Kullback–Leibler Divergence Metric Learning

Shuyi Ji, Zizhao Zhang, Shihui Ying, *Member, IEEE*, Liejun Wang, Xibin Zhao, and Yue Gao, *Senior Member, IEEE*

***Abstract*—The Kullback–Leibler divergence (KLD), which is widely used to measure the similarity between two distributions, plays an important role in many applications. In this article, we address the KLD metric-learning task, which aims at learning the best KLD-type metric from the distributions of datasets. Concretely, first, we extend the conventional KLD by introducing a linear mapping and obtain the best KLD to well express the similarity of data distributions by optimizing such a linear mapping. It improves the expressivity of data distribution, which means it makes the distributions in the same class close and those in different classes far away. Then, the KLD metric learning is modeled by a minimization problem on the manifold of all positive-definite matrices. To deal with this optimization task, we develop an intrinsic steepest descent method, which preserves the manifold structure of the metric in the iteration. Finally, we apply the proposed method along with ten popular metric-learning approaches on the tasks of 3-D object classification and document classification. The experimental results illustrate that our proposed method outperforms all other methods.**

***Index Terms*—3-D object classification, Kullback–Leibler divergence (KLD), metric learning, text classification.**

I. Introduction

DISTRIBUTION is always adopted as a feature of samples in many applications. For example, the Gaussian mixture models (GMMs) [1] are usually used to describe the multidimensional distributions. Therefore, how to measure the similarity between two distributions becomes a core issue in distribution-represented learning methods. In recent years, a number of metrics have been proposed to quantify the similarity between distributions, such as the Jensen–Shannon divergence [2], the Earth mover's distance (EMD) [3], and maximum mean discrepancy [4]. Among these metrics, the Kullback–Leibler divergence (KLD) is one of the most popular and effective metrics because it well explains how one probability distribution diverges from another

$$D_{\mathrm{KL}}(\mathcal{P}||\mathcal{Q}) = \int p(x) \log \frac{p(x)}{q(x)} dx. \tag{1}$$

It has the following desirable properties.

1) *Non-Negativity:* $D_{\mathrm{KL}}(\mathcal{P}||\mathcal{Q}) \geq 0$, with identity if and only if $\mathcal{P} = \mathcal{Q}$.
2) *Convexity:* $D_{\mathrm{KL}}(\mathcal{P}||\mathcal{Q})$ is convex in both $\mathcal{P}$ and $\mathcal{Q}$.
3) *Additivity:* If P_1, P_2, Q_1, are Q_2 are both independent distributions, with the joint distribution $P(x, y) = P_1(x)P_2(y)$, and $Q(x, y) = Q_1(x)Q_2(y)$ likewise, then $D_{\mathrm{KL}}(P\|Q) = D_{\mathrm{KL}}(P_1\|Q_1) + D_{\mathrm{KL}}(P_2\|Q_2)$.

However, the KLD is not a real metric due to its intrinsic asymmetry and the violation of triangle inequality. Interestingly, such commonly recognized deficiency can bring about a series of unique properties when applied in, for example, relative information measurement, as it clearly distinguishes information gain and information loss through asymmetric distances, and further generates a topology in probability distribution space when measuring the similarity of two different distributions. Besides, KLD simultaneously considers the first-order and second-order statistics, thus taking more information into account. From the view of relative information, a reasonable measure ought to preserve three axioms: 1) *locality*; 2) *coordinate invariance*; and 3) *subsystem independence*, and KLD is the only divergence satisfying all these axioms [5], indicating that KLD is arguably the only reasonable measure of relative information.

In practice, KLD has been applied in several applications, such as computer vision [6], information systems [7], document retrieval [8], speech recognition [9], etc. It should be pointed out that most of the existing works focus on using the KLD directly as the similarity metric [10], [11], while some recent works also seek to obtain a more efficient way with a closed form of KLD [12], [13]. Nevertheless, there is a lack of study on the KLD metric itself. In fact, the conventional KLD may not always obtain satisfying performance on measuring the similarity between two distributions because the currently observed distributions may not be the best representation for the data. Therefore, it is highly required to find the best description of distribution to the data under KLD which is to construct a group of KLD-type metrics and then find the optimal one to describe the relation between samples.

Inspired by linear metric-learning methods [14], we propose a KLD metric-learning algorithm in this article. In metric learning, they learn the best linear metric and the distribution of samples can be well described under this metric. On the data

Manuscript received June 23, 2019; revised December 23, 2019 and April 14, 2020; accepted June 25, 2020. Date of publication July 28, 2020; date of current version April 5, 2022. This work was supported in part by the National Natural Science Funds of China under Grant U1701262. This article was recommended by Associate Editor S. Ozawa. *(Corresponding author: Yue Gao.)*

Shuyi Ji, Zizhao Zhang, and Xibin Zhao are with BNRist, KLISS, School of Software, Tsinghua University, Beijing 100084, China (e-mail: jisy19@mails.tsinghua.edu.cn; zhangziz18@mails.tsinghua.edu.cn; zxb@tsinghua.edu.cn).

Shihui Ying is with the Department of Mathematics, Shanghai University, Shanghai 200444, China (e-mail: shying@shu.edu.cn).

Liejun Wang is with the College of Software Engineering, Xinjiang University, Ürümqi 830046, China (e-mail: wljxju@xju.edu.cn).

Yue Gao is with BNRist, KLISS, School of Software, Tsinghua University, Beijing 100084, China, and also with THUIBCS, Tsinghua University, Beijing 100084, China (e-mail: kevin.gaoy@gmail.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCYB.2020.3008248.

Digital Object Identifier 10.1109/TCYB.2020.3008248

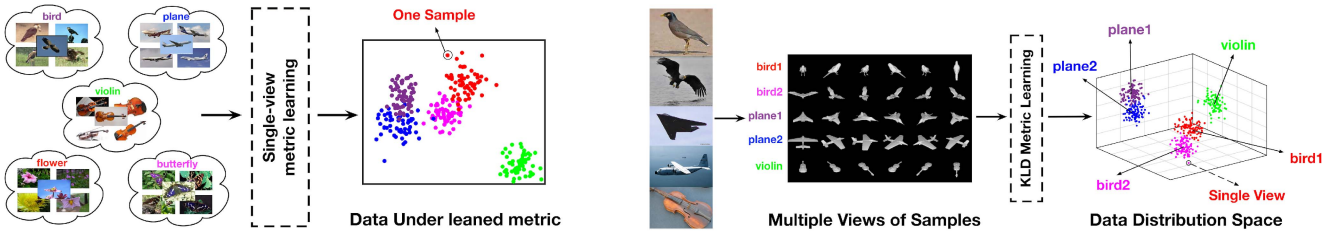

Fig. 1. Comparison between the proposed method and the traditional single-view metric-learning methods. (a) Traditional single-view metric-learning methods. (b) KLD metric learning.

viewpoint, it just finds the best linear mapping that projects the data on the new feature space. In this feature space, the samples from different classes are further away from the samples in the same class. Therefore, in this article, we extend the conventional KLD metric to a group of KLD-type metrics by introducing such linear mapping and then find the best one. Concretely, we first select a group of data pairs of interclass neighbors and inner class neighbors from the labeled data for training, and then the optimal metric is learned to minimize the KLD between the samples belonging to the same class and maximize the KLD between the samples from different classes with respect to the linear mapping. Instead of using the gradient descent method, we deduce an intrinsic Riemannian gradient descent algorithm to retain the positive definiteness of the metric. Such a learned KLD metric achieves better performance on measuring the difference between distributions, that is, makes the data distributions easier to classify.

It is worth noting that there exists a huge difference between the proposed method and the traditional distance metric-learning methods. As depicted in Fig. 1, traditional distance metric-learning methods [e.g., information-theoretic metric learning (ITML)] are often adopted to deal with features from one single view. Under multiview conditions, these methods need special adaptations for learning strategy or feature processing which, to some extent, impedes their wide applications. By comparison, the proposed method can deal with features from multiple views based on the data distributions and, thus, it can better exploit more complementary information from multiple feature representations.

Finally, we apply our KLD learning method on two tasks: 1) 3-D object classification and 2) document classification. Experiments have been conducted on four public benchmarks, including two subsets of the National Taiwan University 3-D model dataset [15], the engineering shape benchmark (ESB) [16], and the Twitter Sentiment Corpus dataset (TWITTER) [17]. The experimental results have demonstrated the outstanding performance of the proposed method compared with the conventional KLD, as well as other state-of-the-art methods.

The contributions of our work can be summarized as follows.

1) We establish a KLD metric-learning model with an intrinsic algorithm, which can dynamically optimize the KLD using the labeled data. The learned metric can better describe the distribution of data, thus making the data distribution easier to be identifiable. Therefore, the learned KLD can perform better in measuring the similarity between data distributions.
2) We apply the proposed KLD metric-learning method on two tasks, that is, 3-D object classification and document classification, and better performance has been witnessed compared with the conventional KLD and other state-of-the-art methods.

The remainder of this article is organized as follows. First, we review the related works on KLD and metric learning in Section II. Then, we introduce the framework of our proposed KLD learning method in Section III and its applications in Section IV. The experimental results for 3-D object classification and document classification are displayed in Section V, and the entire article is concluded and discussed in Section VI.

II. Related Work

In this section, we briefly recall the existing literature on KLD and metric learning.

The KLD, also known as the relative entropy, was first introduced in [18] as the directed divergence between two distributions and has been widely used in various fields, thanks to its outstanding performance in measuring similarity. Most of the applications of KLD focus on employing it directly as a metric to quantify the distance between two probability distributions. For example, Do and Vetterli [10] employed KLD for wavelet-based texture retrieval. In this method, the two related tasks, that is, feature extraction and similarity measurement, are combined into a joint modeling and classification schema, and the joint KLD is used to rank the images in the database. Regarding the nonrigid multimodal image registration task, Guetter *et al.* [19] utilized the KLD to measure the similarity between an observed and a learned joint intensity distribution for images. In the task of document classification [11], each word is first clustered into groups based on the distribution of class labels associated with the word. Then, KLD is employed to measure the similarity between distributions.

Besides, KLD can also be used as the evaluation metric. In [20], the continuous probability distribution of image emotions represented in a valence–arousal space (VA space) is predicted by a multitask-shared sparse regression learning model. In addition, different levels of emotion features are extracted and several baseline algorithms are provided, and KLD is employed to evaluate the performance of different

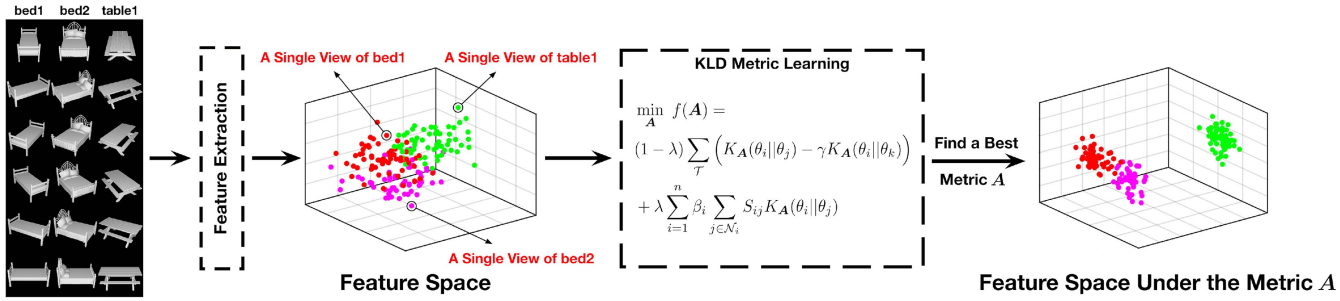

Fig. 2. Framework of our proposed KLD metric-learning model.

methods with different features, in other words, measuring the similarity between the predicted distributions and the ground truth. More specifically, the value of KLD denotes the closeness of the predicted distribution and the ground truth, and the lower value indicates better performance. Rosenberg *et al.* [21] used KLD to estimate the global scene illumination parameters. They utilize the KLD to calculate the closeness of the expected distribution of observed colors over a set of images drawn from some categories of images by assuming a specific set of illumination parameters, and further the actual distribution of observed colors for the specific image.

It should be pointed out that the conventional KLD is commonly calculated on the distributions of data in the Euclidean space. But the description of distribution in the Euclidean space is not always the best. Therefore, metric learning is presented by learning the best metric from the data, under which the description of the data distribution is the best. It is also regarded as finding the best linear mapping, which maps the data into a new feature space. In this feature space, the samples in the different classes are further away from the samples in the same class. For example, a large margin nearest neighbor (LMNN) [22] leverages the local information of the data to learn the linear transform matrix and construct the Mahalanobis distance. ITML [23] combines the information theory with a distance metric and formulates the learning process by a Bregman optimization problem. It aims to learn a Mahalanobis distance function, which can minimize the differential relative entropy between two multivariate Gaussians. However, ITML needs a special adjustment in feature processing or learning procedure in the case of multiview features, which may impede its further applications.

Beyond LMNN and ITML, considerable literature have grown up around the theme of metric learning [24], [25]. Many methods focus on learning the discriminant function from the training image sets. For example, covariance discriminative learning (CDL) [26] models an image set on the Lie group with its natural second-order statistic. Zhu *et al.* [27] extended the original point-to-point distance metric learning to set-to-set distance metric learning (SSDML) by characterizing each sample with the covariance matrix. The hybrid Euclidean-and-Riemannian metric learning (HERML) [28] combines three heterogeneous statistics for robust image set classification. In HERML, both Euclidean and Riemannian metrics are adopted and then hybrid metrics are jointly learned with a discriminant constraint. Lu *et al.* [29] explored holistic multiple statistics of each image set for set representation and further introduced a localized multikernel metric-learning (LMKML) method to learn a distance metric. Nguyen *et al.* [30] proposed a large-margin distance metric-learning (LMDML) model, which leverages the principle of margin maximization to learn a Mahalanobis distance metric. Zhang *et al.* [31] proposed to learn a discriminative ground distance matrix for EMD by alternatively optimizing the EMD metric and the EMD flow network. Regarding the metric-learning loss, by simultaneously considering the relative and absolute distance, Xiao *et al.* [32] introduced a new metric-learning loss with hard sample mining, called margin sample mining loss (MSML). To achieve semantic alignment, Hao *et al.* [33] attempted to align the semantically relevant local regions on images through a relation matrix collecting the distances of each local region pairs and attention technique is adopted to put more weights on the semantically relevant pairs.

Some recent researches also seek to calculate KLD more efficiently. In many fields, such as speech recognition and image classification, the KLD between two GMMs is frequently required. Although KLD has no analytically tractable formula for GMM, there are many approximation methods, such as the Monte Carlo sampling [34], the Gaussian approximation, unscented transformation [35], etc. Olsen and Dharanipragada [12] utilized the variational approximation and the variational upper bound to approximate KLD and further used the model to determine if two acoustic models are similar. To reduce the expensive cost of the naive Monte Carlo sampling methods, Chen *et al.* [36] applied the Taylor expansion, variational approximation, and variational upper bound approximation to accelerate the Monte Carlo sampling methods. Wang *et al.* [37] also modeled the data distribution as GMM and performed discriminative learning on the symmetric positive-definite (SPD) manifold of Gaussians, where a series of metrics including the KLD is studied. Harandi *et al.* [38] studied more general distribution modeling with kernel density estimation and performed learning on the manifold of PDFs. Based on the unscented transform and matching between the Gaussian elements of the two Gaussian mixture densities, Goldberger *et al.* [13] proposed two approximating methods and then utilized them for image retrieval tasks.

It is noted that there is little attention focused on learning an optimal KLD metric toward better discriminative performance. It is exactly the motivation for this work.

TABLE I
DESCRIPTIONS OF IMPORTANT NOTATIONS

Notations	Description
θ_i	the i-th sample
Θ	the set of all labeled samples
x_i^s	the s-th view of the i-th sample
c	the number of classes
Y	the set of labels
N	the number of all labeled samples
n	the number of each sample's view
$\boldsymbol{A}$	the linear mapping
$\mathcal{T}$	the set of sample triplets
μ_i	the mean vector of sample i
σ_i	the covariance matrix of sample i
$K_{\boldsymbol{A}}(\theta_i\|\|\theta_j)$	KLD between θ_i and θ_j under the linear mapping $\boldsymbol{A}$
λ	tradeoff between two terms of the objective function

III. KLD METRIC LEARNING

In this section, we introduce the proposed KLD metric-learning model and algorithm in detail. Fig. 2 illustrates the general framework of our proposed method. Given a set of samples with labels, our goal is to learn an optimal distance metric, which could be more discriminative for classification. That is, the KLD of the samples in the same class should be as small as possible, while for the samples in different classes, the distance should be as large as possible. Below, we first introduce the proposed KLD metric-learning task. Then, an intrinsic gradient descent algorithm is designed to search for the optimal solution of our metric-learning task.

A. Formulations of KLD Metric Learning

Let $\Theta = \{\theta_1, \theta_2, \ldots, \theta_N\}$ be N samples, where each sample denotes the clusters of data points $\{x_i^s | s = 1, \ldots, n,\ 1 \le i \le N\}$, together with labels $Y = \{y_1, y_2, \ldots, y_N\}$. Any $y_i \in Y (1 \le i \le N)$ belongs to the label set $C = \{1, 2, \ldots, c\}$. When it comes to the multiview learning task, each view can be seen as a data point. Each data point is represented by a D-dimensional feature vector. The goal of metric learning is to find an optimal metric, with which data can be easily classified. Table I shows the descriptions of some important notations in this article.

In order to discriminate the samples from different classes, we need to define the distance metric of our model first. As mentioned above, the KLD (i.e., relative entropy) is one of the most widely adopted distance metric, which owns an outstanding performance in measuring the difference between two probability distributions. Specifically, the KLD between two distributions $\mathcal{P}$ and $\mathcal{Q}$ is defined as

$$D_{\mathrm{KL}}(\mathcal{P}||\mathcal{Q}) = E_{\mathcal{P}}\left[\log \frac{\mathcal{P}}{\mathcal{Q}}\right] \tag{2}$$

where E indicates the expectation.

In metric learning, many methods utilize multivariate Gaussians to model data [23], [39]–[42]. For example, in ITML [23], one of the most popular methods in metric learning, the metric-learning problem is formulated as minimizing the differential relative entropy between two multivariate Gaussians under the Mahalanobis distance function constraint. Besides, Nguyen *et al.* [42] intended to learn a linear transformation by maximizing the Jeffrey divergence between two multivariate Gaussian distributions. Inspired by these methods, we model each sample with different views as a Gaussian $g(\theta; \mu, \Sigma)$ to better exploit the complementary information of the multiple feature representations. Specifically, consider two multivariate Gaussians $\mathcal{P}_1$ and $\mathcal{P}_2$ in $\mathbb{R}^d$, the KLD between these two distributions is determined by calculating the expectation as well as utilizing the trace properties as follows:

$$D_{\mathrm{KL}}(\mathcal{P}_1||\mathcal{P}_2) = \frac{1}{2}\left(\log \frac{\det \Sigma_2}{\det \Sigma_1} - n + \mathrm{tr}\left(\Sigma_2^{-1}\Sigma_1\right) + (\mu_2 - \mu_1)^T \Sigma_2^{-1}(\mu_2 - \mu_1)\right) \tag{3}$$

where μ and Σ are the mean vector and the covariance matrix, respectively.

Aiming to better discriminate the samples of different classes, we perform an adaptation by applying a linear mapping $\boldsymbol{A}$ on all of the mean vectors, that is, we replace μ_i with $\boldsymbol{A}\mu_i$.

In addition, to simplify the model, we make a hypothesis that for any two Gaussian distributions p_i and p_j, they should share the same covariance matrix, that is, $\Sigma_i = \Sigma_j$. Then, the KLD between the two rescaled Gaussian distributions $p_i = g(\theta_i; \boldsymbol{A}\mu_i, \Sigma_i)$ and $p_j = g(\theta_j; \boldsymbol{A}\mu_j, \Sigma_j)$ can be easily derived as

$$K_{\boldsymbol{A}}(p_i||p_j) = \frac{1}{2}(\mu_j - \mu_i)^T \boldsymbol{A}^T \Sigma^{-1} \boldsymbol{A}(\mu_j - \mu_i) \tag{4}$$

where Σ represents the identical covariance matrix. Note that $\boldsymbol{A}$ is a positive-definite matrix and is actually a global linear mapping of the underlying space. $K_A(p_i||p_j)$ is the KLD between samples i and j with the linear mapping $\boldsymbol{A}$, which can reflect the similarity between them.

Generally, in metric learning, the distance between interclass data should be as large as possible, while for inner class data, the metric should be minimized. Given a set of sample triplets $\mathcal{T} = \{(i, j, k), y_i = y_j, y_i \ne y_k\}$, usually we should ensure that the triplets satisfy the triplet constraint

$$\mathcal{T} = \left\{(x_i, x_j, x_k) : D_{ij}^2 < D_{ik}^2\right\}$$

which indicates that the distances between the samples from the same class are smaller than those between samples from different classes. Here, $y_i = y_j$ indicates that samples i and j share the same labels, and $y_i \ne y_k$ means that sample i and sample k are with different labels. Based on this triplet constraint, the supervised distance metric learning can be modeled by

$$\min_{\boldsymbol{A}}\ f(\boldsymbol{A}) = \sum_{\mathcal{T}}\left(K_{\boldsymbol{A}}(\theta_i||\theta_j) - K_{\boldsymbol{A}}(\theta_i||\theta_k)\right) \quad \text{s.t. } \boldsymbol{A} \succ 0. \tag{5}$$

The summation of the first term $\sum_{\mathcal{T}} K_A(\theta_i||\theta_j)$ is the sum of KLD of samples from the same class, and the summation of the second term $\sum_{\mathcal{T}} K_A(\theta_i||\theta_k)$ means the total KLD of samples from different classes.

In practice, numerous methods in metric learning leverage a hinge loss function to encode the triplet constraint, for example, LMNN [22]. However, the hinge loss function clearly bears two shortcomings. First, the function is nonsmooth, which requires further special efforts. On the other side, the performance of the function depends heavily on the distribution of the dataset, and thus there is no guarantee that this function can always perform well when dealing with diverse datasets.

To address these two issues, instead of utilizing the hinge loss function, we adopt a positive parameter γ to balance the effect caused by both the homogeneous data and inhomogeneous data. Then, the triplet constraint is rewritten as

$$\mathcal{T}' = \left\{ (x_i, x_j, x_k) : D_{ij}^2 < \gamma D_{ik}^2 \right\}.$$

It is obvious that when $\gamma = 1$, the triplet constraint $\mathcal{T}'$ will degenerate to $\mathcal{T}$. By adjusting this parameter, the distances between the heterogeneous data can be increased while preserving the distances between the homogeneous data. Guided by the distribution of dataset, the parameter is set as $\gamma = (1/(1+\bar{D}^{-1}))$, where $\bar{D}$ is the mean KLD of the dataset $\mathcal{D}$. Clearly, with the distances between the inner class data and interclass data increasing, the parameter γ approaches to 1 since the entropy of a system is large when it is uniform. Therefore, the revised triplet constraint can well describe the characteristic of the classification.

Then, with the revised triplet constraint, we modify the model as follows:

$$\begin{aligned} \min_{\boldsymbol{A}} \ & f(\boldsymbol{A}) = \sum_{\mathcal{T}'} \left(K_{\boldsymbol{A}}(\theta_i || \theta_j) - \gamma K_{\boldsymbol{A}}(\theta_i || \theta_k) \right) \\ \text{s.t.} \ & \boldsymbol{A} \succ 0. \end{aligned} \tag{6}$$

Apart from the revised triplet constraint, a regularization term is also indispensable in our work for preventing overfitting phenomenon, which may sometimes occur in many metric-learning methods, especially when dealing with high-dimensional data [43]. To preserve the local topology structure in the input space, inspired by some recent regularized methods [44]–[46], a regularizer, which is designed and developed based on local topology and represented by local neighbors from the local smoothness, is added to (6).

Then, the KLD metric-learning model is rewritten as

$$\begin{aligned} \min_{\boldsymbol{A}} \ & f(\boldsymbol{A}) = (1-\lambda) \sum_{\mathcal{T}'} \left(K_{\boldsymbol{A}}(\theta_i || \theta_j) - \gamma K_{\boldsymbol{A}}(\theta_i || \theta_k) \right) \\ & \qquad + \lambda \sum_{i=1}^{n} \beta_i \sum_{j \in \mathcal{N}_i} S_{ij} K_{\boldsymbol{A}}(\theta_i || \theta_j) \\ \text{s.t.} \ & \boldsymbol{A} \succ 0. \end{aligned} \tag{7}$$

In the above formula, λ is a tradeoff parameter between the loss term and the regularization that ranges from 0 to 1. n is the number of samples. $\beta_i \in \mathbb{R}^+$ actually indicates the density $p(x_i)$ for input x_i

$$\beta_i = \sum_{j \in \mathcal{N}_i} K_h \left(\frac{X_i - X_j}{h} \right) \tag{8}$$

where K_h is a Gaussian kernel, h is the kernel width that controls the influence of the distance between samples and is usually set as 0.4, and $\mathcal{N}_i$ is the neighborhood index set of the anchor sample x_i. Here, its size $|\mathcal{N}_i|$ is set to 3.

In (7), another important parameter is S_{ij}, which indicates the similarity between two samples. We adopt the Gaussian kernel to calculate S_{ij} as

$$S_{ij} = \exp\left(-D_{ij}^2 / 2\sigma^2 \right) \tag{9}$$

where $\sigma = \min D + 1/v(\max D - \min D)$, $\max D$ and $\min D$ are the maximum and minimum KLD between all samples, and D_{ij} is the KLD between samples i and j. v is set to 10. Note that when calculating S_{ij}, we adopt the conventional KLD and S_{ij} is not updated in the later steps.

B. Optimizations of KLD Metric Learning

In this section, the optimization procedure of the proposed KLD metric-learning model is presented in detail.

We note that the most time-consuming process in our approach is the optimization procedure since in practical applications, the number of all triplets $(\theta_i, \theta_j, \theta_k)$ in the training dataset is very large, thus making it difficult to simultaneously satisfy the constraints for all triplets. Guided by LMNN [22], there is no need to compute pairwise distances between all triplets. Therefore, for each sample θ_i, we just select k_i nearest neighbors with the same label (namely, target neighbors) and k_g nearest neighbors with different labels from each category (namely, imposters) as training data.

Given the training data, here, we optimize the objective function in (7) to learn the optimal KLD metric with respect to the linear mapping $\boldsymbol{A}$. The gradient of the objective function with respect to $\boldsymbol{A}$ is computed by

$$\begin{aligned} \nabla f(\boldsymbol{A}) = & (1-\lambda) \sum_{\mathcal{T}'} \left(\nabla \ K_{\boldsymbol{A}}(\theta_i || \theta_j) - \gamma \nabla \ K_{\boldsymbol{A}}(\theta_i || \theta_k) \right) \\ & + \lambda \sum_i \beta_i \sum_{j \in \mathcal{N}_i} S_{ij} \nabla \ K_{\boldsymbol{A}}(\theta_i || \theta_j) \end{aligned} \tag{10}$$

where $\mathcal{T}'$ is the selected training triplets. By using the trace properties, we can obtain the derivatives of $K_A(\theta_i || \theta_j)$

$$\begin{aligned} \nabla K_{\boldsymbol{A}}(\theta_i || \theta_j) &= \frac{\partial \left(\operatorname{tr} \left(\frac{1}{2} (\mu_j - \mu_i)^T \boldsymbol{A}^T \Sigma_2^{-1} \boldsymbol{A} (\mu_j - \mu_i) \right) \right)}{\partial \boldsymbol{A}} \\ &= \frac{1}{2} \left(\Sigma_2^{-1} \boldsymbol{A} \boldsymbol{P} + \left(\Sigma_2^{-1} \right)^T \boldsymbol{A} \boldsymbol{P}^T \right) \end{aligned} \tag{11}$$

where $\boldsymbol{P} = (\mu_j - \mu_i)(\mu_j - \mu_i)^T$.

To ensure the positive definiteness of the linear mapping $\boldsymbol{A}$, a commonly used optimization technique is the projected gradient method, which applies the gradient descent followed by a projection onto the positive-definite cone [47]. However, in recent years, several intrinsical iterative methods, which are more efficient than the projected gradient methods, have been proposed to solve such optimization problems [48]–[51]. These approaches focus on preserving the manifold structure of positive-definite matrix groups, which means the target optimization variable $\boldsymbol{A}$ still belongs to the corresponding manifolds in each iteration, thus achieving a better convergence

and accuracy. Therefore, to retain the positive definiteness of $\boldsymbol{A}$, we adopt the intrinsic gradient descent algorithm [14] to optimize $\boldsymbol{A}$.

We first let $\mathcal{M}$ denote the set of SPD matrices. It is a smooth Riemannian manifold. Then, the tangent space of $\mathcal{M}$ at the point P is the set of all tangent vectors at point P, denoted by $T_P\mathcal{M}$. Given two points on the manifold, the locally length-minimizing curve between the two points is called the geodesic, which can be seen as the extension of a "straight line." Then, using invariance under congruent transformations, the geodesic $P(t)$ starting from the point P along the direction of tangent vector S is given by

$$P(t) = P^{\frac{1}{2}} \exp\left(tP^{-\frac{1}{2}} S P^{-\frac{1}{2}}\right) P^{\frac{1}{2}}. \tag{12}$$

Hence, the iteration process from the current step $\boldsymbol{A}_t$ to the next step $\boldsymbol{A}_{t+1}$ on the positive-definite matrix group turns to be

$$\boldsymbol{A}_{t+1} = \boldsymbol{A}_t^{\frac{1}{2}} \exp\left(-\alpha \boldsymbol{A}_t^{-\frac{1}{2}} \nabla f(\boldsymbol{A}_t) \boldsymbol{A}_t^{-\frac{1}{2}}\right) \boldsymbol{A}_t^{\frac{1}{2}} \tag{13}$$

where exp is the exponential map and α is the optimal step size in each iteration.

Note that we should guarantee that each $\boldsymbol{A}$ is symmetric and positive definite. However, the nonsymmetry of the objective function in (7) may further lead $\boldsymbol{A}_{t+1}$ to be nonsymmetric. In such a case, a symmetrization operator should be added to the gradient to overcome this issue. That is, we have

$$\boldsymbol{A}_{t+1} = \boldsymbol{A}_t^{\frac{1}{2}} \exp\left(-\frac{1}{2}\alpha \boldsymbol{A}_t^{-\frac{1}{2}} \left(\nabla f(\boldsymbol{A}_t) + \nabla f(\boldsymbol{A}_t)^T\right) \boldsymbol{A}_t^{-\frac{1}{2}}\right) \boldsymbol{A}_t^{\frac{1}{2}}. \tag{14}$$

Summarily, we perform the Riemannian gradient descent on the manifold of the SPD matrices endowed with the affine-invariant Riemannian metric, after having projected the gradient of the cost to the tangent space of this same manifold.

In this way, we use the intrinsic gradient descent method to update the KLD metric $\boldsymbol{A}$ until convergence. The optimal linear mapping $\boldsymbol{A}$ can be used to compute the new KLD for all training data and the new KLD metrics can be further applied to several applications. The overall workflow of KLD metric learning is summarized in Algorithm 1.

Algorithm 1 KLD Metric Learning

Require: Training dataset θ, maximal iteration of KLD metric learning k_m, and parameters k_i, k_g, λ
Ensure: KLD metric $\boldsymbol{A}$
Initialize $\boldsymbol{A} = \boldsymbol{I}$
Calculate conventional KLD between each pair of samples using initial $\boldsymbol{A}$.
Calculate S_{ij} between each pair of samples using conventional KLD.
Construct the training dataset $\mathcal{T}$ by selecting k_i target neighbors and k_g imposters for each sample.
Initialize the learning rate α
for each $t \in [1, k_m]$ **do**
  Compute the newly generated KLD with updated $\boldsymbol{A}$
  Compute the gradient $\nabla f(\boldsymbol{A})$ using Eq.(10)
  Update $\boldsymbol{A}$ by Eq.(14)
end for
  return $\boldsymbol{A}$

IV. Applications of KLD Metric Learning

In this section, we briefly introduce the two applications of the KLD metric-learning model, that is, 3-D object classification and document classification. It is worth noting that the proposed KLD metric-learning model is not limited to these two tasks and can be also used in other applications.

3-D Object Classification: In this task, each object is described by a set of views from different directions. For each view, the multiview convolutional neural network (MVCNN) [52] is adopted to extract features to characterize the 3-D object. Then, aiming to achieve better object representation, all views are grouped to generate the view clusters, following the settings in [53]. In this way, each object can be represented by a group of representative views selected from these clusters and the proposed KLD metric-learning model can be used to learn an optimal KLD distance metric for object classification.

Document Classification: In this task, each document is represented by a bag-of-words (BOW) feature and then a text distribution can be used for document representation [54]. Concretely, first, all nonstop words of documents are embedded into a *word2vec* space [55]. That is, a vector representation for each word is learned using a three-layered neural network, that is, the *word2vec* model. Then, the normalized BOW (nBOW) vector is generated from each document. In this way, we can model the text classification task and the proposed KLD metric-learning method can be applied to it.

Note that our model is generally applicable. In other words, although we use MVCNN and *word2vec* to extract features, our model has no specific requirements for that, which means other features are also plausible.

V. Experiments

To make a thorough comparison against the proposed method and the state of the arts, we conduct experiments on four datasets: two subsets of the National Taiwan University 3-D model dataset (NTU16 and NTU47) [15], the ESB [16], and the Twitter Sentiment Corpus dataset (TWITTER) [17].

A. Datasets

NTU16 and NTU47 Datasets: The NTU16 dataset contains 401 objects belonging to 16 classes and the NTU47 dataset has 549 objects from 47 classes. Each object in NTU16 and NTU47 contains 60 views, where each view is represented by 4096-D MVCNN features. The distinction between NTU16 and NTU47 lies in that NTU47 possesses much more "tiny" classes that consist only two or three objects, thus making the classification problem harder to deal with. Note that to reduce the computational cost of KLD metric learning, principal component analysis (PCA) [56] is utilized first to reduce the feature dimension to 59.

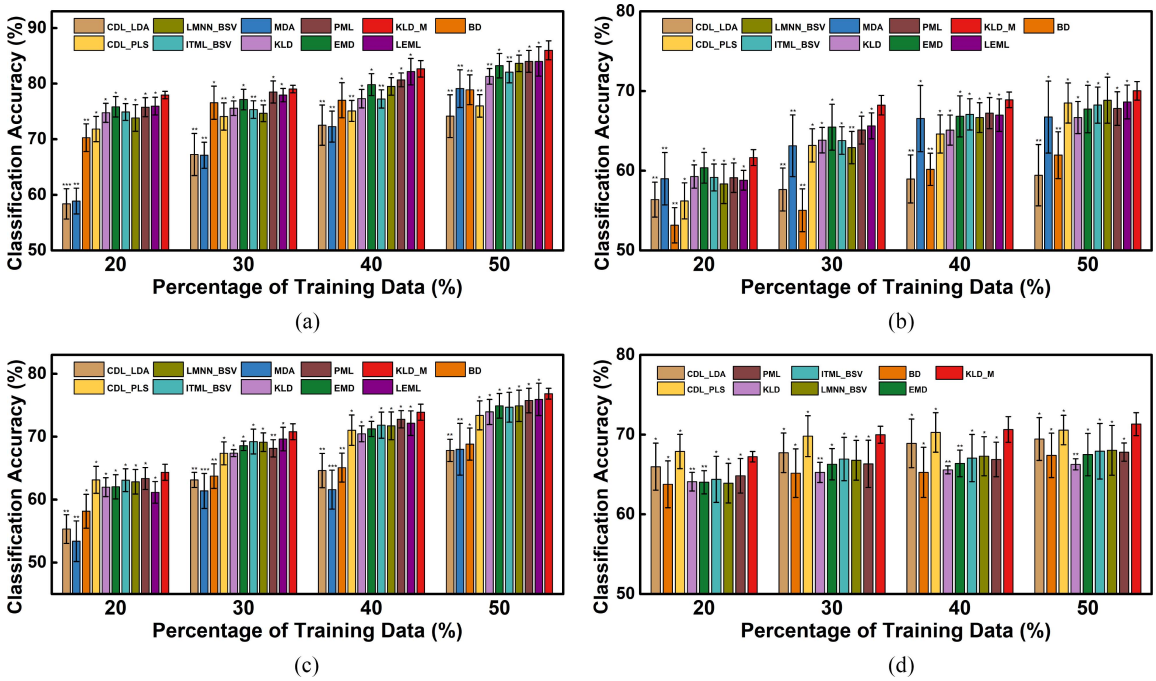

Fig. 3. Performance evaluation by varying the sampling rates. $*p < 0.05$, $**p < 0.01$, and $***p < 0.001$. (a) NTU16 dataset. (b) NTU47 dataset. (c) ESB dataset. (d) TWITTER dataset.

ESB Dataset: It consists mechanical CAD models, including 866 shapes from 43 categories, such as back doors, bracket-like parts, gears, motor bodies, nuts, and thin plates.

TWITTER Dataset: TWITTER contains 3108 tweets with three types of sentiment labels, including "positive," "negative," and "neutral." For each tweet, a 300-D word feature [54] is used as the data representation.

For each sample, all views are involved during the training and testing process.

B. Experimental Settings and Comparing Methods

In all the experiments, 20%, 30%, 40%, and 50% data for each class are randomly selected for training, and the other data are used for testing. For each proportion, we repeat the data splitting process and further conduct experiments ten times, and the average classification accuracy of them is regarded as the true result.

For the proposed method, we perform grid search for hyper-parameters: λ from 0.3 to 0.5, k_i from 2 to 13, and k_g from 1 to 5. Then, λ is set as 0.4 on NTU16, NTU47, and ESB datasets, and 0.35 on the TWITTER dataset. Taking both of the performance and computational cost into account, k_i is set as 7, and k_g is set as 2 for the NTU47 and ESB datasets, and 3 for the NTU16 and TWITTER datasets. As for the comparing methods, the parameter settings are decided according to there original papers and are fixed for all runs.

The following methods are selected for comparison.

1) *CDL With Linear Discriminant Analysis (CDL_LDA) [26]:* In CDL_LDA, the image set is modeled with its natural second-order statistic and a kernel function is derived to map the covariance matrix from the SPD Riemannian manifold to a Euclidean space. LDA learns a discriminant subspace and maps the samples to this subspace.
2) *CDL With Partial Least Squares (CDL_PLSs) [26]:* CDL_PLS is raised from the CDL method by directly learning a regression model between the observed samples and their corresponding class labels.
3) *Manifold Discriminant Analysis (MDA) [57]:* By modeling each image set as a manifold, MDA learns an embedding space with the goal of maximizing the manifold margin, where the local data within each manifold become more compact, while manifolds from different classes are better separated.
4) *Projection Metric Learning (PML) [58]:* PML directly learns the projection metric on the Grassmann manifold with a Fisher LDA-like framework. Data from the original Grassmann manifold are mapped to a more discriminant one for learning.
5) *Log-Euclidean Metric Learning (LEML) [59]:* LEML seeks to learn a tangent map that can transform the matrix logarithms from the original tangent space to a more discriminant tangent space directly.
6) *Information Theory Metric Learning With Best Single View (ITML_BSV) [23]:* ITML formulates the original problem as a Bregman optimization problem and then minimizes the differential relative entropy between two multivariate Gaussians to learn the optimal metric.
7) *LMNN With Best Single View (LMNN_BSV) [22]:* LMNN is one of the most popular metric-learning methods. It leverages the local information of the data to learn the linear transform matrix and construct the Mahalanobis distance.
8) *EMD [3]:* EMD evaluates the dissimilarity between two multidimensional distributions in some feature space

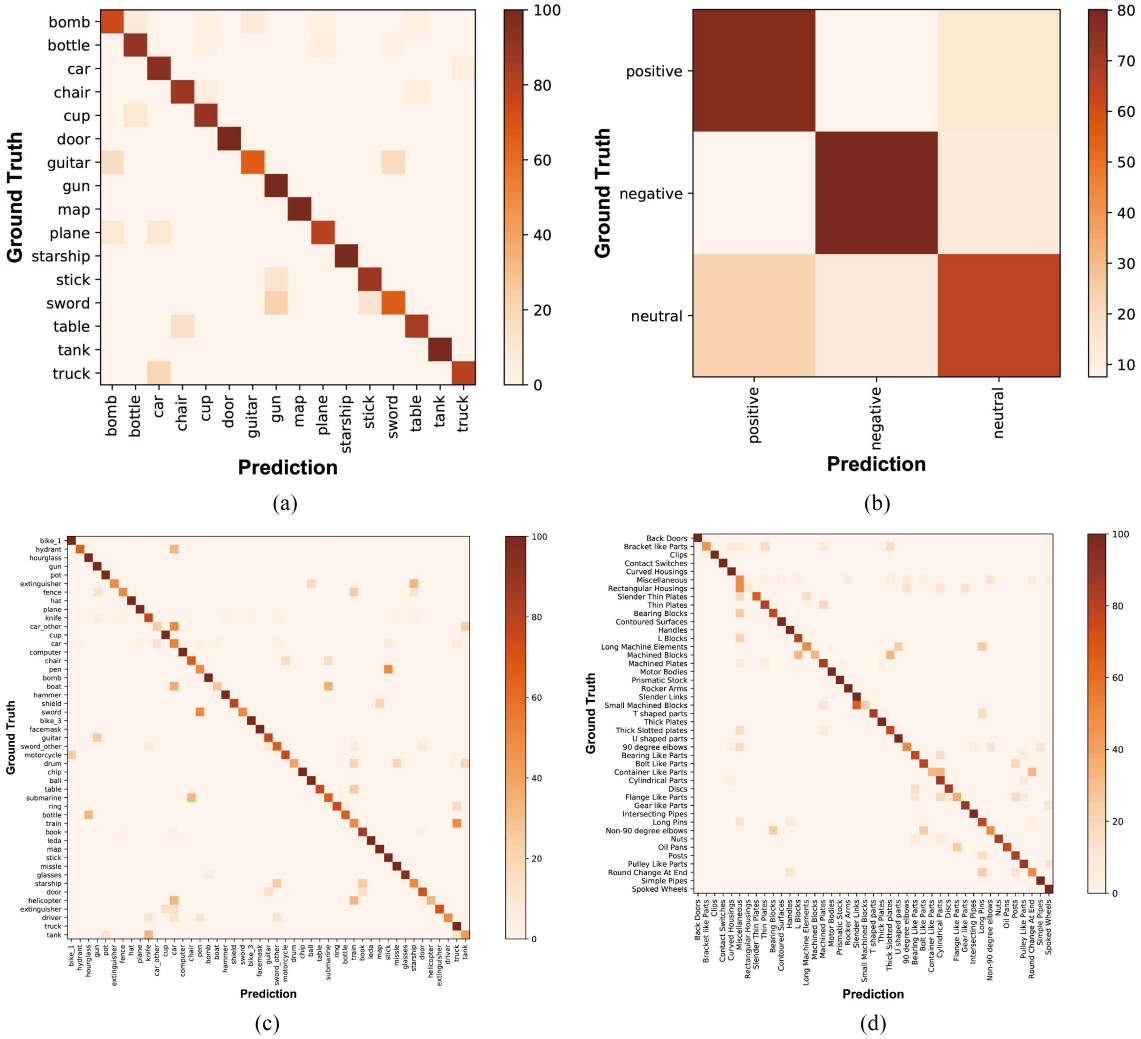

Fig. 4. Confusion matrices for our proposed model on different datasets. (a) NTU16 dataset. (b) TWITTER dataset. (c) NTU47 dataset. (d) ESB dataset.

where a distance measure between single features is given.

9) *Bhattacharyya Distance (BD):* BD can be used to evaluate the relative closeness of two probability distributions.
10) KLD, that is, the conventional KLD method.
11) KLD metric learning (KLD-M), the proposed KLD metric-learning method.

Note that LEML and MDA cannot be performed on TWITTER since duplicate words in TWITTER can generate identical feature space, thus resulting in some mistakes during the training process. So we exclude LEML and MDA when conducting experiments on TWITTER.

C. Experimental Results

The experimental results on four datasets are demonstrated in Fig. 3. In these experiments, the class that a single testing sample belongs to is predicted by its nearest neighborhood. We compare the predictions of all samples with their ground truth for the overall classification accuracy. The results illustrate the following.

1) Among all the comparing methods, the proposed KLD-M achieves the best performance on all four datasets. For example, on the NTU47 dataset, KLD-M achieves the classification accuracy of 67.99% when 30% data are used for training, and compared with PML, CDL_LDA, CDL_PLS, BD, EMD, MDA, LEML, ITML_BSV, and LMNN_BSV, gains are 3.12%, 10.58%, 5.05%, 13.18%, 2.76%, 4.10%, 2.60%, 5.32%, and 4.44%, respectively.
2) Compared with the conventional KLD method, the proposed KLD-M method achieves superior classification performance on all four datasets. For example, when applied on the TWITTER dataset with 20%, 30%, 40%, and 50% data used for training, KLD-M obtains gains of 3.66%, 4.79%, 5.32%, and 5.51%.

Fig. 4 shows confusion matrices of the proposed model on different datasets. The samples classified correctly are displayed on the diagonal. The shade of the color bar indicates the classification proportion accordingly. In other words, the darker the color, the larger portion that the corresponding block occupies in its row. Results show that the KLD-M can conduct accurate classifications for most categories while being misguided by some confusing samples. For example, on the TWITTER dataset, neutral tweets may sometimes be judged as

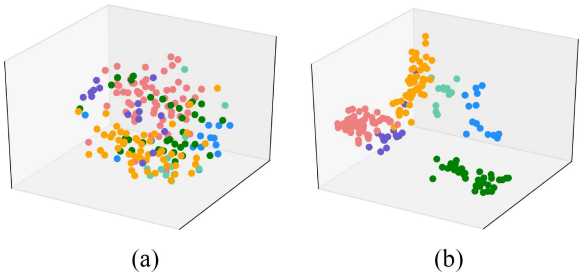

Fig. 5. Visualization of data feature space. (a) Original feature space. (b) Feature space under learned metric.

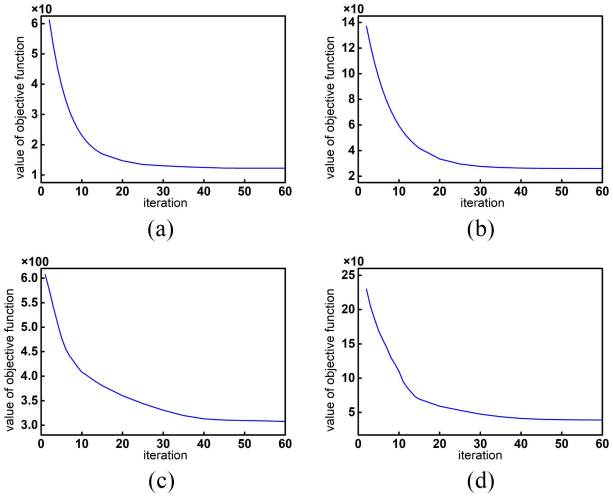

Fig. 6. Variation of the objective function value in our learning method. In these results, 30% data are used for training. (a) NTU16 dataset. (b) NTU47 dataset. (c) TWITTER dataset. (d) ESB dataset.

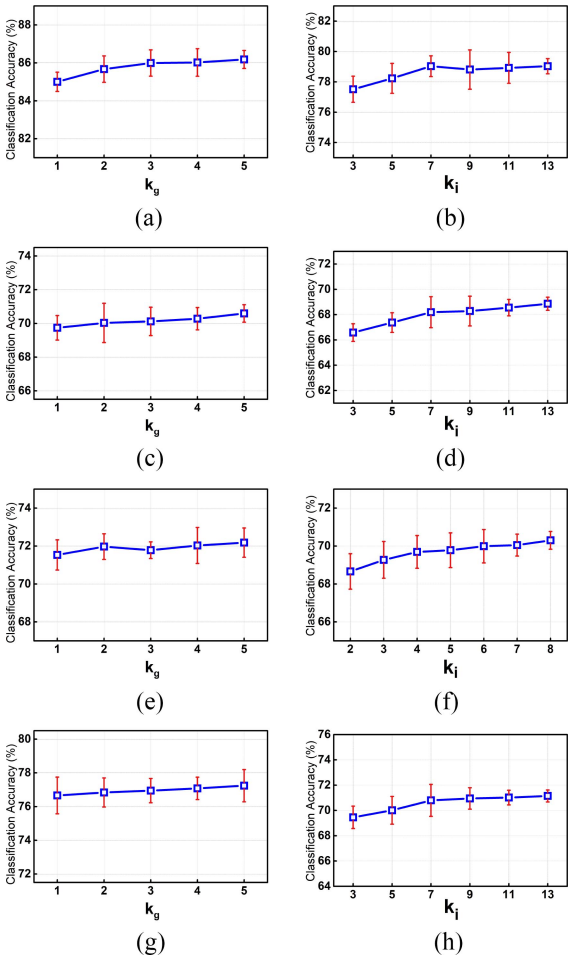

Fig. 7. Performance evaluation by varying the parameters k_g and k_i with 30% data used for training, respectively. (a) and (b) NTU16 dataset. (c) and (d) NTU47 dataset. (e) and (f) TWITTER dataset. (g) and (h) ESB dataset.

positive ones. Moreover, the pen may occasionally be mistaken for stick by the model, which can be difficult to discriminate even for human beings.

Given the high dimensions and numerous categories of the raw data, it is quite difficult to present a distinct visualization of the entire dataset. Thereby, we randomly select six classes from NTU16 and further project them to a 3-D space with the PCA method, as depicted in Fig. 5. It is obvious that the separability of data is improved under our learned metric compared with the original data distribution.

The reason why the proposed method outperforms the conventional KLD lies in that, for conventional KLD, the similarity between two distributions is directly measured in the Euclidean space. Therefore, if the raw distributions are not discriminative for data representation, the conventional KLD may yield inferior performance. In contrast, the KLD-M method learns the best linear mapping and maps the original data into a new feature space before measuring the similarity in order to assure the distributions are discriminative enough, thus achieving outstanding performance.

D. On Convergence

We have demonstrated the variation of the objective function value during the learning process on four datasets in Fig. 6. As depicted in Fig. 6, the value of the objective function reduces rapidly, which indicates the efficiency of the proposed method.

E. On Parameter

There are several key parameters in our framework, including λ in (7), the size of the neighborhood set ($\mathcal{N}_i$) of the anchor sample x_i, and the number of selected training data (k_i and k_g). We further test the sensitivity of these key parameters with other parameters fixed.

For the training data selection, the number of selected target neighbors and imposters for each data is determined by parameters k_i and k_g, respectively. Intuitively, more training triplets will contribute to better performance while a higher training computational cost; while fewer training triplets will result in the opposite. Thus, there exists a tradeoff between the scale of training data and performance. Here, we fix k_g as 2 and vary k_i between 3 and 13 on the NTU16 dataset, and 2 and 8 on the TWITTER dataset. On the NTU47 dataset, k_g is fixed as 3 and k_i ranges among $[3, 13]$. Then, we fix k_i as 7 and vary k_g between 1 and 5 on all the datasets. All the experiments are conducted with 30% training data.

The experimental results are demonstrated in Fig. 7, from which we can discover when k_i is large enough (>7), the

TABLE II
COMPARISONS OF RUNNING TIME (SECONDS) OF ALL COMPARED METHODS ON FOUR DATASETS WITH 30% DATA USED FOR TRAINING

	PML	LEML	CDL_LDA	CDL_PLS	MDA	EMD	BD	ITML_BSV	LMNN_BSV	KLD	KLD-M
NTU16	3.95	4.82	1.36	1.53	15.19	1.96	2.68	123.48	23.84	2.13	5.84
NTU47	5.06	6.06	2.28	2.30	23.94	3.43	4.22	173.96	39.08	3.97	9.17
ESB	18.74	14.90	2.03	3.95	24.22	9.98	16.73	368.08	70.37	12.52	78.5
TWITTER	1563.27	/	647.92	650.284	/	128.61	189.83	4728.75	202.34	157.56	1298.05

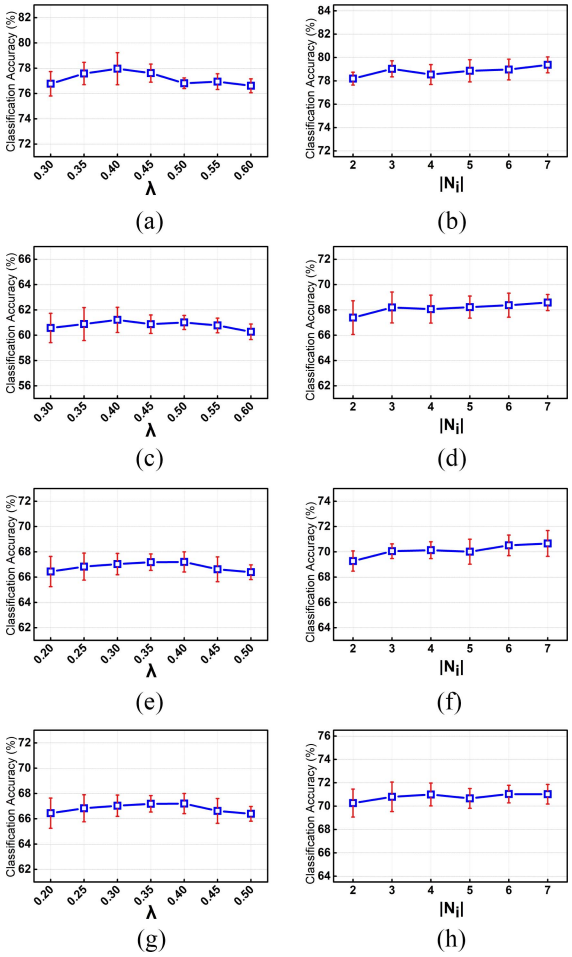

Fig. 8. Performance evaluation by varying parameters λ and $|\mathcal{N}_i|$ with 20% and 30% data used for training, respectively. (a) and (b) NTU16 dataset. (c) and (d) NTU47 dataset. (e) and (f) TWITTER dataset. (g) and (h) ESB dataset.

performance is steady, which demonstrates the robustness of the proposed method toward the selection of k_i. Another interesting insight is that increasing k_g can hardly improve the performance yet incur heavy computational cost. In other words, a small k_g is fully capable of providing a sufficiently "pure" neighborhood of the inquiry input.

Another important parameter is λ, which balances the loss term and the regularizer and needs tuning under different experimental settings. Fig. 8(a), (c), (e), and (g) illustrates the classification accuracy when the parameter ranges from 0.3 to 0.6 on NTU16, NTU47, and ESB, and from 0.2 to 0.5 on TWITTER, from which we can observe that the performances are stable when λ varies. Also, Fig. 8(b), (d), (f), and (h) depicts the influence of different $|\mathcal{N}_i|$ selections on

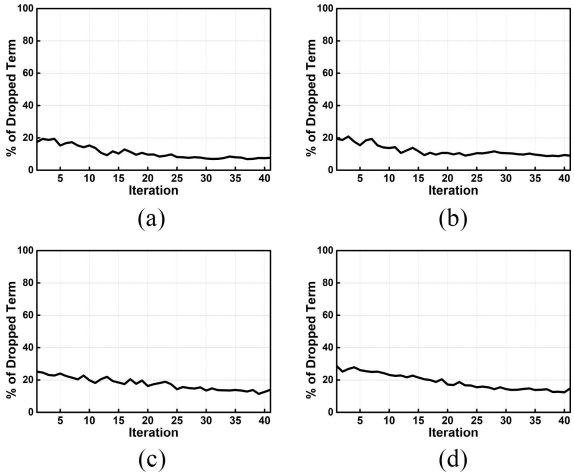

Fig. 9. Percentage of dropped term in loss. (a) NTU16 dataset. (b) NTU47 dataset. (c) ESB dataset. (d) TWITTER dataset.

the performance. We vary $|\mathcal{N}_i|$ from 2 to 7 on four datasets, and the results demonstrate that the proposed method is not sensitive to the selection of $|\mathcal{N}_i|$.

F. On Running Time

The running time of different approaches is shown in Table II. On the NTU16 and NTU47 datasets, KLD-M achieves a comparable efficiency against LEML and PML, while greatly outperforms MDA, ITML_BSV, and LMNN_BSV. On the ESB and TWITTER datasets, KLD-M still yields a satisfying accuracy, while it owns a relatively low efficiency compared with those on the NTU16 and NTU47 datasets, probably due to the fact that scales of training samples in the ESB and TWITTER datasets are much larger than those of the above two datasets.

G. On Model Simplification

Broadly speaking, the more complex the model is, the worse generalization ability the model possesses. Recalling that in Section III, in order to simplify the model and avoid the overfitting problem, we make a hypothesis that any two Gaussian distributions share the uniform covariance. To some extent, such simplification is equivalent to dropping the higher order term from the original expressions and can be regarded as an approximation to the traditional KLD. Fig. 9 depicts the percentage that the dropped term shares in the loss, from which we can draw the conclusion that in our model, the main component of loss is the reserved term. In future work, we will take a step forward by exploring the higher-order nonlinear model without such a hypothesis.

VI. Conclusion

In this article, we proposed a KLD metric-learning model. This method was capable of learning an optimal KLD metric and obtained a better distance measurement by considering the labeled information. The learned KLD metric was able to minimize the KLD between the data in the same class while maximizing the KLD between the data from different classes, thus making the original data distributions easier to be identifiable. We also developed an intrinsic steepest descent method to solve the optimization problem. Finally, we applied the proposed method on the tasks of 3-D object classification and document classification. Experiments on four public datasets demonstrated the effectiveness and robustness of the proposed method compared with the state-of-the-art methods. The framework of our proposed method can be also further applied to many other applications, such as image retrieval and speech recognition.

References

[1] D. Reynolds, "Gaussian mixture models," in *Encyclopedia of Biometrics*. Heidelberg, Germany: Springer, 2015, pp. 827–832.

[2] D. M. Endres and J. E. Schindelin, "A new metric for probability distributions," *IEEE Trans. Inf. Theory*, vol. 49, no. 7, pp. 1858–1860, Jul. 2003.

[3] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *Int. J. Comput. Vis.*, vol. 40, no. 2, pp. 99–121, 2000.

[4] K. M. Borgwardt, A. Gretton, M. J. Rasch, H.-P. Kriegel, B. Schölkopf, and A. J. Smola, "Integrating structured biological data by kernel maximum mean discrepancy," *Bioinformatics*, vol. 22, no. 14, pp. e49–e57, 2006.

[5] A. Caticha, "Relative entropy and inductive inference," in *Proc. AIP Conf.*, vol. 707, 2004, pp. 75–96.

[6] J. Inglada, "Change detection on SAR images by using a parametric estimation of the Kullback–Leibler divergence," in *Proc. IEEE Geosci. Remote Sens. Symp.*, vol. 6, 2003, pp. 4104–4106.

[7] F. Pérez-Cruz, "Kullback–Leibler divergence estimation of continuous distributions," in *Proc. IEEE Int. Symp. Inf. Theory*, 2008, pp. 1666–1670.

[8] A. Huang, "Similarity measures for text document clustering," in *Proc. New Zealand Comput. Sci. Res. Student Conf.*, 2008, pp. 49–56.

[9] P. J. Moreno, P. P. Ho, and N. Vasconcelos, "A Kullback–Leibler divergence based kernel for SVM classification in multimedia applications," in *Proc. Adv. Neural Inf. Process. Syst.*, 2004, pp. 1385–1392.

[10] M. N. Do and M. Vetterli, "Wavelet-based texture retrieval using generalized Gaussian density and Kullback–Leibler distance," *IEEE Trans. Image Process.*, vol. 11, no. 2, pp. 146–158, Feb. 2002.

[11] L. D. Baker and A. K. McCallum, "Distributional clustering of words for text classification," in *Proc. ACM Special Interest Group Inf. Retrieval*, 1998, pp. 96–103.

[12] P. A. Olsen and S. Dharanipragada, "An efficient integrated gender detection scheme and time mediated averaging of gender dependent acoustic models," in *Proc. Eur. Conf. Speech Commun. Technol.*, 2003, pp. 139–142.

[13] J. Goldberger, S. Gordon, and H. Greenspan, "An efficient image similarity measure based on approximations of KL-divergence between two Gaussian mixtures," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2003, p. 487.

[14] 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. 29, no. 7, pp. 2731–2742, Jul. 2018.

[15] D.-Y. Chen, X.-P. Tian, Y.-T. Shen, and M. Ouhyoung, "On visual similarity based 3D model retrieval," *Comput. Graph. Forum*, vol. 22, no. 3, pp. 223–232, 2003.

[16] S. Jayanti, Y. Kalyanaraman, N. Iyer, and K. Ramani, "Developing an engineering shape benchmark for CAD models," *Comput.-Aided Design*, vol. 38, no. 9, pp. 939–953, 2006.

[17] N. J. Sanders, *Sanders–Twitter Sentiment Corpus*, Sanders Anal. LLC, Cordova, TN, USA, 2011.

[18] S. Kullback, *Information Theory and Statistics*. Chelmsford, MA, USA: Courier Corporat., 1951.

[19] C. Guetter, C. Xu, F. Sauer, and J. Hornegger, "Learning based non-rigid multi-modal image registration using Kullback–Leibler divergence," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2005, pp. 255–262.

[20] S. Zhao, H. Yao, Y. Gao, R. Ji, and G. Ding, "Continuous probability distribution prediction of image emotions via multitask shared sparse regression," *IEEE Trans. Multimedia*, vol. 19, no. 3, pp. 632–645, Mar. 2017.

[21] C. Rosenberg, M. Hebert, and S. Thrun, "Color constancy using KL-divergence," in *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 1, 2001, pp. 239–246.

[22] 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, Feb. 2009.

[23] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, "Information-theoretic metric learning," in *Proc. ACM Int. Conf. Mach. Learn.*, 2007, pp. 209–216.

[24] P. Moutafis, M. Leng, and I. A. Kakadiaris, "An overview and empirical comparison of distance metric learning methods," *IEEE Trans. Cybern.*, vol. 47, no. 3, pp. 612–625, Mar. 2017.

[25] J. Xie, G. Dai, F. Zhu, L. Shao, and Y. Fang, "Deep nonlinear metric learning for 3-D shape retrieval," *IEEE Trans. Cybern.*, vol. 48, no. 1, pp. 412–422, Jan. 2018.

[26] R. Wang, H. Guo, L. S. Davis, and Q. Dai, "Covariance discriminative learning: A natural and efficient approach to image set classification," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, 2012, pp. 2496–2503.

[27] P. Zhu, L. Zhang, W. Zuo, and D. Zhang, "From point to set: Extend the learning of distance metrics," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 2664–2671.

[28] Z. Huang, R. Wang, S. Shan, and X. Chen, "Hybrid Euclidean-and-Riemannian metric learning for image set classification," in *Proc. Asian Conf. Comput. Vis.*, 2014, pp. 562–577.

[29] J. Lu, G. Wang, and P. Moulin, "Image set classification using holistic multiple order statistics features and localized multi-kernel metric learning," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 329–336.

[30] B. Nguyen, C. Morell, and B. De Baets, "Scalable large-margin distance metric learning using stochastic gradient descent," *IEEE Trans. Cybern.*, vol. 50, no. 3, pp. 1072–1083, Mar. 2020.

[31] Z. Zhang, Y. Zhang, X. Zhao, and Y. Gao, "EMD metric learning," in *Proc. Assoc. Adv. Artif. Intell. Conf.*, 2018, pp. 4490–4497.

[32] Q. Xiao, H. Luo, and C. Zhang, "Margin sample mining loss: A deep learning based method for person re-identification," 2017. [Online]. Available: arXiv:1710.00478.

[33] F. Hao, F. He, J. Cheng, L. Wang, J. Cao, and D. Tao, "Collect and select: Semantic alignment metric learning for few-shot learning," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 8460–8469.

[34] R. Y. Rubinstein and D. P. Kroese, *Simulation and the Monte Carlo Method*. New York, NY, USA: Wiley, 2016.

[35] S. J. Julier and J. K. Uhlmann, "A general method for approximating nonlinear transformations of probability distributions," Robot. Res. Group, Dept. Eng. Sci., Univ. of Oxford, Oxford, U.K., Rep., 1996. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.46.6718

[36] J.-Y. Chen, J. R. Hershey, P. A. Olsen, and E. Yashchin, "Accelerated Monte Carlo for Kullback–Leibler divergence between Gaussian mixture models," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2008, pp. 4553–4556.

[37] W. Wang, R. Wang, Z. Huang, S. Shan, and X. Chen, "Discriminant analysis on Riemannian manifold of Gaussian distributions for face recognition with image sets," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 2048–2057.

[38] M. Harandi, M. Salzmann, and M. Baktashmotlagh, "Beyond gauss: Image-set matching on the Riemannian manifold of PDFs," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 4112–4120.

[39] Y. Xu, W. Ping, and A. T. Campbell, "Multi-instance metric learning," in *Proc. IEEE Int. Conf. Data Min.*, 2011, pp. 874–883.

[40] N. Shimizu, M. Hagiwara, Y. Ogawa, K. Toyama, and H. Nakagawa, "Metric learning for synonym acquisition," in *Proc. Int. Conf. Comput. Linguist.*, 2008, pp. 793–800.

[41] S. Wang and R. Jin, "An information geometry approach for distance metric learning," in *Proc. Artif. Intell. Stat.*, 2009, pp. 591–598.

[42] B. Nguyen, C. Morell, and B. De Baets, "Supervised distance metric learning through maximization of the Jeffrey divergence," *Pattern Recognit.*, vol. 64, pp. 215–225, Apr. 2017.

[43] B. Kulis *et al.*, "Metric learning: A survey," *Found. Trends Mach. Learn.*, vol. 5, no. 4, pp. 287–364, 2013.
[44] Q. Wang, P. C. Yuen, and G. Feng, "Semi-supervised metric learning via topology preserving multiple semi-supervised assumptions," *Pattern Recognit.*, vol. 46, no. 9, pp. 2576–2587, 2013.
[45] S. C. Hoi, W. Liu, and S.-F. Chang, "Semi-supervised distance metric learning for collaborative image retrieval and clustering," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 6, no. 3, p. 18, 2010.
[46] D. Cai, X. He, and J. Han, "Semi-supervised discriminant analysis," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2007, pp. 1–7.
[47] J. Han, G. Cheng, Z. Li, and D. Zhang, "A unified metric learning-based framework for co-saliency detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 10, pp. 2473–2483, Oct. 2018.
[48] R. Arora, "On learning rotations," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 55–63.
[49] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache, "Geometric means in a novel vector space structure on symmetric positive-definite matrices," *SIAM J. Matrix Anal. Appl.*, vol. 29, no. 1, pp. 328–347, 2007.
[50] M. Moakher, "A differential geometric approach to the geometric mean of symmetric positive-definite matrices," *SIAM J. Matrix Anal. Appl.*, vol. 26, no. 3, pp. 735–747, 2005.
[51] S. Ying, H. Qin, Y. Peng, and Z. Wen, "Compute Karcher means on SO(n) by the geometric conjugate gradient method," *Neurocomputing*, vol. 215, pp. 169–174, Nov. 2016.
[52] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3D shape recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 945–953.
[53] Y. Gao *et al.*, "Camera constraint-free view-based 3-D object retrieval," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 2269–2281, Apr. 2012.
[54] M. Kusner, Y. Sun, N. Kolkin, and K. Weinberger, "From word embeddings to document distances," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 957–966.
[55] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013. [Online]. Available: arXiv:1301.3781.
[56] I. Jolliffe, "Principal component analysis," in *International Encyclopedia of Statistical Science*. New York, NY, USA: Springer, 2011, pp. 1094–1096.
[57] R. Wang and X. Chen, "Manifold discriminant analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern*, 2009, pp. 429–436.
[58] Z. Huang, R. Wang, S. Shan, and X. Chen, "Projection metric learning on Grassmann manifold with application to video based face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 140–149.
[59] Z. Huang, R. Wang, S. Shan, X. Li, and X. Chen, "Log-Euclidean metric learning on symmetric positive definite manifold with application to image set classification," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 720–729.

Shuyi Ji received the B.E. degree from the School of Software, Tsinghua University, Beijing, China, in 2019, where she is currently pursuing the Ph.D. degree in machine learning with the School of Software.

Zizhao Zhang received the B.E. degree in machine learning and computer vision from the School of Software, Tsinghua University, Beijing, China, in 2018, where she is currently pursuing the Ph.D. degree in machine learning and computer vision.

Shihui Ying (Member, IEEE) received the B.Eng. degree in mechanical engineering and the Ph.D. degree in applied mathematics from Xi'an Jiaotong University, Xi'an, China, in 2001 and 2008, respectively.

He is currently a Professor with the Department of Mathematics, School of Science, Shanghai University, Shanghai, China. He held a postdoctoral position with the Biomedical Research Imaging Center, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, from 2012 to 2013. His current research interests include mathematical theory and methods for medical imaging and image analysis.

Liejun Wang received the Ph.D. degree from the School of Information and Communication Engineering, Xi'an Jiaotong University, Xi'an, China, in 2012.

He is currently a Professor with Xinjiang University, Ürümqi, China. His current research interests include computer vision, natural language processing, and wireless sensor.

Xibin Zhao received the B.S., M.E., and Ph.D. degrees in reliability analysis of hybrid network systems and information system security from the School of Computer Science and Telecommunication Engineering, Jiangsu University, Zhenjiang, China, in 1994, 2000, and 2004, respectively.

He is currently an Associate Professor with the School of Software, Tsinghua University, Beijing, China. His research interests include reliability analysis of hybrid network systems and information system security.

Yue Gao (Senior Member, IEEE) received the B.S. degree from the Harbin Institute of Technology, Harbin, China, in 2005, and the M.E. and Ph.D. degrees from Tsinghua University, Beijing, China, in 2008 and 2012, respectively.

He is currently an Associate Professor with the School of Software, Tsinghua University. He has also been working with the School of Computing, National University of Singapore, Singapore, and the Medicine School, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA.