupled Spectral Regression (KCSR) [44], Canonical Correlation Analysis (CCA) [43], Kernel Canonical Correlation Analysis (KCCA) [45], Common Discriminant Feature Extraction (CDFE) [46] and Multi-view Discriminant Analysis(MvDA) [47]. As can be seen, when compared with the best of these compared methods (CSR), CML_{AUC} improved over CSR significantly, specifically with VR@FPR=0.1% and Rank-1. Rank-1 is not directly related to AUC optimization, however, as can be seen, both CML_{AUC} and CML_{pAUC} achieved good Rank-1 rates. This illustrates that the proposed methods can not only perform well for AUC related performance measures, but also for other performance measures.

4) Results of Deep Features: Besides SIFT feature, we have also tested the proposed methods on deep features (VGG-Face). Results are summarized in Table III. With better features, performance on this dataset are improved to a large extent compared with SIFT features. The proposed methods still improved over the compared methods.



(a) CML_{AUC} (b) CML_{pAUC}

Fig. 5. Results of CML_{AUC} and CML_{pAUC} with different number of dimensions on CASIA NIR-VIS 2.0 dataset.

TABLE IV
COMPARISON WITH SINGLE MODAL METHODS ON CASIA NIR-VIS 2.0 DATASET, VR DENOTES THE VERIFICATION RATE AT FPR=0.1%

Methods	1-EER(%)	VR(%)	AUC	Rank-1(%)
PCA	81.0 ± 0.8	15.9 ± 2.1	0.889	28.4 ± 1.5
KDA	93.5 ± 0.4	44.5 ± 5.3	0.981	63.7 ± 1.4
LMNN	89.0 ± 1.1	29.1 ± 3.2	0.957	43.8 ± 1.6
ITML	83.9 ± 0.9	19.4 ± 2.2	0.921	30.3 ± 2.5
DML-eig	87.9 ± 0.6	28.8 ± 1.7	0.950	40.9 ± 1.4
Maha_{AUC}	93.1 ± 0.2	42.7 ± 7.1	0.980	72.0 ± 1.1
$\text{Maha}_{\text{pAUC}}$	93.3 ± 0.3	57.1 ± 3.3	0.981	72.4 ± 1.0
PCA-C	88.6 ± 0.4	29.0 ± 2.6	0.954	41.8 ± 1.4
KDA-C	92.8 ± 0.4	34.9 ± 5.4	0.978	55.8 ± 1.3
LMNN-C	89.3 ± 0.3	29.4 ± 3.3	0.959	44.1 ± 1.5
ITML-C	90.5 ± 0.5	33.8 ± 2.4	0.967	45.1 ± 1.2
DML-C	87.1 ± 1.2	24.2 ± 3.8	0.944	33.4 ± 4.7
CML_{AUC}	93.9 ± 0.4	55.0 ± 4.7	0.984	74.2 ± 0.8
CML_{pAUC}	$\mathbf{94.1 \pm 0.3}$	$\mathbf{58.3 \pm 3.8}$	**0.985**	$\mathbf{74.9 \pm 0.9}$

D. Results on CASIA NIR-VIS 2.0 Dataset

1) Influence of Dimension: The influence of the dimension on CASIA NIR-VIS 2.0 dataset is shown in Fig. 5. Since there are more training samples on CASIA NIR-VIS 2.0 dataset than CUFSF, the reduced dimension on this dataset can be higher than that on CUFSF. The tested dimensions on this dataset were $[200, 400, ..., 1600]$. As can be seen, all the performance measures increased with the dimension. The results almost converged when the dimension number reached 1,600. On this dataset, for final results reporting, the dimension used was 1,600.

2) Comparison with Single Modal Methods: Results of single modal methods on CASIA NIR-VIS 2.0 dataset are given in Table IV. The results of CML_{pAUC} were better than those of CML_{AUC}. Besides, $\text{Maha}_{\text{pAUC}}$ was better than Maha_{AUC}. In fact, for 1-EER, AUC and Rank-1, the differences of AUC and pAUC optimization were small. However, for VR@FPR=0.1%, the differences were relatively large. This is mainly because CML_{pAUC} and $\text{Maha}_{\text{pAUC}}$ optimizes the left most of the ROC curve, leading to an advantage over CML_{AUC} and Maha_{AUC} with respect to the VR@FPR=0.1% performance measure. Besides, compared with the state-of-the-art single modal methods under two settings, the four methods (Maha_{AUC}, $\text{Maha}_{\text{pAUC}}$, CML_{AUC}, CML_{pAUC}) under the proposed framework achieved very competitive results.

TABLE V
COMPARISON WITH MULTI-MODAL METHODS ON CASIA NIR-VIS 2.0 DATASET, VR DENOTES THE VERIFICATION RATE AT FPR=0.1%

Methods	1-EER(%)	VR(%)	AUC	Rank-1(%)
CSR	93.2 ± 0.4	53.3 ± 3.0	0.980	68.1 ± 1.1
KCSR	93.5 ± 0.4	47.4 ± 3.9	0.982	65.6 ± 1.1
CCA	88.6 ± 0.4	29.0 ± 2.6	0.954	41.8 ± 1.4
KCCA	91.2 ± 0.3	36.8 ± 2.9	0.970	49.5 ± 1.4
CDFE	83.5 ± 0.6	20.5 ± 1.4	0.914	29.4 ± 1.2
MvDA	81.8 ± 0.5	18.2 ± 1.9	0.899	23.6 ± 1.9
CMML	80.6 ± 0.8	17.3 ± 2.2	0.886	26.7 ± 1.4
HMLCR	92.0 ± 0.4	43.3 ± 4.6	0.975	60.9 ± 1.2
CML_{AUC}	93.9 ± 0.4	55.0 ± 4.7	0.984	74.2 ± 0.8
CML_{pAUC}	$\mathbf{94.1 \pm 0.3}$	$\mathbf{58.3 \pm 3.8}$	**0.985**	$\mathbf{74.9 \pm 0.9}$

TABLE VI
RESULTS OF DEEP FEATURES ON CASIA NIR-VIS 2.0 DATASET, VR DENOTES THE VERIFICATION RATE AT FPR=0.1%

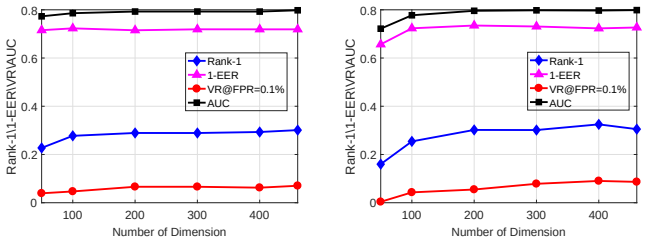
Methods	1-EER(%)	VR(%)	AUC	Rank-1(%)
PCA	96.4 ± 0.2	76.0 ± 1.7	0.995	82.1 ± 1.6
CSR	$\mathbf{98.3 \pm 0.2}$	88.1 ± 1.1	**0.998**	91.0 ± 1.2
KCSR	$\mathbf{98.3 \pm 0.2}$	$\mathbf{89.5 \pm 0.9}$	**0.998**	$\mathbf{92.4 \pm 0.8}$
CCA	98.1 ± 0.2	88.6 ± 1.1	**0.998**	91.5 ± 1.2
HMLCR	97.2 ± 0.3	78.6 ± 3.2	0.996	88.2 ± 0.6
CML_{AUC}	98.0 ± 0.1	86.8 ± 1.0	**0.998**	91.0 ± 1.1
CML_{pAUC}	97.9 ± 0.1	86.7 ± 0.9	**0.998**	91.0 ± 1.1

3) Comparison with Multi-Modal Methods: In Table V, comparisons with multi-modal methods are given. From the table, CML_{AUC} and CML_{pAUC} achieved AUC values of 0.984 and 0.985 respectively. The 1-EER for the two methods were $93.9 \pm 0.4\%$ and $\mathbf{94.1 \pm 0.3}\%$. For these two performance measures, only KCSR was comparable with our methods. For the Rank-1 performance measure, which is not related to AUC optimization, it is clear our methods also have an advantage over the compared methods. CML_{pAUC} at VR@FPR=0.1% was the best among all the methods, clearly demonstrating the effectiveness of pAUC optimization.

4) Results of Deep Features: Table VI provides results of deep features on CASIA NIR-VIS 2.0. With deep features, CSR, KCSR, CCA, CML_{AUC} and CML_{pAUC} all achieved the best AUC. KCSR was the best for the three other performance measures. The results of the proposed methods are quite comparable with the results of KCSR. Comparing Table VI and Table V, deep features improved on the SIFT features significantly.

E. Results on NJU-ID Dataset

1) Influence of Dimension: As the number of training samples on NJU-ID dataset is relatively small, the maximum reduced dimension is 461. On this dataset, the dimension number was varied between [50,100,200,300,400,461]. Corresponding results are prsented in Fig. 6. On this dataset, the two sample ratios were both set to 1. With varing dimensions, the results of all the four adopted performance measures tended to increase with the dimension. Another observation is that CML_{AUC} appeared more stable than CML_{pAUC}. This is mainly due to that the number of cannot-link constraints used by CML_{AUC} was larger than



(a) CML_{AUC} (b) CML_{pAUC}

Fig. 6. Results of CML_{AUC} and CML_{pAUC} with different number of dimensions on NJU-ID dataset.

TABLE VII
COMPARISON WITH SINGLE MODAL METHODS ON NJU-ID DATASET, VR DENOTES THE VERIFICATION RATE AT FPR=0.1%

Methods	1-EER(%)	VR(%)	AUC	Rank-1(%)
PCA	64.9 ± 5.3	3.1 ± 3.1	0.701	23.0 ± 7.6
KDA	71.9 ± 6.9	3.5 ± 4.8	0.781	27.8 ± 9.0
LMNN	70.3 ± 6.3	3.9 ± 4.2	0.770	23.8 ± 5.0
ITML	70.7 ± 3.9	3.1 ± 4.1	0.766	26.6 ± 4.8
NCA	65.6 ± 3.8	0.4 ± 1.2	0.708	21.1 ± 5.7
LDML	73.1 ± 4.9	6.6 ± 5.8	0.799	29.7 ± 8.5
Maha_{AUC}	$\mathbf{73.5 \pm 5.7}$	7.8 ± 6.6	**0.815**	32.9 ± 9.6
$\text{Maha}_{\text{pAUC}}$	72.7 ± 6.0	8.2 ± 5.6	0.812	$\mathbf{33.6 \pm 10.5}$
KDA-C	68.0 ± 6.0	3.9 ± 3.2	0.730	20.0 ± 8.2
LMNN-C	69.6 ± 4.9	2.8 ± 4.2	0.749	21.5 ± 6.9
ITML-C	68.0 ± 2.6	1.9 ± 2.7	0.724	14.5 ± 4.7
NCA-C	68.8 ± 2.3	3.9 ± 3.7	0.749	21.9 ± 7.8
LDML-C	69.2 ± 5.1	4.7 ± 3.6	0.754	24.7 ± 9.6
CML_{AUC}	73.1 ± 4.2	$\mathbf{8.6 \pm 6.3}$	0.805	30.9 ± 11.8
CML_{pAUC}	72.3 ± 4.2	$\mathbf{8.6 \pm 6.3}$	0.805	30.1 ± 9.4

that of CML_{pAUC} and this dataset is relatively small. Thus CML_{pAUC} could not use sufficient information for training. On this dataset, for final results reporting, we used the dimension of 461.

2) Comparison with Single Modal Methods: Results of single modal methods on NJU-ID dataset are given in Table VII. As this dataset is relatively small, the results on this dataset are generally worse than those on CUFSF and CASIA NIR-VIS 2.0. On this dataset, Maha_{AUC} and $\text{Maha}_{\text{pAUC}}$ were slightly better than CML_{AUC} and CML_{pAUC}. This may be because this dataset is smaller, cross-modal metric with more parameters compared with Mahalanobis metric is prone to overfitting. However, Maha_{AUC}, $\text{Maha}_{\text{pAUC}}$, CML_{AUC}, CML_{pAUC} still outperformed the compared methods as shown in Table VII.

3) Comparison with Multi-Modal Methods: Results of comparison with multi-modal methods on NJU-ID are given in Table VIII. Among all the compared methods, only CSR was comparable with CML_{pAUC}. As the RID card face verification is a real-world application, which requires low FPR rate and high TPR rate, the VR@FPR=0.1% performance measure on this dataset is low, indicating that there are still rooms for improvement. We further give results of deep features on this dataset.

4) Results of Deep Features: Results of Deep Features on NJU-ID are summarized in Table IX. The best results were achieved by CML_{pAUC}, with CSR and HMLCR achieved

TABLE VIII
COMPARISON WITH MULTI-MODAL METHODS ON NJU-ID DATASET, VR DENOTES THE VERIFICATION RATE AT FPR=0.1%

Methods	1-EER(%)	VR(%)	AUC	Rank-1(%)
CSR	71.9 ± 5.1	7.9 ± 7.0	0.791	29.3 ± 9.9
KCSR	70.0 ± 4.6	3.9 ± 3.7	0.758	22.0 ± 9.7
CCA	68.0 ± 3.9	3.5 ± 4.3	0.736	20.3 ± 7.6
KCCA	68.8 ± 4.3	4.7 ± 5.5	0.742	21.9 ± 7.9
CDFE	72.3 ± 4.0	3.9 ± 5.5	0.780	24.6 ± 4.9
MvDA	66.4 ± 5.1	1.6 ± 3.9	0.709	16.5 ± 7.3
CMML	66.8 ± 5.4	1.6 ± 2.7	0.727	20.3 ± 7.4
HMLCR	69.9 ± 4.8	3.9 ± 4.2	0.771	27.7 ± 11.3
CML_{AUC}	$\mathbf{73.1 \pm 4.2}$	$\mathbf{8.6 \pm 6.3}$	**0.805**	$\mathbf{30.9 \pm 11.8}$
CML_{pAUC}	72.3 ± 4.2	$\mathbf{8.6 \pm 6.3}$	**0.805**	30.1 ± 9.4

TABLE IX
RESULTS OF DEEP FEATURES ON NJU-ID DATASET, VR DENOTES THE VERIFICATION RATE AT FPR=0.1%

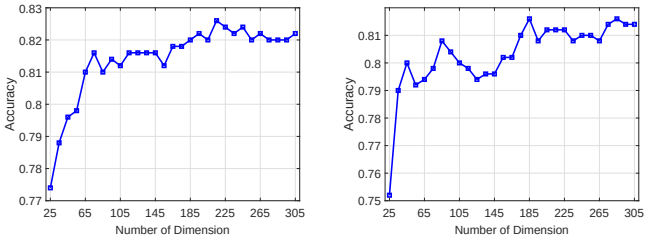
Methods	1-EER(%)	VR(%)	AUC	Rank-1(%)
PCA	90.3 ± 4.5	27.3 ± 15.7	0.949	65.6 ± 12.4
CCA	93.0 ± 4.0	27.8 ± 7.5	0.971	71.4 ± 10.1
CSR	94.9 ± 3.2	56.2 ± 9.9	0.985	87.5 ± 7.2
KCSR	91.0 ± 2.7	29.8 ± 18.9	0.967	73.0 ± 9.2
HMLCR	94.9 ± 1.9	53.1 ± 11.8	0.985	87.9 ± 6.0
CML_{AUC}	94.5 ± 2.7	50.5 ± 14.6	0.984	87.1 ± 4.2
CML_{pAUC}	$\mathbf{95.3 \pm 2.5}$	$\mathbf{58.7 \pm 13.0}$	**0.986**	$\mathbf{90.2 \pm 5.0}$

comparable results. Compared with the results of SIFT features, results of deep features are much better. However, the best result of VR@FPR=0.1% was 58.7 ± 13.0, which is still far from satisfactory. Therefore, there is still need for further improvement even with deep features.

F. Results on LFW Dataset

1) Influence of Dimension: The results of our methods on LFW dataset with different dimensions are shown in Fig. 7. The dimension was varied from 25 to 305. For both CML_{AUC} and CML_{pAUC}, the results increased with the dimension. The improvement in accuracy became convergent when the dimension exceeded 250. Different from LDML whose best dimension was 35, it can be seen the best dimension of the proposed methods was around 300. For the following results, the dimension was set to 300.

2) Comparison with Other Methods: On LFW, the proposed methods were compared with two single modal metric learning methods, LDML [38] and DML-eig [48]. The restricted setting of LFW was used. Note that LMNN is not suitable for this scenario as no label information is available. Results of our methods and the compared methods are given in Table X. Results of the compared methods were cited from their papers for the original and squared root SIFT features. The results for VGG-Face features were obtained using the source codes provided by the authors. The results of CML_{AUC} and CML_{pAUC} were similar on this dataset. Compared with LDML and DML-eig, our methods achieved better results. This illustrates the strength of our methods compared with those pair-based metric learning methods.



(a) CML_{AUC} (b) CML_{pAUC}

Fig. 7. Results of CML_{AUC} and CML_{pAUC} with different number of dimensions on LFW dataset.

TABLE X
COMPARISON OF ACCURACY WITH OTHER METHODS ON LFW

Method	Original	Square Root	VGG-Face
LDML	76.6 ± 0.7	77.5 ± 0.5	97.4 ± 1.0
DML-eig	80.6 ± 1.7	81.3 ± 2.3	96.7 ± 0.9
CML_{AUC}	82.5 ± 1.3	83.0 ± 1.3	$\mathbf{97.8 \pm 0.8}$
CML_{pAUC}	$\mathbf{82.5 \pm 1.2}$	$\mathbf{83.4 \pm 0.8}$	97.7 ± 0.9

G. Comparison of Training Time

A comparison of training time of our methods and LMNN on CUFSF dataset is given in Table XI. For LMNN, we used the code provided by its authors and the nearest neighbor was used for constructing triplets. The sub-sample parameter of the code was set to 0.3. For both CML_{AUC} and CML_{pAUC}, three settings were tested. Under the first setting, the numbers of must-link and cannot-link constraints were set to 100. Under the other two settings, the numbers of must-link constraints were both set to 700 and the numbers of cannot-link constraints were set 1,000 and 10,000 respectively. In Table XI, the two numbers in brackets denote the sampled numbers of must-link and cannot-link constraints. All the experiments were run on the same computer with an Intel Core i5-4278U processor and 8GB RAM running Windows 10. Both LMNN and our methods were tested using MATLAB R2014a. For the projection step of our methods with eigenvalue decomposition, the implementation of [7] was adopted.

From Table XI, it can be seen that our methods are faster when the dimension is less than 600. With increases in the dimension, both LMNN and our methods increase in training time. However, as the implementation of LMNN is not based on optimizing PSD matrix, there is no additional operation at each iteration to project matric parameter onto PSD cone. Differently, the operation ϕ_λ^+ in our algorithms in fact relies on a full eigenvalue decomposition at each iteration, which is time consuming. That is why the increase in training time is faster than that of LMNN. This indicates there is room for improvement to make our methods more efficient, for example, using coordinate descent [49] for LogDet optimization. The second observation is the training time of setting sampled numbers as (100,100), (700,1000) and (700,10000) is similar. There is little increase in training time with the sampled number increases. This is mainly because sampled numbers of constraints only influence the update of matrix $\mathbf{P}_A$. With the techniques in Section III to update matrix $\mathbf{P}_A$, the time for updating of matrix $\mathbf{P}_A$ grows slowly with increase in the number of training pairs used.

TABLE XI
COMPARISON OF TRAINING TIME ON CUFSF DATASET (IN SECONDS)

Methods\Dimension	200	400	600	800
LMNN	357	689	1114	1661
$\text{CML}_{\text{AUC}}(100, 100)$	101	381	979	2068
$\text{CML}_{\text{pAUC}}(100, 100)$	101	378	979	2088
$\text{CML}_{\text{AUC}}(700, 1000)$	110	414	1048	2199
$\text{CML}_{\text{pAUC}}(700, 1000)$	113	409	1041	2193
$\text{CML}_{\text{AUC}}(700, 10000)$	138	433	1068	2223
$\text{CML}_{\text{pAUC}}(700, 10000)$	120	414	1049	2213

V. CONCLUSION

For cross-modal metric learning, most of the previous methods minimize loss defined on must-link and cannot-link pairs. Since the numbers of must-link and cannot-link pairs can be highly imbalanced and AUC is a more meaningful performance measure. We propose to learn cross-modal metrics that directly maximize AUC defined on must-link and cannot-link pairs. An extension of our proposed method can also optimize partial AUC. The proposed methods are formulated as a LogDet regularized semi-definite optimization problem. For efficient optimization, a mini-batch proximal point algorithm is proposed. Extensive experiments have verified that both proposed CML_{AUC} and CML_{pAUC} methods achieved marked improvements over the existing methods. Specifically, CML_{pAUC} works well for performance measures like VR@FPR=0.1% and Rank-1. Future work includes improving the efficiency of the optimization methods, for example, using coordinate descent based methods for LogDet optimization.

REFERENCES

[1] A. Mignon and F. Jurie, "CMML: a new metric learning approach for cross modal matching," in *Proc. Asian Conf. Comput. Vis.*, 2012, pp. 1–14.
[2] L. Wu, L. Du, B. Liu, G. Xu, Y. Ge, Y. Fu, J. Li, Y. Zhou, and H. Xiong, "Heterogeneous metric learning with content-based regularization for software artifact retrieval," in *Proc. Int. Conf. Data Mining*, 2014, pp. 610–619.
[3] L. Wu, J. Li, X. Hu, and H. Liu, "Gleaning wisdom from the past: Early detection of emerging rumors in social media," in *Proc. SIAM Int. Conf. Data Mining*. SIAM, 2017, pp. 99–107.
[4] L. Wu, X. Hu, F. Morstatter, and H. Liu, "Adaptive spammer detection with sparse group modeling." in *Proc. Int. Conf. Web Social Media*, 2017, pp. 319–326.
[5] Q. You, J. Luo, H. Jin, and J. Yang, "Robust image sentiment analysis using progressively trained and domain transferred deep networks." in *Proc. AAAI Conf. Artif. Intell.*, 2015, pp. 381–388.
[6] P. Zhao, R. Jin, T. Yang, and S. C. Hoi, "Online AUC maximization," in *Proc. Int. Conf. Mach Learn.*, 2011, pp. 233–240.
[7] C. Wang, D. Sun, and K.-C. Toh, "Solving log-determinant optimization problems by a newton-cg primal proximal point algorithm," *SIAM J. Optim.*, vol. 20, no. 6, pp. 2994–3013, 2010.
[8] J. Yang, D. Sun, and K.-C. Toh, "A proximal point algorithm for log-determinant optimization with group lasso regularization," *SIAM J. Optim.*, vol. 23, no. 2, pp. 857–893, 2013.
[9] X. Xu, W. Li, and D. Xu, "Distance metric learning using privileged information for face verification and person re-identification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 12, pp. 3150–3162, 2015.

[10] J. Li, X. Lin, X. Rui, Y. Rui, and D. Tao, "A distributed approach toward discriminative distance metric learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 9, pp. 2111–2122, 2015.
[11] H. Jia, Y. Cheung, and J. Liu, "A new distance metric for unsupervised learning of categorical data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 5, pp. 1065–1079, 2016.
[12] C. Shen, J. Kim, F. Liu, L. Wang, and A. Van Den Hengel, "Efficient dual approach to distance metric learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 2, pp. 394–406, 2014.
[13] S. Ying, Z. Wen, J. Shi, Y. Peng, J. Peng, and H. Qiao, "Manifold preserving: An intrinsic approach for semisupervised distance metric learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. PP, no. 99, pp. 1–12, 2017.
[14] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell, "Distance metric learning with application to clustering with side-information," *Proc. Adv. Neural Inf. Process. Syst.*, vol. 15, pp. 505–512, 2003.
[15] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, "Information-theoretic metric learning," in *Proc. Int. Conf. Mach Learn.*, 2007, pp. 209–216.
[16] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *J. Mach. Learn. Res.*, vol. 10, pp. 207–244, 2009.
[17] B. McFee and G. R. Lanckriet, "Metric learning to rank," in *Proc. Int. Conf. Mach Learn.*, 2010, pp. 775–782.
[18] D. Lim, G. Lanckriet, and B. McFee, "Robust structural metric learning," in *Proc. Int. Conf. Mach Learn.*, 2013, pp. 615–623.
[19] G. Chechik, V. Sharma, U. Shalit, and S. Bengio, "Large scale online learning of image similarity through ranking," *J. Mach. Learn. Res.*, vol. 11, pp. 1109–1135, 2010.
[20] W. Liu, C. Mu, R. Ji, S. Ma, J. R. Smith, and S.-F. Chang, "Low-rank similarity metric learning in high dimensions," in *Proc. AAAI Conf. Artif. Intell.*, 2015, pp. 2792–2799.
[21] K. Liu, A. Bellet, and F. Sha, "Similarity learning for high-dimensional sparse data." in *Proc. Int. Conf. Artif. Intell. Statist.*, 2015, pp. 653–662.
[22] P. Zhou, L. Du, M. Fan, and Y.-D. Shen, "An LLE based heterogeneous metric learning for cross-media retrieval," in *Proc. SIAM Int. Conf. Data Mining*, pp. 64–72.
[23] N. Quadrianto and C. H. Lampert, "Learning multi-view neighborhood preserving projections," in *Proc. Int. Conf. Data Mining*, 2011, pp. 425–432.
[24] C. Kang, S. Liao, Y. He, J. Wang, W. Niu, S. Xiang, and C. Pan, "Cross-modal similarity learning: A low rank bilinear formulation," in *Proc. Int. Conf. Inform. Knowl. Manage.*, 2015, pp. 1251–1260.
[25] Y. Zhen, P. Rai, H. Zha, and L. Carin, "Cross-modal similarity learning via pairs, preferences, and active supervision," in *Proc. AAAI Conf. Artif. Intell.*, 2015, pp. 3203–3209.
[26] T. Joachims, "A support vector method for multivariate performance measures," in *Proc. Int. Conf. Mach Learn.*, 2005, pp. 377–384.
[27] W. Gao, R. Jin, S. Zhu, and Z.-H. Zhou, "One-pass AUC optimization," in *Proc. Int. Conf. Mach Learn.*, 2013, pp. 906–914.
[28] Y. Ding, P. Zhao, S. C. Hoi, and Y.-S. Ong, "An adaptive gradient method for online auc maximization," in *Proc. AAAI Conf. Artif. Intell.*, 2015, pp. 2568–2574.
[29] H. Narasimhan and S. Agarwal, "A structural svm based approach for optimizing partial auc," in *Proc. Int. Conf. Mach Learn.*, 2013, pp. 516–524.
[30] P. Kar, H. Narasimhan, and P. Jain, "Online and stochastic gradient methods for non-decomposable loss functions," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 694–702.
[31] L. Bottou, "Online algorithms and stochastic approximations," in *Online Learn. Neural Netw.*, D. Saad, Ed. Cambridge University Press, 1998.
[32] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," Univ. of Massachusetts, Amherst, Tech. Rep. 07-49, October 2007.
[33] G. B. H. E. Learned-Miller, "Labeled faces in the wild: Updates and new reporting procedures," Univ. of Massachusetts, Amherst, Tech. Rep. UM-CS-2014-003, May 2014.
[34] W. Zhang, X. Wang, and X. Tang, "Coupled information-theoretic encoding for face photo-sketch recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 513–520.
[35] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, "The feret evaluation methodology for face-recognition algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 10, pp. 1090–1104, 2000.
[36] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
[37] S. Z. Li, D. Yi, Z. Lei, and S. Liao, "The casia nir-vis 2.0 face database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2013, pp. 348–353.
[38] M. Guillaumin, J. Verbeek, and C. Schmid, "Is that you? metric learning approaches for face identification," in *Proc. Int. Conf. Comput. Vis.*, 2009, pp. 498–505.
[39] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *Proc. Brit. Mach. Vis. Conf.*, 2015.
[40] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 1991, pp. 586–591.
[41] D. Cai, X. He, and J. Han, "Speed up kernel discriminant analysis," *VLDB J.*, vol. 20, no. 1, pp. 21–33, 2011.
[42] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov, "Neighbourhood components analysis," in *Proc. Adv. Neural Inf. Process. Syst.*, 2004, pp. 513–520.
[43] H. Hotelling, "Relations between two sets of variates," *Biometrika*, pp. 321–377, 1936.
[44] Z. Lei and S. Z. Li, "Coupled spectral regression for matching heterogeneous faces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 1123–1128.
[45] P. L. Lai and C. Fyfe, "Kernel and nonlinear canonical correlation analysis," *Int'l J. Neural Syst.*, vol. 10, no. 05, pp. 365–377, 2000.
[46] D. Lin and X. Tang, "Inter-modality face recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 13–26.
[47] M. Kan, S. Shan, H. Zhang, S. Lao, and X. Chen, "Multi-view discriminant analysis," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 808–821.
[48] Y. Ying and P. Li, "Distance metric learning with eigenvalue optimization," *J. Mach. Learn. Res.*, vol. 13, pp. 1–26, 2012.
[49] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2008.

Jing Huo received the Ph.D. degree from the Department of Computer Science and Technology of Nanjing University in 2017. She is currently an assistant researcher in the Department of Computer Science and Technology of Nanjing University, China. Her research interests lie in machine learning and computer vision. Her work currently focuses on metric learning, subspace learning and their applications to heterogeneous face recognition.

Yang Gao received the Ph.D. degree from the Department of Computer Science and Technology of Nanjing University in 2000. He is a Professor at the Department of Computer Science and Technology, Nanjing University. His research interests include artificial intelligence and machine learning. He has published more than 100 papers in top conferences and journals in and outside of China. He is a member of the IEEE.

Yinghuan Shi is currently an associate professor in the Department of Computer Science and Technology of Nanjing University, China. He received his Ph.D. and B.Sc. degrees from Department of Computer Science of Nanjing University in 2013 and 2007, respectively. He was a visiting scholar in University of North Carolina at Chapel Hill, and University of Technology Sydney, respectively. His research interests include computer vision and medical image analysis. He has published more than 40 research papers in related journals and conferences such as TPAMI, TBME, TNNLS, TCYB, CVPR, AAAI, ACMMM, MICCAI and IPMI. He serves as a program committee member for several conferences, and also as a referee for several journals.

Hujun Yin is a Senior Lecturer (Associate Professor) at the University of Manchester, School of Electrical and Electronic Engineering. He received the PhD degree in neural networks from University of York, and MSc degree in signal processing and BEng degree in electronic engineering from Southeast University. His main areas of research and expertise are neural networks, self-organising learning, image processing, face recognition, time series and bio-/neuro-informatics. He has published over 150 peer-reviewed articles in a range of topics from density modelling, image processing, face recognition, text mining and knowledge management, gene expression analysis and peptide sequencing, novelty detection, to financial time series modelling and decoding neuronal responses. He is a senior member of the IEEE.