

---

# Curvilinear Distance Metric Learning

---

Shuo Chen<sup>†</sup>, Lei Luo<sup>‡\*</sup>, Jian Yang<sup>†\*</sup>, Chen Gong<sup>†</sup>, Jun Li<sup>§</sup>, Heng Huang<sup>‡</sup>

## Abstract

Distance Metric Learning aims to learn an appropriate metric that faithfully measures the distance between two data points. Traditional metric learning methods usually calculate the pairwise distance with fixed distance functions (*e.g.*, Euclidean distance) in the projected feature spaces. However, they fail to learn the underlying geometries of the sample space, and thus cannot exactly predict the intrinsic distances between data points. To address this issue, we first reveal that the traditional linear distance metric is equivalent to the cumulative arc length between the data pair’s nearest points on the learned straight measurer lines. After that, by extending such straight lines to general curved forms, we propose a Curvilinear Distance Metric Learning (CDML) method, which adaptively learns the nonlinear geometries of the training data. By virtue of Weierstrass theorem, the proposed CDML is equivalently parameterized with a 3-order tensor, and the optimization algorithm is designed to learn the tensor parameter. Theoretical analysis is derived to guarantee the effectiveness and soundness of CDML. Extensive experiments on the synthetic and real-world datasets validate the superiority of our method over the state-of-the-art metric learning models.

## 1 Introduction

The goal of a Distance Metric Learning (DML) algorithm is to learn the distance function for data pairs to measure their similarities. The learned distance metric successfully reflects the relationships within data points and significantly improves the performance of many subsequent learning tasks, such as classification [22], clustering [23], retrieval [24], and verification [14]. It has recently become an active research topic in machine learning community [30, 29].

The well-studied DML methods are usually linear, namely Mahalanobis distance metric based models [23]. Under the supervisions of pairwise similarities, they intend to learn a Semi-Positive-Definite (SPD) matrix  $M = PP^T \in \mathbb{R}^{d \times d}$  to decide the squared parametric distance  $\text{Dist}_P^2(x, \hat{x}) = (x - \hat{x})^T M (x - \hat{x})$  between data points  $x$  and  $\hat{x}$  in the  $d$ -dimensional space. It is notable that such a linear Mahalanobis distance is equivalent to the Euclidean distance in the  $m$ -dimensional feature space projected by  $P \in \mathbb{R}^{d \times m}$ . To perform the learning of the parameter  $M$ , intensive efforts have been put to design various loss functions and constraints in optimization models. The early works Large Margin Nearest Neighbor (LMNN) [27] and Information-Theoretic Metric Learning (ITML) [7] utilized the must-link and cannot-link information to constrain the ranges of the learned distances. Instead of fixing the distance ranges in the objective, Geometric Mean Metric Learning (GMML) [31]

---

<sup>†</sup>S. Chen, J. Yang, and C. Gong are with the PCA Lab, Key Lab of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, and Jiangsu Key Lab of Image and Video Understanding for Social Security, School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (E-mail: {shuochen, csjyang, chen.gong}@njust.edu.cn).

<sup>‡</sup>L. Luo and H. Huang are with the Electrical and Computer Engineering, University of Pittsburgh, and also with JD Finance America Corporation, USA (E-mail: lel94@pitt.edu, henghuanghh@gmail.com).

<sup>§</sup>J. Li is with the Institute for Medical Engineering & Science, Massachusetts Institute of Technology, Cambridge, MA, USA (E-mail: junli@mit.edu).

\*L. Luo and J. Yang are corresponding authors.

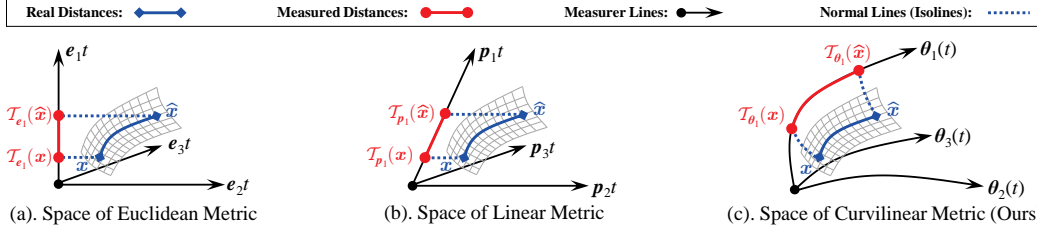


Figure 1: Conceptual illustrations of Euclidean metric, linear (Mahalanobis) metric, and our proposed curvilinear metric in three-dimensional space. For a pair of data points (*i.e.*,  $\mathbf{x}$  and  $\hat{\mathbf{x}}$ ) from the spatial surface, metrics find out the nearest (calibration) points (*i.e.*,  $\mathcal{T}(\mathbf{x})$  and  $\mathcal{T}(\hat{\mathbf{x}})$ ) on each learned measurer line, and then use the arc length between nearest points as measured distance results. By the curved measurer lines, our method can measure the intrinsic curvilinear distance more exactly.

proposed a geometric loss to jointly minimize the intra-class distances and maximize the inter-class distances as much as possible. Considering that the above methods utilizing one single matrix  $\mathbf{M}$  are not flexible for complex data, the traditional Mahalanobis distance is extended to a combined form of multiple linear metrics [30]. Recently, the strategies of adversarial training [8] and collaborative training [20] were introduced in Adversarial Metric Learning (AML) [4] and Bilevel Distance Metric Learning (BDML) [29], respectively, which showed further improvements on the linear metric.

To enhance the flexibility of DML for fitting data pairs from nonlinear sample spaces, the early works transferred the original data points to the high-dimensional kernel space by using traditional kernel methods [26]. Recently, the projection matrix  $\mathbf{P}$  of the linear DML was extended to a nonlinear feature mapping form  $\mathbf{P}\mathcal{W}(\cdot)$ , in which the mapping  $\mathcal{W}(\cdot)$  is implemented by typical Deep Neural Networks (DNN), such as Convolutional Neural Network (CNN) [32] and Multiple Layer Perceptron (MLP) [10]. To further utilize the fitting capability of DNN and characterize the relative distances, the traditional pairwise loss was extended to multi-example forms, such as Npair loss [22] and Angular loss [25]. It is worth pointing out that the above kernelized metrics and DNN based metrics are still calculated with fixed Euclidean distance in the extracted feature space, which ignores the geometric structures of the sample space. To this end, some recent works proposed to learn the projection matrix  $\mathbf{P}$  on differential manifolds (*e.g.*, SPD manifold [33] and Grassmann manifold [13]) to improve the representation capability on some specific nonlinear data structures. However, the geometries of their used manifolds are usually specified and cannot be learned to adapt to various nonlinear data, and hence remarkably hindering the generality of the manifold based DML approaches.

Although above existing DML models have achieved promising results to some extent, most of them fail to learn the spatial geometric structures of the sample space, and thus their obtained metrics cannot reflect the intrinsic curvilinear distances between data points. To address this challenging problem, in this paper, we firstly present a new interpretation to reveal that the traditional linear distance metric is equivalent to the cumulative arc length between data pair’s nearest points on the straight measurer lines (see Fig. 1(a) and (b)). Such straight measurer lines can successfully learn the directions of real distances, but they are not capable of adapting to the curvilinear distance geometries on many nonlinear datasets. Therefore, we propose the Curvilinear Distance Metric Learning (CDML) model, which extends the straight measurer lines to the general smooth curved lines (see Fig. 1(c)). Thanks to the generalized forms of such curvilinear measurers, the geometries of training data can be adaptively learned, so that the nonlinear pairwise distances can be reasonably measured. We theoretically analyze the effectiveness of CDML by showing its fitting capability and generalization bound. Furthermore, we prove that our proposed curvilinear distance satisfies the topological definitions of the (pseudo-)metric, which demonstrates the geometric soundness of such a distance metric. The main contributions of this paper are summarized as: **(I)**. We provide a new intuitive interpretation for traditional linear metric learning by explicitly formulating the measurer lines and measurement process; **(II)**. We propose a generalized metric learning model dubbed CDML with discovering the curvilinear distance hidden in the nonlinear data, and the corresponding optimization algorithm is designed to solve the proposed model which is guaranteed to converge; **(III)**. The complete theoretical guarantee is established, which analyzes the fitting capability, generalization bound, and topological property of CDML, and therefore ensuring the model effectiveness and soundness; **(IV)**. CDML is experimentally validated to outperform state-of-the-art metric learning models on both synthetic datasets and real-world datasets.

**Notations.** Throughout this paper, we write matrices, vectors, and 3-order tensors as bold uppercase characters, bold lowercase characters, and bold calligraphic uppercase characters, respectively. For a 3-order tensor  $\mathcal{M}$ , the notations  $\mathcal{M}_{i::}$ ,  $\mathcal{M}_{:i}$ , and  $\mathcal{M}_{::i}$  denote the horizontal, lateral, and frontal slices. The tube fiber, row fiber, and column fiber as  $\mathcal{M}_{i,j}$ ,  $\mathcal{M}_{i,:}$ , and  $\mathcal{M}_{:,j}$ . Let  $\mathbf{y} = (y_1, y_2, \dots, y_N)^\top$  be the label vector of training data pairs  $\mathcal{X} = \{(\mathbf{x}_j, \hat{\mathbf{x}}_j) | j = 1, 2, \dots, N\}$  with  $\mathbf{x}_j, \hat{\mathbf{x}}_j \in \mathbb{R}^d$ , where  $y_j = 1$  if  $\mathbf{x}_j$  and  $\hat{\mathbf{x}}_j$  are similar, and  $y_j = 0$  otherwise. Here  $d$  is the data dimensionality, and  $N$  is the total number of data pairs. The operators  $\|\cdot\|_2$  and  $\|\cdot\|_F$  denote the vector  $\ell_2$ -norm and matrix (tensor) Frobenius-norm, respectively. The notation  $\mathbb{N}_n = \{1, 2, \dots, n\}$  for any  $n \in \mathbb{N}$ .

## 2 Curvilinear Distance Metric Learning

In this section, we first present a new geometric interpretation for traditional linear metric learning models. Then the Curvilinear Distance Metric Learning (CDML) is formulated based on such an interpretation. The optimization algorithm is designed to solve CDML with convergence guarantee.

### 2.1 A Geometric Interpretation for Linear Metric

It is well known that the linear distance metric (*i.e.*, squared Mahalanobis distance) [30, 29] between two given data points  $\mathbf{x}, \hat{\mathbf{x}} \in \mathbb{R}^d$  is defined as

$$\text{Dist}_{\mathbf{P}}^2(\mathbf{x}, \hat{\mathbf{x}}) = (\mathbf{x} - \hat{\mathbf{x}})^\top \mathbf{M}(\mathbf{x} - \hat{\mathbf{x}}) = (\mathbf{x} - \hat{\mathbf{x}})^\top \mathbf{P}\mathbf{P}^\top(\mathbf{x} - \hat{\mathbf{x}}), \quad (1)$$

where the matrix  $\mathbf{M} = \mathbf{P}\mathbf{P}^\top$  is assumed to be SPD in  $\mathbb{R}^{d \times d}$ . In previous works, the above linear distance metric is usually interpreted as the Euclidean distance in the projection space, where the projection matrix  $\mathbf{P} \in \mathbb{R}^{d \times m}$  plays the role of feature extractions [28]. Here  $d$  and  $m$  are the dimensionalities of the original sample space and the extracted feature space, respectively. Although such an interpretation offers a friendly way for model extensions, it is not clear enough that why the linear distance metric fails to characterize the curvilinear distances on nonlinear data.

Now we present a new understanding for the linear distance metric from its measurement process, which offers a clear way to hand the nonlinear geometric data. We denote  $\mathbf{p}_i \in \mathbb{R}^d$  as the  $i$ -th column of  $\mathbf{P}$ . By using the rule of inner products, Eq. (1) equals to the following cumulative form<sup>1</sup>

$$\sum_{i=1}^m \|\mathbf{p}_i\|_2^2 \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 \cos^2 \langle \mathbf{p}_i, \mathbf{x} - \hat{\mathbf{x}} \rangle = \sum_{i=1}^m \|\mathbf{p}_i\|_2^2 \|\mathbf{p}_i T_i(\mathbf{x}) - \mathbf{p}_i T_i(\hat{\mathbf{x}})\|_2^2, \quad (2)$$

where  $T_i(\mathbf{x})$  and  $T_i(\hat{\mathbf{x}})$  are the projection points of  $\mathbf{x}$  and  $\hat{\mathbf{x}}$ , which satisfy  $(\mathbf{p}_i T_i(\mathbf{x}) - \mathbf{x})^\top \mathbf{p}_i = 0$  and  $(\mathbf{p}_i T_i(\hat{\mathbf{x}}) - \hat{\mathbf{x}})^\top \mathbf{p}_i = 0$ , respectively. After that, the  $\ell_2$ -norm distance  $\|\mathbf{p}_i T_i(\mathbf{x}) - \mathbf{p}_i T_i(\hat{\mathbf{x}})\|_2$  is equivalently converted to the arc length from  $T_i(\mathbf{x})$  to  $T_i(\hat{\mathbf{x}})$  on the straight line  $\mathbf{z} = \mathbf{p}_i t$  ( $t \in \mathbb{R}$ ), and thus the squared Mahalanobis distance is rewritten as

$$\text{Dist}_{\mathbf{P}}^2(\mathbf{x}, \hat{\mathbf{x}}) = \sum_{i=1}^m \|\mathbf{p}_i\|_2^2 \left( \int_{T_i(\mathbf{x})}^{T_i(\hat{\mathbf{x}})} \|\mathbf{p}_i\|_2 dt \right)^2, \quad (3)$$

where the integral value is the arc length of the straight *measurer line*  $\mathbf{z} = \mathbf{p}_i t$  from  $T_i(\mathbf{x})$  to  $T_i(\hat{\mathbf{x}})$ . Here the weight  $\|\mathbf{p}_i\|_2^2$  is regarded as the scale of the measurer line which equals to the squared unit arc length from 0 to 1. By using the convexity of  $g(t) = \|\mathbf{p}_i t - \mathbf{x}\|_2^2$ , the orthogonal condition  $(\mathbf{p}_i T_i(\mathbf{x}) - \mathbf{x})^\top \mathbf{p}_i = 0$  is equivalent to finding the nearest point  $T_i(\mathbf{x})$  on the measurer line, namely

$$T_i(\mathbf{x}) = \arg \min_{t \in \mathbb{R}} \|\mathbf{p}_i t - \mathbf{x}\|_2^2. \quad (4)$$

Based on the results of Eq. (3) and Eq. (4), we can clearly observe that the Mahalanobis distance of the data pair  $\{\mathbf{x}, \hat{\mathbf{x}}\}$  is intrinsically computed as *the cumulative arc length between  $\{\mathbf{x}, \hat{\mathbf{x}}\}$ 's nearest points on the learned measurer line  $\mathbf{z} = \mathbf{p}_i t$* , which is shown in Fig. 1. It reveals that linear metrics merely learn the rough directions of real distances, yet cannot capture the complex data geometry.

### 2.2 Model Establishment

As we mentioned before, traditional metrics learn the direction  $\mathbf{p}_i$  of the measurer line  $\mathbf{z} = \mathbf{p}_i t$  in the  $d$ -dimensional sample space. However, such a straight line is far from adapting to complex nonlinear

<sup>1</sup>For calculation details,  $\text{Dist}_{\mathbf{P}}^2(\mathbf{x}, \hat{\mathbf{x}}) = (\mathbf{x} - \hat{\mathbf{x}})^\top \mathbf{P}\mathbf{P}^\top(\mathbf{x} - \hat{\mathbf{x}}) = \|\mathbf{P}^\top(\mathbf{x} - \hat{\mathbf{x}})\|_2^2 = \|(\mathbf{p}_1^\top(\mathbf{x} - \hat{\mathbf{x}}), \mathbf{p}_2^\top(\mathbf{x} - \hat{\mathbf{x}}), \dots, \mathbf{p}_m^\top(\mathbf{x} - \hat{\mathbf{x}}))\|_2^2 = \sum_{i=1}^m (\mathbf{p}_i^\top(\mathbf{x} - \hat{\mathbf{x}}))^2 = \sum_{i=1}^m \|\mathbf{p}_i\|_2^2 \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 \cos^2 \langle \mathbf{p}_i, \mathbf{x} - \hat{\mathbf{x}} \rangle$ .

data in the real world. We thus use a general vector-valued function  $\theta_i: \mathbb{R} \rightarrow \mathbb{R}^d$  to extend the straight line  $\mathbf{z} = \mathbf{p}_i t$  ( $t \in \mathbb{R}$ ) to the smooth curved line  $\tilde{\mathbf{z}} = \theta_i(t)$  ( $t \in \mathbb{R}$ ). Specifically, it can be written as

$$\tilde{\mathbf{z}} = (\tilde{z}_1, \tilde{z}_2, \dots, \tilde{z}_d)^\top = (\theta_{i1}(t), \theta_{i2}(t), \dots, \theta_{id}(t))^\top = \theta_i(t), \quad (5)$$

where the smooth function  $\theta_{ik}(t)$  is the  $k$ -th element of the vector-valued function  $\theta_i(t)$ . It should be noticed that such a curved line is also assumed to be zero-crossing, *i.e.*,  $\theta_i(0) = \mathbf{0}$  which is consistent with the linear distance metric. Then the nearest point  $T_i(\mathbf{x})$  defined in Eq. (4) can be easily extended to the nearest point set  $\mathcal{N}_{\theta_i}(\mathbf{x})$ , and we naturally have that

$$\mathcal{N}_{\theta_i}(\mathbf{x}) = \arg \min_{t \in \mathbb{R}} \|\theta_i(t) - \mathbf{x}\|_2^2. \quad (6)$$

Nevertheless, the point set  $\mathcal{N}_{\theta_i}(\mathbf{x})$  might contain more than one element, so we simply use the smallest element of  $\mathcal{N}_{\theta_i}(\mathbf{x})$ , as described in Definition 1.

**Definition 1.** For a data point  $\mathbf{x} \in \mathbb{R}^d$  and a curved line  $\theta_i$ , we define the calibration point  $\mathcal{T}_{\theta_i}(\mathbf{x})$  as  $\mathcal{T}_{\theta_i}(\mathbf{x}) = \arg \min_{t \in \mathcal{N}_{\theta_i}(\mathbf{x})} t$ , where  $\mathcal{N}_{\theta_i}(\mathbf{x})$  includes all nearest points of  $\mathbf{x}$  on the curved line  $\theta_i(t)$ .

According to our offered interpretation in Section 2.1, the curvilinear distance is consistently regarded as the cumulative arc length values (see Fig. 1(c)). Here we follow the common formula of arc length in calculus [21], which is given in Definition 2.

**Definition 2.** The arc length from  $T_1$  to  $T_2$  on the curved line  $\theta_i$  is defined as

$$\text{Length}_{\theta_i}(T_1, T_2) = \int_{\min(T_1, T_2)}^{\max(T_1, T_2)} \|\theta_i'(t)\|_2 dt, \quad (7)$$

where the derivative vector  $\theta_i'(t) = (\theta_{i1}'(t), \theta_{i2}'(t), \dots, \theta_{id}'(t))^\top$ .

Based on the above definitions of the arc length and the calibration point, the traditional linear distance metric (*i.e.*, Eq. (3)) is easily extended to the general curvilinear form. Specifically, the squared curvilinear distance between data points  $\mathbf{x}$  and  $\hat{\mathbf{x}}$  is calculated by

$$\text{Dist}_{\Theta}^2(\mathbf{x}, \hat{\mathbf{x}}) = \sum_{i=1}^m s_{\theta_i} \cdot \text{Length}_{\theta_i}^2(\mathcal{T}_{\theta_i}(\mathbf{x}), \mathcal{T}_{\theta_i}(\hat{\mathbf{x}})), \quad (8)$$

where  $\Theta = (\theta_1, \theta_2, \dots, \theta_m)$  is the learning parameter of the curvilinear distance metric, and  $m$  is the number of measurer lines. Here the scale value  $s_{\theta_i} = \text{Length}_{\theta_i}^2(0, 1)$ . When we use the empirical risk loss to learn  $\Theta$ , the objective of Curvilinear Distance Metric Learning (CDML) is formulated as

$$\min_{\Theta \in \mathbb{F}_m} \frac{1}{N} \sum_{j=1}^N \mathcal{L}(\text{Dist}_{\Theta}^2(\mathbf{x}_j, \hat{\mathbf{x}}_j); y_j) + \lambda \mathcal{R}(\Theta), \quad (9)$$

where  $\mathbb{F}_m = \{(\theta_1, \theta_2, \dots, \theta_m) \mid \theta_{ik}(t) = 0 \text{ and } \theta_{ik}(t) \text{ is smooth for } i \in \mathbb{N}_m \text{ and } k \in \mathbb{N}_d\}$ , the regularization parameter  $\lambda > 0$  is tuned by users. In the above objective, the loss function  $\mathcal{L}$  evaluates the inconsistency between the predicted distances and their similarity labels, and the regularizer  $\mathcal{R}$  is used to reduce the over-fitting.

**Implementation of  $\Theta$ .** As the learning parameter  $\Theta$  of CDML (*i.e.*, Eq. (9)) is in an abstract form and cannot be directly solved, we have to give a concrete form for each curved line  $\theta_i(t)$  for learning tasks. Here we employ the polynomial function to approximate  $\theta_i(t)$ , due to the guarantee of infinite approximation which is described in Theorem 1. It is notable that  $\theta_i(t)$  can also be infinitely approximated by other ways, such as Fourier series, deep neural networks, and piecewise linear functions [16]. Without loss of generality, we employ the polynomial function in this paper.

**Theorem 1** (Weierstrass Approximation [21]). Assume that the vector-valued function  $\theta_i(t) \in \mathbb{R}^d$  ( $i=1, 2, \dots, m$ ) is continuous and zero-crossing defined on  $[a, b]$ . Then for any  $\epsilon > 0$  and  $t \in [a, b]$ , there exists the  $c$ -order polynomial vector-valued function

$$\mathcal{M}_{i::}(t) = \left( \sum_{k=1}^c \mathcal{M}_{i1k} t^k, \sum_{k=1}^c \mathcal{M}_{i2k} t^k, \dots, \sum_{k=1}^c \mathcal{M}_{idk} t^k \right)^\top, \quad (10)$$

such that  $\sum_{i=1}^m \|\theta_i(t) - \mathcal{M}_{i::}(t)\|_2 < \epsilon$ , where the 3-order tensor  $\mathcal{M} = [\mathcal{M}_{ijk}] \in \mathbb{R}^{m \times d \times c}$ .

We let  $\theta_i(t) := \mathcal{M}_{i::}(t)$  for  $i = 1, 2, \dots, m$ , and then the abstract parameter  $\Theta$  in Eq. (9) can be materialized by the tensor  $\mathcal{M}$  with infinite approximation in the following optimization objective

$$\min_{\mathcal{M} \in \mathbb{R}^{m \times d \times c}} \frac{1}{N} \sum_{j=1}^N \mathcal{L}(\text{Dist}_{\mathcal{M}}^2(\mathbf{x}_j, \hat{\mathbf{x}}_j); y_j) + \lambda \mathcal{R}(\mathcal{M}). \quad (11)$$

We thus successfully convert the abstract optimization problem Eq. (9) to the above concrete form Eq. (11) *w.r.t.* the tensor  $\mathcal{M} \in \mathbb{R}^{m \times d \times c}$ , which can be easily solved by existing algorithms [11].

---

**Algorithm 1** Solving Eq. (11) via Stochastic Gradient Descent.

---

**Input:** Training data pairs  $\mathcal{X} = \{(\mathbf{x}_j, \hat{\mathbf{x}}_j) | j \in \mathbb{N}_N\}$ ; labels  $\mathbf{y} \in \{0, 1\}^N$ ; batch size  $h$ ; learning rate  $\rho$ ; regularization parameter  $\lambda$ ; tensor size parameters  $c$  and  $m$ .

**Initialize:**  $k = 1$ ;  $\mathcal{M}^{(1)} = \mathbf{0}$ .

**Repeat:**

- 1). Uniformly randomly pick  $h$  data pairs  $\{(\mathbf{x}_{b_j}, \hat{\mathbf{x}}_{b_j}) | j \in \mathbb{N}_h\}$  from  $\mathcal{X}$ .
- 2). Compute calibration points  $\mathcal{T}_{\mathcal{M}_{i::}}(\mathbf{x}_{b_j})$  and  $\mathcal{T}_{\mathcal{M}_{i::}}(\hat{\mathbf{x}}_{b_j})$  for  $i = 1, 2, \dots, m$  by solving the real roots of  $f'_i(t) = 0$  in Eq. (13).
- 3). Update the learning parameter  $\mathcal{M}$  by

$$\mathcal{M}^{(k+1)} := \mathcal{M}^{(k)} - \rho \left( \frac{1}{h} \sum_{j=1}^h \mathcal{L}'_j \nabla_{\mathcal{M}^{(k)}} \text{Dist}_{\mathcal{M}^{(k)}}^2(\mathbf{x}_{b_j}, \hat{\mathbf{x}}_{b_j}) + \lambda \nabla_{\mathcal{M}^{(k)}} \mathcal{R}(\mathcal{M}^{(k)}) \right). \quad (12)$$

- 4). Update  $k := k + 1$ .

**Until Converge.**

**Output:** The converged  $\mathcal{M}^*$ .

---

### 2.3 Optimization Algorithm

Since the pair number  $N$  is usually large in Eq. (11), we use the Stochastic Gradient Descent (SGD) method to solve it. As we know that the central operations in SGD are the gradient computation of the objective function. Here we only need to offer the gradient of  $\text{Dist}_{\mathcal{M}}^2(\mathbf{x}, \hat{\mathbf{x}})$  which mainly depends on the calibration points  $\mathcal{T}_{\mathcal{M}_{i::}}(\mathbf{x})$ ,  $\mathcal{T}_{\mathcal{M}_{i::}}(\hat{\mathbf{x}})$ , and the arc length  $\text{Length}_{\mathcal{M}_{i::}}(\mathcal{T}_{\mathcal{M}_{i::}}(\mathbf{x}), \mathcal{T}_{\mathcal{M}_{i::}}(\hat{\mathbf{x}}))$ .

According to Definition 1, the calibration point  $\mathcal{T}_{\mathcal{M}_{i::}}(\mathbf{x})$  can be directly obtained from the nearest point set  $\mathcal{N}_{\mathcal{M}_{i::}}(\mathbf{x}) = \{t^* | f_i(t^*) \leq f_i(\hat{t}), \text{ and } t^*, \hat{t} \in \Gamma_i\}$ , where the polynomial function is

$$f_i(t) = \|\mathcal{M}_{i::}(t) - \mathbf{x}\|_2^2 = \sum_{j,k=1}^c (\mathcal{M}_{i::j}^\top \mathcal{M}_{i::k}) t^{j+k} - 2 \sum_{k=1}^c \mathcal{M}_{i::k}^\top \mathbf{x} t^k + \mathbf{x}^\top \mathbf{x}, \quad (13)$$

and  $\Gamma_i$  is the real root set for polynomial equation  $f'_i(t) = 0$ . Here the real roots of  $f'_i(t) = 0$  can be efficiently solved by simple numerical algorithms [17], of which the computation complexity does not depend on the number of training data pair  $N$  and feature dimensionality  $d$ .

By using the definition of integral, the arc length is equivalently converted to

$$\text{Length}_{\mathcal{M}_{i::}}(T_1, T_2) = \int_{\min(T_1, T_2)}^{\max(T_1, T_2)} \|\mathcal{M}'_{i::}(t)\|_2 dt = \lim_{L \rightarrow \infty} \sum_{l=0}^L \|\mathcal{M}'_{i::}(\min(T_1, T_2) + l\Delta t)\|_2 \Delta t, \quad (14)$$

where  $\Delta t = |T_1 - T_2|/L$ . In practical uses, we fix  $L$  to a large number (e.g.,  $L=10^3$ ) and thus obtain a well approximation  $G_{\mathcal{M}_{i::}}(T_1, T_2) := (\sum_{l=0}^L \|\mathcal{M}'_{i::}(\min(T_1, T_2) + l\Delta t)\|_2 \Delta t)^2$  for the squared arc length value  $\text{Length}_{\mathcal{M}_{i::}}^2(T_1, T_2)$ .

According to the chain rule of derivative, the gradient of  $\text{Dist}_{\mathcal{M}}^2$  w.r.t.  $\mathcal{M}$  is obtained as<sup>2</sup>

$$\begin{aligned} \nabla_{\mathcal{M}_{i::}} \text{Dist}_{\mathcal{M}}^2(\mathbf{x}, \hat{\mathbf{x}}) &= \nabla_{\mathcal{M}_{i::}} G_{\mathcal{M}_{i::}}(0, 1) \cdot G_{\mathcal{M}_{i::}}(\mathcal{T}_{\mathcal{M}_{i::}}(\mathbf{x}), \mathcal{T}_{\mathcal{M}_{i::}}(\hat{\mathbf{x}})) \\ &\quad + G_{\mathcal{M}_{i::}}(0, 1) \cdot \tilde{\nabla}_{\mathcal{M}_{i::}} G_{\mathcal{M}_{i::}}(\mathcal{T}_{\mathcal{M}_{i::}}(\mathbf{x}), \mathcal{T}_{\mathcal{M}_{i::}}(\hat{\mathbf{x}})). \end{aligned} \quad (15)$$

It is noticed that the gradient of  $G_{\mathcal{M}_{i::}}(\mathcal{T}_{\mathcal{M}_{i::}}(\mathbf{x}), \mathcal{T}_{\mathcal{M}_{i::}}(\hat{\mathbf{x}}))$  does not exist necessarily, because the intermediate variable  $\mathcal{T}_{\mathcal{M}_{i::}}(\mathbf{x})$  is not differentiable w.r.t.  $\mathcal{M}_{i::}$ . Therefore, here we use the smoothed gradient<sup>3</sup> instead of the original gradient of  $G_{\mathcal{M}_{i::}}(\mathcal{T}_{\mathcal{M}_{i::}}(\mathbf{x}), \mathcal{T}_{\mathcal{M}_{i::}}(\hat{\mathbf{x}}))$ . The optimization algorithm for solving Eq. (11) is summarized in Algorithm 1.

**Convergence.** Notice that the main difference between Algorithm 1 and traditional SGD is that we utilize a smoothed gradient instead of the original gradient. Previous works have proved that the smoothed gradient still ensures that the SGD algorithm converges to a stationary point [12, 2].

---

<sup>2</sup>  $\nabla_{\mathcal{M}_{i::}} G_{\mathcal{M}_{i::}}(0, 1) = \frac{2}{L} \sqrt{G_{\mathcal{M}_{i::}}(0, 1)} \sum_{l=0}^L \frac{\mathcal{M}'_{i::}((l/L) \cdot ((l/L)^1, (l/L)^2, \dots, (l/L)^c)}{\|\mathcal{M}'_{i::}((l/L)\|_2}$ .

<sup>3</sup>  $\tilde{\nabla}_{\mathcal{M}_{i::}} G_{\mathcal{M}_{i::}}(\mathcal{T}_{\mathcal{M}_{i::}}(\mathbf{x}), \mathcal{T}_{\mathcal{M}_{i::}}(\hat{\mathbf{x}})) = \frac{\Delta \mathcal{M}_{i::}}{\mu \|\Delta \mathcal{M}_{i::}\|_2^2} (G_{\mathcal{M}_{i::} + \mu \Delta \mathcal{M}_{i::}}(\mathcal{T}_{\mathcal{M}_{i::} + \mu \Delta \mathcal{M}_{i::}}(\mathbf{x}), \mathcal{T}_{\mathcal{M}_{i::} + \mu \Delta \mathcal{M}_{i::}}(\hat{\mathbf{x}})) - G(\mathcal{T}_{\mathcal{M}_{i::}}(\mathbf{x}), \mathcal{T}_{\mathcal{M}_{i::}}(\hat{\mathbf{x}})))$ , where  $\Delta \mathcal{M}_{i::} \in \mathbb{R}^{d \times c}$  is generated from the standard normal distribution, and  $\mu > 0$  is a given small number used to smooth the gradient.

### 3 Theoretical Analysis

In this section, we provide theoretical results for the fitting capability, generalization bound, and topological property of CDML. All proofs of theorems are given in the *supplementary materials*.

#### 3.1 Fitting Capability

We first reveal that the curvilinear distance learned by Eq. (11) is capable of distinguishing the similarities of all training data pairs. We assume that the (dis)similar pair sets  $\mathcal{X}_{\text{Similar}}$  and  $\mathcal{X}_{\text{Dissimilar}}$  are the partitions of the whole training data pairs set  $\mathcal{X}$ , where the similarity label  $y_{(\mathbf{x}, \hat{\mathbf{x}})} = 1$  if  $(\mathbf{x}, \hat{\mathbf{x}}) \in \mathcal{X}_{\text{Similar}}$ , and  $y_{(\mathbf{x}, \hat{\mathbf{x}})} = 0$  if  $(\mathbf{x}, \hat{\mathbf{x}}) \in \mathcal{X}_{\text{Dissimilar}}$ . Then the conclusion is described as follows.

**Theorem 2.** For given  $\Delta_{\text{margin}} > 0$ , there exist  $m, c \in \mathbb{N}$  and  $\widetilde{\mathcal{M}} \in \mathbb{R}^{m \times d \times c}$  such that

$$\text{Dist}_{\widetilde{\mathcal{M}}}(\boldsymbol{\beta}, \widehat{\boldsymbol{\beta}}) - \text{Dist}_{\widetilde{\mathcal{M}}}(\boldsymbol{\alpha}, \widehat{\boldsymbol{\alpha}}) > \Delta_{\text{margin}}, \quad (16)$$

where  $(\boldsymbol{\alpha}, \widehat{\boldsymbol{\alpha}}) \in \mathcal{X}_{\text{Similar}}$  and  $(\boldsymbol{\beta}, \widehat{\boldsymbol{\beta}}) \in \mathcal{X}_{\text{Dissimilar}}$ .

From the above result, we know that the well learned curvilinear distance correctly predicts the similarities of data pairs in the training set  $\mathcal{X}$ , which ensures that the inter-class distances are always greater than the intra-class distances. In most metric learning models, the loss functions are designed to reward the larger inter-class distances and smaller intra-class distances. It means that the distance  $\text{Dist}_{\widetilde{\mathcal{M}}}(\cdot, \cdot)$  in Eq. (16) can be successfully achieved by minimizing the loss functions. Therefore, the fitting capability of the curvilinear distance can be reliably guaranteed by the parameter tensor  $\mathcal{M}$ .

#### 3.2 Generalization Bound

Now we further analyze the effectiveness of CDML by offering the generalization bound of the solution to Eq. (11). Such a bound evaluates the bias between the generalization error  $\varepsilon(\mathcal{M}) := \mathbb{E}_{(\mathbf{x}, \hat{\mathbf{x}}) \sim \mathcal{D}}(\mathcal{L}(\text{Dist}_{\mathcal{M}}^2(\mathbf{x}, \hat{\mathbf{x}}); y_{(\mathbf{x}, \hat{\mathbf{x}})}))$  and empirical error  $\bar{\varepsilon}_{\mathcal{X}}(\mathcal{M}) := \frac{1}{N} \sum_{j=1}^N \mathcal{L}(\text{Dist}_{\mathcal{M}}^2(\mathbf{x}_j, \hat{\mathbf{x}}_j); y_j)$ , where  $\mathcal{D}$  is the real data distribution and  $\mathbb{E}(\cdot)$  denotes the expectation function. We simply use the squared tensor Frobenius-norm [18] for regularization and have the following conclusion.

**Theorem 3.** Assume that  $\mathcal{R}(\mathcal{M}) = \|\mathcal{M}\|_{\text{F}}^2 = \sum_{i,j,k} (\mathcal{M}_{ijk})^2$  and  $\mathcal{M}^* \in \mathbb{R}^{m \times d \times c}$  is the solution to Eq. (11). Then, we have that for any  $0 < \delta < 1$  with probability  $1 - \delta$

$$\varepsilon(\mathcal{M}^*) - \bar{\varepsilon}_{\mathcal{X}}(\mathcal{M}^*) \leq X^* \sqrt{2 \ln(1/\delta)/N} + B_{\lambda} R_N(\mathcal{L}), \quad (17)$$

where  $B_{\lambda} \rightarrow 0$  as  $\lambda \rightarrow +\infty^4$ . Here  $R_N(\mathcal{L})$  is the Rademacher complexity of the loss function  $\mathcal{L}$  related to the space  $\mathbb{R}^{m \times d \times c}$  for  $N$  training pairs, and  $X^* = \max_{k \in \mathbb{N}_N} |\mathcal{L}(\text{Dist}_{\mathcal{M}^*}^2(\mathbf{x}_k, \hat{\mathbf{x}}_k); y_k)|$ .

In Eq. (17), the first term of the upper bound converges with the increasing of the number of training data pairs  $N$ . We can also find that the second term converges to 0 with the increasing of  $\lambda$ , which means the regularizer  $\mathcal{R}(\mathcal{M})$  effectively improves the generalization ability of CDML.

#### 3.3 Topological Property

In general topology, the *metric*<sup>5</sup> is defined as the distance function satisfying the non-negativity, symmetry, triangle, and coincidence properties [23, 9]. As an extended *metric*, the *pseudo-metric* merely has the first three properties as revealed in [19]. Here we prove that the curvilinear distance defined in Eq. (8) satisfies the topological definitions, and thus its geometric soundness is guaranteed.

**Theorem 4.** For the curvilinear distance  $\text{Dist}_{\boldsymbol{\Theta}}(\mathbf{x}, \hat{\mathbf{x}})$  and its corresponding parameter  $\boldsymbol{\Theta}$ , we denote  $\boldsymbol{\Theta}'(\boldsymbol{\tau}) = (\boldsymbol{\theta}'_1(\tau_1), \boldsymbol{\theta}'_2(\tau_2), \dots, \boldsymbol{\theta}'_m(\tau_m)) \in \mathbb{R}^{d \times m}$  and have that

- 1).  $\text{Dist}_{\boldsymbol{\Theta}}(\mathbf{x}, \hat{\mathbf{x}})$  is a *pseudo-metric* for any  $\boldsymbol{\Theta} \in \mathbb{F}^m$ ;
- 2).  $\text{Dist}_{\boldsymbol{\Theta}}(\mathbf{x}, \hat{\mathbf{x}})$  is a *metric*, if  $\boldsymbol{\Theta}'(\boldsymbol{\tau})$  is full row rank for any  $\boldsymbol{\tau} = (\tau_1, \tau_2, \dots, \tau_m)^{\top} \in \mathbb{R}^m$ .

<sup>4</sup>Here  $B_{\lambda} = 2\mathbb{E}_{\mathcal{X}, \mathcal{Z}}[\sup_{\mathcal{M} \in \mathcal{F}(\lambda)} \bar{\varepsilon}_{\mathcal{Z}}(\mathcal{M}) - \bar{\varepsilon}_{\mathcal{X}}(\mathcal{M})] / \mathbb{E}_{\mathcal{X}, \mathcal{Z}}[\sup_{\mathcal{M} \in \mathbb{R}^{m \times d \times c}} \bar{\varepsilon}_{\mathcal{Z}}(\mathcal{M}) - \bar{\varepsilon}_{\mathcal{X}}(\mathcal{M})]$  and  $\mathcal{F}(\lambda)$  is a shrinking hypothesis space induced by the regularizer  $\mathcal{R}(\mathcal{M})$ .

<sup>5</sup>The distance function  $\text{Dist}(\cdot, \cdot)$  is a metric if and only if it satisfies the four conditions for any  $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \boldsymbol{\alpha}_3 \in \mathbb{R}^d$ : **(I)**. Non-negativity:  $\text{Dist}(\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2) \geq 0$ ; **(II)**. Symmetry:  $\text{Dist}(\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2) = \text{Dist}(\boldsymbol{\alpha}_2, \boldsymbol{\alpha}_1)$ ; **(III)**. Triangle:  $\text{Dist}(\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2) + \text{Dist}(\boldsymbol{\alpha}_2, \boldsymbol{\alpha}_3) \geq \text{Dist}(\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_3)$ ; **(IV)**. Coincidence:  $\text{Dist}(\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2) = 0 \iff \boldsymbol{\alpha}_1 = \boldsymbol{\alpha}_2$ .

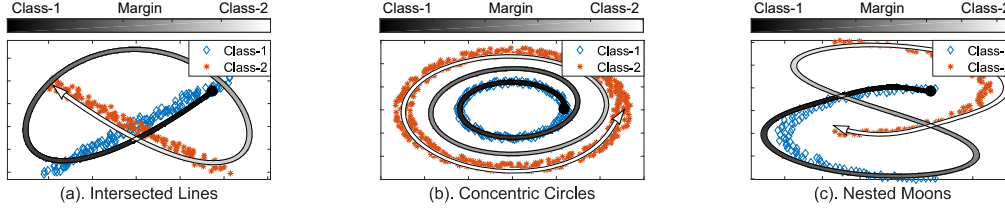


Figure 2: Visualizations of the measurer lines learned by CDML in two-dimensional space. The grayscale denotes the distance from the origin point to the current point of the learned measurer line.

Table 1: Classification error rates (% , mean  $\pm$  std) of all methods on synthetic datasets including *Intersected Lines*, *Concentric Circles*, and *Nested Moons*.

Datasets	LMNN [27]	ITML [7]	Npair [22]	Angular [25]	ODML [28]	CERML [14]	CDML
Intersected Lines	14.33 $\pm$ 1.21	17.46 $\pm$ 2.11	8.52 $\pm$ 0.99	8.10 $\pm$ 3.24	10.52 $\pm$ 2.17	6.21 $\pm$ 1.92	<b>5.12<math>\pm</math>1.13</b>
Concentric Circles	16.62 $\pm$ 2.14	15.92 $\pm$ 3.12	9.13 $\pm$ 1.51	8.98 $\pm$ 1.89	11.31 $\pm$ 2.23	10.32 $\pm$ 2.16	<b>6.95<math>\pm</math>1.41</b>
Nested Moons	17.02 $\pm$ 2.23	12.04 $\pm$ 2.14	9.22 $\pm$ 1.89	10.12 $\pm$ 2.09	15.12 $\pm$ 1.98	11.12 $\pm$ 2.41	<b>8.64<math>\pm</math>2.45</b>

Notice that the above Theorem 4 has the same conclusion with the traditional linear distance metric when  $\theta_i(t)$  is specialized by  $p_i t$ , and thus such a result is a generalization of the property in the linear model [23]. In fact, most of the real-world data indeed have non-negativity, symmetry, triangle, and coincidence properties. Hence this theorem clearly tells us that the basic geometric characteristics of real-world data can be well persevered in the curvilinear distance metric.

## 4 Experimental Results

In this section, we show our experimental results on both synthetic and real-world benchmark datasets to validate the effectiveness of CDML. We first visualize the learning results of CDML on synthetic datasets. After that, we compare classification and verification accuracy of CDML with two classical metric learning methods (LMNN [27] and ITML [7]) and four state-of-the-art methods (Npair loss [22], Angular Loss [25], ODML [28], and CERML [14]). Here LMNN, ITML, and ODML are linear and the others are nonlinear. In our experiments, the parameters  $\lambda$  and  $c$  are fixed to 1.2 and 10, respectively. The SGD parameters  $h$  and  $\rho$  are fixed to  $10^3$  and  $10^{-3}$ , respectively. We follow ITML and use the squared hinge loss and squared Frobenius-norm as  $\mathcal{L}$  and  $\mathcal{R}$  in Eq. (11), respectively.

### 4.1 Visualizations on Synthetic Datasets

To demonstrate the model effectiveness on nonlinear data, we first visualize the learning results of CDML on nonlinear synthetic datasets including *Intersected Lines*, *Concentric Circles*, and *Nested Moons* [5]. Each dataset contains more than 300 data points across 2 classes in the two-dimensional space. On each dataset, 60% of all data is randomly selected for training, and the rest is used for test. The measurer line count  $m$  is fixed to 1 to clearly visualize the learned results.

As illustrated in Fig. 2, the learned measurer lines are plotted with gray lines, of which the gray-level denotes the arc length distance from the origin point to the current point. According to the definition in Eq. (6), we can clearly observe that the nearest points of data points from two classes are distributed apart on the two sides of the measurer lines (*i.e.*, the low gray-level and high gray-level). Therefore, such measure results correctly predict large values for inter-class distances and small values for intra-class distances. Furthermore, the test error rates of all compared methods are shown in Table 1, and we find that DDML, PML, and CDML obtained superior results over other methods due to their non-linearity. Meanwhile, CDML achieves relatively lower error rates than the deep neural network based model (DDML) and manifold based model (PML) on the three datasets, which validates the superiority of our method.

### 4.2 Comparisons on Classification Datasets

To evaluate the performances of all compared methods on the classification task, we adopt the  $k$ -NN classifier ( $k=5$ ) based on the learned metrics to investigate the classification error rates of various methods. The datasets are from the well-known UCI machine learning repository [1] including *MNIST*, *Autompg*, *Sonar*, *Australia*, *Hayes-r*, *Glass*, *Segment*, *Balance*, *Isolet*, and *Letters*.

Table 2: Classification error rates (% , mean  $\pm$  std) of all methods on real-world datasets. The last row lists the Win/Tie/Loss counts of CDML against other methods with  $t$ -test at significance level 0.05.

Datasets	LMNN [27]	ITML [7]	Npair [22]	Angular [25]	ODML [28]	CERML [14]	CDML
MNIST	17.46 $\pm$ 5.32	14.32 $\pm$ 7.32	11.56 $\pm$ 1.07	12.26 $\pm$ 5.12	12.12 $\pm$ 5.22	13.36 $\pm$ 2.32	<b>8.12<math>\pm</math>3.64</b>
Automp	25.92 $\pm$ 3.32	26.62 $\pm$ 3.21	21.95 $\pm$ 1.52	19.02 $\pm$ 3.01	25.32 $\pm$ 5.32	26.36 $\pm$ 3.02	<b>15.32<math>\pm</math>6.11</b>
Sonar	16.04 $\pm$ 5.31	18.02 $\pm$ 3.52	15.31 $\pm$ 2.56	16.86 $\pm$ 1.21	17.95 $\pm$ 6.78	19.21 $\pm$ 6.33	15.40 $\pm$ 3.64
Australia	15.51 $\pm$ 2.53	17.52 $\pm$ 2.13	15.12 $\pm$ 5.11	15.54 $\pm$ 1.23	16.23 $\pm$ 4.12	18.26 $\pm$ 6.22	<b>12.22<math>\pm</math>2.54</b>
Hayes-r	30.46 $\pm$ 7.32	34.24 $\pm$ 6.32	24.36 $\pm$ 2.17	<b>23.12<math>\pm</math>1.37</b>	29.76 $\pm$ 1.07	30.12 $\pm$ 5.32	25.15 $\pm$ 5.23
Glass	30.12 $\pm$ 2.32	29.11 $\pm$ 3.28	22.32 $\pm$ 4.72	23.02 $\pm$ 1.22	28.26 $\pm$ 1.22	29.11 $\pm$ 0.12	<b>22.12<math>\pm</math>4.64</b>
Segment	2.73 $\pm$ 0.82	5.16 $\pm$ 2.22	8.77 $\pm$ 0.32	4.11 $\pm$ 1.22	3.76 $\pm$ 1.34	5.36 $\pm$ 3.12	<b>1.23<math>\pm</math>0.32</b>
Balance	9.93 $\pm$ 1.62	9.31 $\pm$ 2.21	8.12 $\pm$ 1.97	7.12 $\pm$ 2.22	8.63 $\pm$ 2.22	9.45 $\pm$ 5.45	<b>5.01<math>\pm</math>2.64</b>
IsoLet	3.23 $\pm$ 1.32	9.23 $\pm$ 2.32	5.43 $\pm$ 2.12	5.49 $\pm$ 1.12	<b>2.68<math>\pm</math>0.72</b>	7.26 $\pm$ 2.32	3.12 $\pm$ 1.64
Letters	4.21 $\pm$ 2.05	6.24 $\pm$ 0.32	5.14 $\pm$ 1.04	4.67 $\pm$ 1.82	4.88 $\pm$ 0.82	5.32 $\pm$ 2.22	<b>2.09<math>\pm</math>0.64</b>
W/T/L	8/2/0	9/1/0	5/5/0	6/4/0	8/2/0	8/2/0	—

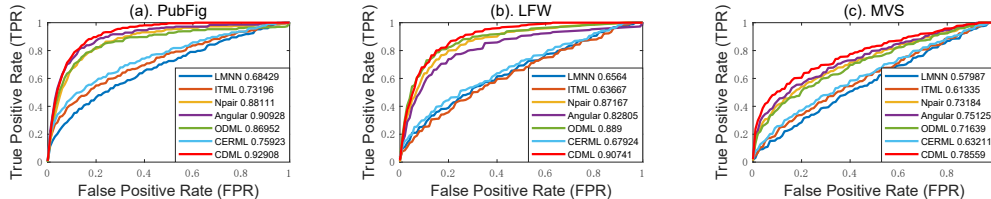


Figure 3: ROC curves of all methods on the 3 datasets. AUC values are presented in the legends.

We compare all methods over 20 random trials. In each trial, 80% of examples are randomly selected as the training examples, and the rest are used for testing. The training pairs are generated by randomly picking up  $1000K(K-1)$  pairs among the training examples [31], where  $K$  is the number of classes. Here the measurer line count  $m$  is fixed to the feature dimensionality (*i.e.*,  $d$ ). The average classification error rates of all compared methods are shown in Table 2. We also perform the  $t$ -test (significance level 0.05) to validate the superiority of our method over all baseline methods on each dataset. From the experimental results, we can observe that CDML obtains significant improvements on the linear metric learning models, which demonstrates the usefulness of our proposed curvilinear generalization. Furthermore, the statistical records of average error rates and  $t$ -test results reliably validate the superiority of our method over other baseline methods.

### 4.3 Comparisons on Verification Datasets

We use two face datasets and one image matching dataset to evaluate the capabilities of all compared methods on image verification. The *PubFig* face dataset includes  $2 \times 10^4$  image pairs belonging to 140 people [15], in which the first 80% data pairs are selected for training and the rest are used for test. Similar experiments are performed on the *LFW* face dataset which includes 13233 unconstrained face images [15]. The *MVS* dataset [3] consists of over  $3 \times 10^4$  image patches sampled from 3D objects, in which  $10^5$  pairs are selected to form the training set, and  $10^4$  pairs are used for test.

The adopted features are extracted by DSIFT [6] and Siamese-CNN [32] for face datasets and image patch dataset, respectively. We plot the Receiver Operator Characteristic (ROC) curve by changing the thresholds of different distance metrics. Then the values of Area Under Curve (AUC) are calculated to quantitatively evaluate the performances of all comparators. From the ROC curves and AUC values in Fig. 3, it is clear to see that DDML and CDML consistently outperform other methods. In comparison, CDML obtains better results than the best baseline method DDML on three datasets.

## 5 Conclusion

In this paper, we introduced the new insight of the mechanism of metric learning models, where the measured distance is naturally interpreted as the arc length between nearest points on the learned straight measurer lines. We extended such straight measurer lines to general curved lines for further learning the intrinsic geometries of the training data. To the best of our knowledge, this is the first work of metric learning with adaptively constructing the geometric relations between data points. We provided theoretical analysis to show that the proposed framework can be well applied to the general nonlinear data. Visualizations on toy data indicate that the learned measurer lines critically capture the underlying rules, and thus making the learning algorithm acquire more reliable and precise metric than the state-of-the-art methods.



## Acknowledgment

S.C., J.Y., and C.G. were supported by the National Science Fund (NSF) of China under Grant (Nos: U1713208, 61602246, and 61973162), Program for Changjiang Scholars, “111” Program AH92005, the Fundamental Research Funds for the Central Universities (No: 30918011319), NSF of Jiangsu Province (No: BK20171430), the “Young Elite Scientists Sponsorship Program” by Jiangsu Province, and the “Young Elite Scientists Sponsorship Program” by CAST (No: 2018QNRC001).

L.L. and H.H. were supported by U.S. NSF-IIS 1836945, NSF-IIS 1836938, NSF-DBI 1836866, NSF-IIS 1845666, NSF-IIS 1852606, NSF-IIS 1838627, and NSF-IIS 1837956.

## References

- [1] Arthur Asuncion and David Newman. Uci machine learning repository, 2007. 4.2
- [2] Krishnakumar Balasubramanian and Saeed Ghadimi. Zeroth-order (non)-convex stochastic optimization via conditional gradient and gradient updates. In *NeurIPS*, 2018. 2.3
- [3] Matthew Brown, Gang Hua, and Simon Winder. Discriminative learning of local image descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1), 2011. 4.3
- [4] Shuo Chen, Chen Gong, Jian Yang, Xiang Li, Yang Wei, and Jun Li. Adversarial metric learning. In *IJCAI*, 2018. 1
- [5] Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *NeurIPS*, 2018. 4.1
- [6] W Cheung and G Hamarneh. n-sift: n-dimensional scale invariant feature transform. *IEEE Transactions on Image Processing*, 18(9), 2009. 4.3
- [7] Jason V Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S Dhillon. Information-theoretic metric learning. In *ICML*, 2007. 1, 4.1, 4, 4.2
- [8] Yueqi Duan, Wenzhao Zheng, Xudong Lin, Jiwen Lu, and Jie Zhou. Deep adversarial metric learning. In *CVPR*, 2018. 1
- [9] Søren Hauberg, Oren Freifeld, and Michael J Black. A geometric take on metric learning. In *NeurIPS*, 2012. 3.3
- [10] Junlin Hu, Jiwen Lu, and Yap-Peng Tan. Discriminative deep metric learning for face verification in the wild. In *CVPR*, 2014. 1
- [11] Feihu Huang, Songcan Chen, and Heng Huang. Faster stochastic alternating direction method of multipliers for nonconvex optimization. In *ICML*, 2019. 2.2
- [12] Feihu Huang, Bin Gu, Zhouyuan Huo, Songcan Chen, and Heng Huang. Faster gradient-free proximal stochastic methods for nonconvex nonsmooth optimization. In *AAAI*, 2019. 2.3
- [13] Zhiwu Huang, Ruiping Wang, Shiguang Shan, and Xilin Chen. Projection metric learning on grassmann manifold with application to video based face recognition. In *CVPR*, 2015. 1
- [14] Zhiwu Huang, Ruiping Wang, Shiguang Shan, Luc Van Gool, and Xilin Chen. Cross euclidean-to-riemannian metric learning with application to face recognition from video. *IEEE transactions on pattern analysis and machine intelligence*, 2018. 1, 4.1, 4, 4.2
- [15] Zhouyuan Huo, Feiping Nie, and Heng Huang. Robust and effective metric learning using capped trace norm: Metric learning via capped trace norm. In *KDD*. ACM, 2016. 4.3
- [16] Niels Landwehr, Mark Hall, and Eibe Frank. Logistic model trees. *Machine learning*, 2005. 2.2
- [17] John M McNamee and Victor Pan. *Numerical methods for roots of polynomials*. Newnes, 2013. 2.3
- [18] Carl D Meyer. *Matrix analysis and applied linear algebra*, volume 71. Siam, 2000. 3.2

- [19] Benjamin Paaßen, Claudio Gallicchio, Alessio Micheli, and Barbara Hammer. Tree edit distance learning via adaptive symbol embeddings. In *ICML*, 2018. 3.3
- [20] Joel B Predd, Sanjeev R Kulkarni, and H Vincent Poor. A collaborative training algorithm for distributed learning. *IEEE Transactions on Information Theory*, 2009. 1
- [21] Walter Rudin et al. *Principles of mathematical analysis*. McGraw-hill New York, 1964. 2.2, 1
- [22] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *NeurIPS*, 2016. 1, 4.1, 4, 4.2
- [23] Juan Luis Suárez, Salvador García, and Francisco Herrera. A tutorial on distance metric learning: Mathematical foundations, algorithms and software. *arXiv:1812.05944*, 2018. 1, 3.3, 3.3
- [24] Fang Wang, Le Kang, and Yi Li. Sketch-based 3d shape retrieval using convolutional neural networks. In *CVPR*, 2015. 1
- [25] Jian Wang, Feng Zhou, Shilei Wen, Xiao Liu, and Yuanqing Lin. Deep metric learning with angular loss. In *ICCV*, 2017. 1, 4.1, 4, 4.2
- [26] Jun Wang, Huyen T Do, Adam Woznica, and Alexandros Kalousis. Metric learning with multiple kernels. In *NeurIPS*, 2011. 1
- [27] Kilian Q Weinberger, John Blitzer, and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. In *NeurIPS*, 2006. 1, 4.1, 4, 4.2
- [28] Pengtao Xie, Wei Wu, Yichen Zhu, and Eric P Xing. Orthogonality-promoting distance metric learning: convex relaxation and theoretical analysis. In *ICML*, 2018. 2.1, 4.1, 4, 4.2
- [29] Jie Xu, Lei Luo, Cheng Deng, and Heng Huang. Bilevel distance metric learning for robust image recognition. In *NeurIPS*, 2018. 1, 2.1
- [30] Han-Jia Ye, De-Chuan Zhan, Xue-Min Si, Yuan Jiang, and Zhi-Hua Zhou. What makes objects similar: a unified multi-metric learning approach. In *NeurIPS*, 2016. 1, 2.1
- [31] Pourya Zadeh, Reshad Hosseini, and Suvrit Sra. Geometric mean metric learning. In *ICML*, 2016. 1, 4.2
- [32] Sergey Zagoruyko and Nikos Komodakis. Learning to compare image patches via convolutional neural networks. In *ICCV*, 2015. 1, 4.3
- [33] Pengfei Zhu, Hao Cheng, Qinghua Hu, Qilong Wang, and Changqing Zhang. Towards generalized and efficient metric learning on riemannian manifold. In *IJCAI*, 2018. 1