Toward Certified Robustness of Distance Metric Learning

Xiaochen Yang, Yiwen Guo, Mingzhi Dong, and Jing-Hao Xue, *Senior Member, IEEE*

***Abstract*—Metric learning aims to learn a distance metric such that semantically similar instances are pulled together while dissimilar instances are pushed away. Many existing methods consider maximizing or at least constraining a distance margin in the feature space that separates similar and dissimilar pairs of instances to guarantee their generalization ability. In this article, we advocate imposing an adversarial margin in the input space so as to improve the generalization and robustness of metric learning algorithms. We first show that the adversarial margin, defined as the distance between training instances and their closest adversarial examples in the input space, takes account of both the distance margin in the feature space and the correlation between the metric and triplet constraints. Next, to enhance robustness to instance perturbation, we propose to enlarge the adversarial margin through minimizing a derived novel loss function termed the perturbation loss. The proposed loss can be viewed as a data-dependent regularizer and easily plugged into any existing metric learning methods. Finally, we show that the enlarged margin is beneficial to the generalization ability by using the theoretical technique of algorithmic robustness. Experimental results on 16 datasets demonstrate the superiority of the proposed method over existing state-of-the-art methods in both discrimination accuracy and robustness against possible noise.**

***Index Terms*—Adversarial perturbation, generalization ability, metric learning, nearest neighbor (NN), robustness.**

I. Introduction

METRIC learning focuses on learning similarity or dissimilarity between data. Research on metric learning originates from at least 2002, where [1] first proposes to formulate it as an optimization problem. Since then, many metric learning methods have been proposed for classification [2], [3], [4], clustering [5], and information retrieval [6], [7]. In particular, the methods have shown to be particularly superior in open-set classification and few-shot classification with notable applications in, for example, face verification [8], [9] and person re-identification [10], [11].

One commonly studied distance metric is the generalized Mahalanobis distance, which defines the distance between any two instances $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathbb{R}^p$ as

$$d_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sqrt{(\boldsymbol{x}_i - \boldsymbol{x}_j)^T \boldsymbol{M} (\boldsymbol{x}_i - \boldsymbol{x}_j)}$$

where $\boldsymbol{M}$ is a positive semidefinite (PSD) matrix. Owing to its PSD property, $\boldsymbol{M}$ can be decomposed into $\boldsymbol{L}^T\boldsymbol{L}$. Thus, computing the Mahalanobis distance is equivalent to linearly transforming the instances from the input space to the feature space via $\boldsymbol{L}$ and then computing the Euclidean distance $\|\boldsymbol{L}\boldsymbol{x}_i - \boldsymbol{L}\boldsymbol{x}_j\|_2$ in the transformed space.

To learn a specific distance metric for each task, prior knowledge on instance similarity and dissimilarity should be provided as side information. Metric learning methods differ by the form of side information they use and the supervision encoded in similar and dissimilar pairs. For example, pairwise constraints enforce the distance between instances of the same class to be small (or smaller than a threshold value) and the distance between instances of different classes to be large (or larger than a threshold value) [1], [5]. The thresholds could be either predefined or learned for similar and dissimilar pairs [12], [13]. In triplet constraints $(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_l)$, distance between the different-class pair $(\boldsymbol{x}_i, \boldsymbol{x}_l)$ should be larger than distance between the same-class pair $(\boldsymbol{x}_i, \boldsymbol{x}_j)$, and typically, plus a margin [14], [15], [16], [17]. More recently, quadruplet constraints are proposed, which require the difference in the distance of two pairs of instances to exceed a margin [18], and $(N+1)$-tuplet extends the triplet constraint for multiclass classification [19].

The gap between thresholds in pairwise constraints and the margin in triplet and quadruplet constraints are both designed to learn a distance metric that could ensure good generalization of the subsequent k-nearest neighbor (kNN) classifier. However, such a distance margin imposed in the feature space does not consider the correlation between the data and the learned metric. Consequently, it may be insufficient to withstand a small perturbation of the instance occurred in the input space, thereby failing to certify the robustness or even possess the anticipated generalization benefit. As illustrated in Fig. 1(upper), while $\boldsymbol{x}_i$ selects the same-class instance $\boldsymbol{x}_j$ as its NN in the feature space, a tiny perturbation from $\boldsymbol{x}_i$ to $\boldsymbol{x}'_i$ in the input space can be magnified by the learned distance metric, leading to a change in its NN from $\boldsymbol{x}_j$ to the different-class instance $\boldsymbol{x}_l$. When the NN algorithm is used as the classifier, the perturbation results in an incorrect label prediction.

Manuscript received 13 December 2020; revised 27 August 2021 and 4 May 2022; accepted 15 August 2022. *(Xiaochen Yang and Yiwen Guo contributed equally to this work.) (Corresponding author: Xiaochen Yang.)*

Xiaochen Yang is with the School of Mathematics and Statistics, University of Glasgow, Glasgow G12 8TA, U.K. (e-mail: xiaochen.yang@glasgow.ac.uk).

Yiwen Guo was with ByteDance AI Lab, Beijing 100080, China. He resides in Beijing 100000, China.

Mingzhi Dong is with the Shanghai Key Laboratory of Data Science and the School of Computer Science, Fudan University, Shanghai 200438, China.

Jing-Hao Xue is with the Department of Statistical Science, University College London, London WC1E 6BT, U.K.

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TNNLS.2022.3199902.

Digital Object Identifier 10.1109/TNNLS.2022.3199902

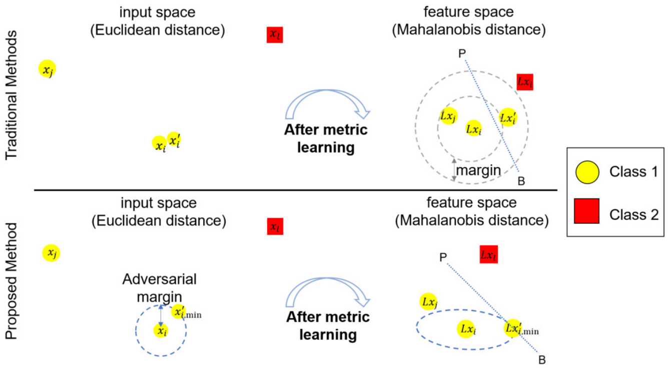


Fig. 1. (Upper) Traditional methods aim to separate the same-class pair $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and the different-class pair $(\boldsymbol{x}_i, \boldsymbol{x}_l)$ by a margin in the feature space. While $\boldsymbol{x}_i$ has $\boldsymbol{x}_j$ as its nearest neighbor (NN) in the feature space and is correctly predicted by using the NN classifier, the metric is sensitive to perturbation in the input space; a tiny perturbation from $\boldsymbol{x}_i$ to $\boldsymbol{x}_i'$ changes the NN to $\boldsymbol{x}_l$ and leads to an incorrect prediction. (Bottom) Proposed method aims to enlarge the adversarial margin in the input space, which equals to the Euclidean distance between $\boldsymbol{x}_i$ and the closest point $\boldsymbol{x}_{i,\min}$ in the input space that lies on the decision boundary in the feature space (indicated by PB) and quantifies the maximum degree to which robustness can be certified.

In this article, we propose a simple yet effective method to enhance the generalization ability of metric learning algorithms and their robustness against instance perturbation. As shown in Fig. 1(bottom), the principal idea is to enlarge the adversarial margin, defined as the distance between a training instance and its closest adversarial example in the input space [20].

In particular, our contributions are fourfold.

1) We identify that the distance margin, widely used in existing methods, is insufficient to withstand adversarial examples, and we introduce a direct measure of robustness termed the adversarial margin, which quantifies the maximum degree to which a training instance could be perturbed without changing the label of its NN (or kNNs if required) in the feature space. Building on a geometric insight, we derive an analytically simple solution to the adversarial margin, which reveals the importance of an adaptive margin considering the correlation between the data and the distance metric (Section II-A and II-B).
2) We define a novel hinge-like perturbation loss to penalize the adversarial margin for being small. The proposed loss function serves as a general approach to enhancing robustness, as it can be optimized jointly with any existing triplet-based metric learning methods; the optimization problem suggests that our method learns a discriminative metric in a weighted manner and simultaneously functions as a data-dependent regularization (see Section II-C).
3) We show the benefit of enlarging the adversarial margin to the generalization ability of the learned distance metric by using the theoretical technique of algorithmic robustness [21] (Theorem 1, Section II-D).
4) We conduct experiments on 16 datasets in both noise-free and noisy settings. Results show that the proposed method outperforms state-of-the-art robust metric learning methods in terms of classification accuracy and validate its robustness to possible noise in the input space (see Section IV).

Notation: Let $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$ denote the set of training instance and label pairs, where $\boldsymbol{x}_i \in \mathcal{X} \subseteq \mathbb{R}^p$ and $y_i \in \mathcal{Y} = \{1, \ldots, C\}$; $\mathcal{X}$ is called the input space. Our framework is based on triplet constraints $\{\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_l\}$ and we adopt the following strategy for generating triplets [14]:

$$\mathcal{S} = \left\{(\boldsymbol{x}_i, \boldsymbol{x}_j) : \boldsymbol{x}_j \in \{k\text{NNs with the same class label of } \boldsymbol{x}_i\}\right\}$$
$$\mathcal{R} = \left\{(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_l) : (\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{S}, y_i \neq y_l\right\}.$$

$\boldsymbol{x}_j$ is termed the target neighbor of $\boldsymbol{x}_i$ and $\boldsymbol{x}_l$ is termed the impostor. $|\mathcal{S}|$ and $|\mathcal{R}|$ denote the numbers of elements in the sets $\mathcal{S}$ and $\mathcal{R}$, respectively. $d_{\boldsymbol{E}}$ and $d_{\boldsymbol{M}}$ denote the Euclidean and Mahalanobis distances, respectively; $\boldsymbol{M} \in \mathbb{S}_+^p$, where $\mathbb{S}_+^p$ is the cone of $p \times p$ real-valued PSD matrices. $\boldsymbol{M}^2 = \boldsymbol{M}\boldsymbol{M}$. $\mathbb{1}[\cdot]$ denotes the indicator function and $[a]_+ = \max(a, 0)$ for $a \in \mathbb{R}$.

II. Methodology

In this section, we introduce our method for enhancing robustness of triplet-based metric learning algorithms through maximizing the adversarial margin. First, we review the existing distance margin and provide the rationale for enlarging the adversarial margin. Second, an explicit formula for the adversarial margin is derived. Third, we propose the perturbation loss to encourage a larger adversarial margin and present its optimization jointly with the existing large (distance) margin NN (LMNN) algorithm. Lastly, we show that enlarging the adversarial margin is beneficial to the generalization ability of the learned distance metric.

A. Motivation for Enlarging the Adversarial Margin

Suppose $\boldsymbol{x}_i$ is a training instance and $\boldsymbol{x}_j, \boldsymbol{x}_l$ are the NN of $\boldsymbol{x}_i$ from the same class and from the different class respectively. Many triplet-based methods, such as LMNN [14], impose the following constraint on the triplet:

$$f(\boldsymbol{x}_i) := d_{\boldsymbol{M}}^2(\boldsymbol{x}_i, \boldsymbol{x}_l) - d_{\boldsymbol{M}}^2(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 1.$$

When the constraint is satisfied, $\boldsymbol{x}_i$ will be correctly classified using the NN classifier. Moreover, the value one represents the unit margin at the distance level and is designed to robustify the model against small noises in training instances.

Nevertheless, the distance margin may be insufficient to withstand deliberately manipulated perturbations. Let $\Delta\boldsymbol{x}_i$ denote a perturbation of $\boldsymbol{x}_i$. When the perturbation size is constrained as $\|\Delta\boldsymbol{x}_i\|_2 \leq r$, $f(\boldsymbol{x}_i + \Delta\boldsymbol{x}_i)$ decreases the most from $f(\boldsymbol{x}_i)$ if $\Delta\boldsymbol{x}_i$ is chosen in the direction of $\boldsymbol{M}(\boldsymbol{x}_l - \boldsymbol{x}_j)$: $f(\boldsymbol{x}_i + \Delta\boldsymbol{x}_i) - f(\boldsymbol{x}_i) = 2\Delta\boldsymbol{x}_i^T\boldsymbol{M}(\boldsymbol{x}_j - \boldsymbol{x}_l) = -2r\|\boldsymbol{M}(\boldsymbol{x}_l - \boldsymbol{x}_j)\|_2$. Therefore, in order to correctly classify the perturbed instance $\boldsymbol{x}_i + \Delta\boldsymbol{x}_i$, it is required that $f(\boldsymbol{x}_i + \Delta\boldsymbol{x}_i)$ is positive, that is, $\|\boldsymbol{M}(\boldsymbol{x}_l - \boldsymbol{x}_j)\|_2$ should be small. One way to reduce this value is by regularizing the spectral norm of $\boldsymbol{M}$. However, it is demanding for the metric to satisfy the large distance margin for all triplets and meanwhile keep a small spectral norm (SN).

To achieve robustness against instance perturbation, we suggest an alternative way by maximizing the adversarial margin, defined as the distance between the training instance and its closest adversarial example [20]. More concretely, an adversarial example is a perturbed point whose NN, identified based on the learned Mahalanobis distance, changes from an instance of the same class to one of a different class; consequently, it will be misclassified by the NN classifier and increase the risk of misclassification by kNN. In terms of previous notations, an adversarial example is a perturbed point $\boldsymbol{x}_i + \Delta\boldsymbol{x}_i$ such that $f(\boldsymbol{x}_i + \Delta\boldsymbol{x}_i) < 0$. If all adversarial examples of an instance are far away from the instance itself, i.e., there is no $\Delta\boldsymbol{x}_i$ such that $\|\Delta\boldsymbol{x}_i\|_2 \leq r$ and $f(\boldsymbol{x}_i + \Delta\boldsymbol{x}_i) < 0$, a high degree of robustness is achieved. Building on this rationale, we will first find the closest adversarial example and then push this point away from the training instance. Moreover, since the test instance can be regarded as a perturbed copy of training instances [21], improving robustness on correctly classified training instances also helps enhance the generalization ability of the learned metric.

B. Derivation of Adversarial Margin

We start by deriving a closed-form solution to the closest adversarial example. Given a training instance $\boldsymbol{x}_i$ and the associated triplet constraint $(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_l)$, we aim to find the closest point $\boldsymbol{x}_{i,\min}$ to $\boldsymbol{x}_i$ in the input space that lies on the decision boundary formed by $\boldsymbol{x}_j$ and $\boldsymbol{x}_l$ in the feature space. Note that closeness is defined in the input space and will be calculated using the Euclidean distance since we target at changes on the original feature of an instance; and that the decision boundary is found in the feature space since kNNs are identified by using the Mahalanobis distance. Mathematically, we can formulate the closest adversarial example $\boldsymbol{x}_{i,\min}$ as follows:

$$\boldsymbol{x}_{i,\min} = \underset{\boldsymbol{x}'_i \in \mathbb{R}^p}{\arg\min} \left(\boldsymbol{x}'_i - \boldsymbol{x}_i\right)^T \left(\boldsymbol{x}'_i - \boldsymbol{x}_i\right)$$

$$\text{s.t.} \left(\boldsymbol{L}\boldsymbol{x}'_i - \frac{\boldsymbol{L}\boldsymbol{x}_j + \boldsymbol{L}\boldsymbol{x}_l}{2}\right)^T \left(\boldsymbol{L}\boldsymbol{x}_l - \boldsymbol{L}\boldsymbol{x}_j\right) = 0. \quad (1)$$

The objective function of (1) corresponds to minimizing the Euclidean distance from the training instance $\boldsymbol{x}_i$. The constraint represents the decision boundary, which is the perpendicular bisector of points $\boldsymbol{L}\boldsymbol{x}_j$ and $\boldsymbol{L}\boldsymbol{x}_l$. In other words, it is a hyperplane that is perpendicular to the line joining points $\boldsymbol{L}\boldsymbol{x}_j$ and $\boldsymbol{L}\boldsymbol{x}_l$ and passes their midpoint $((\boldsymbol{L}\boldsymbol{x}_j + \boldsymbol{L}\boldsymbol{x}_l)/2)$; all points on the hyperplane are equidistant from $\boldsymbol{L}\boldsymbol{x}_j$ and $\boldsymbol{L}\boldsymbol{x}_l$.

Since (1) minimizes a convex quadratic function with an equality constraint, we can find an explicit formula for $\boldsymbol{x}_{i,\min}$ by using the method of Lagrangian multipliers; detailed derivation is provided in Section VI in the supplementary material

$$\boldsymbol{x}_{i,\min} = \boldsymbol{x}_i + \frac{\left(\frac{\boldsymbol{x}_j + \boldsymbol{x}_l}{2} - \boldsymbol{x}_i\right)^T \boldsymbol{M}\left(\boldsymbol{x}_l - \boldsymbol{x}_j\right)}{(\boldsymbol{x}_l - \boldsymbol{x}_j)^T \boldsymbol{M}^2 \left(\boldsymbol{x}_l - \boldsymbol{x}_j\right)} \boldsymbol{M}(\boldsymbol{x}_l - \boldsymbol{x}_j). \quad (2)$$

With the solution of $\boldsymbol{x}_{i,\min}$, we can now calculate the squared Euclidean distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_{i,\min}$

$$d_{\boldsymbol{E}}^2(\boldsymbol{x}_i, \boldsymbol{x}_{i,\min}) = \frac{\left(d_{\boldsymbol{M}}^2(\boldsymbol{x}_i, \boldsymbol{x}_l) - d_{\boldsymbol{M}}^2(\boldsymbol{x}_i, \boldsymbol{x}_j)\right)^2}{4 d_{\boldsymbol{M}^2}^2(\boldsymbol{x}_j, \boldsymbol{x}_l)}. \quad (3)$$

For clarity, we will call $d_{\boldsymbol{E}}(\boldsymbol{x}_i, \boldsymbol{x}_{i,\min})$ the adversarial margin, in contrast to the distance margin as in LMNN. It represents the maximum amount of tolerance for perturbation while retaining prediction correctness. The numerator of (3) is the square of the standard distance margin, and the denominator is the squared L_2-norm of $\boldsymbol{M}(\boldsymbol{x}_l - \boldsymbol{x}_j)$. Therefore, in order to achieve a large adversarial margin, the metric should push $\boldsymbol{x}_l$ away from the neighborhood of $\boldsymbol{x}_i$ by expanding the distance in the direction that has a small correlation with $\boldsymbol{x}_l - \boldsymbol{x}_j$ (the optimal direction is orthogonal to $\boldsymbol{x}_l - \boldsymbol{x}_j$).

Remark 1: The objective function in (1) defines a hypersphere in the input space, which characterizes perturbations of equal magnitude in all directions, e.g., isotropic Gaussian noise. To model heterogeneous and correlated perturbation, we can extend the objective function by defining an arbitrary oriented hyperellipsoid, as discussed in Section VI in the supplementary material.

C. Metric Learning via Minimizing the Perturbation Loss

To improve robustness of distance metric, we design a perturbation loss to promote an increase in the adversarial margin. Two situations need to be distinguished here. First, when the NN of $\boldsymbol{x}_i$ is an instance from the same class, we will penalize a small adversarial margin by using the hinge loss $[\tau^2 - d_{\boldsymbol{E}}^2(\boldsymbol{x}_i, \boldsymbol{x}_{i,\min})]_+$. The reasons are that: 1) the adversarial margin is generally smaller for hard instances that are close to the class boundary in contrast to those locating far away and 2) it is these hard instances that are more vulnerable to perturbation and demand an improvement in their robustness. Therefore, we introduce τ for directing attention to hard instances and controlling the desired margin. Second, in the other situation where the NN of $\boldsymbol{x}_i$ belongs to a different class, metric learning should focus on satisfying the distance requirement specified in the triplet constraint. In this case, we simply assign a large penalty of τ^2 to promote a nonincreasing loss function. Integrating these two situations, we propose the following perturbation loss:

$$J_{\mathrm{P}} = \frac{1}{|\mathcal{R}|} \sum_{\mathcal{R}} \Big\{ \left[\tau^2 - \tilde{d}_{\boldsymbol{E}}^2(\boldsymbol{x}_i, \boldsymbol{x}_{i,\min})\right]_+ \times \mathbb{1}\Big[d_{\boldsymbol{M}}^2(\boldsymbol{x}_i, \boldsymbol{x}_l) > d_{\boldsymbol{M}}^2(\boldsymbol{x}_i, \boldsymbol{x}_j)\Big] + \tau^2 \mathbb{1}\left[d_{\boldsymbol{M}}^2(\boldsymbol{x}_i, \boldsymbol{x}_l) \leq d_{\boldsymbol{M}}^2(\boldsymbol{x}_i, \boldsymbol{x}_j)\right] \Big\} \quad (4)$$

where $\sum_{\mathcal{R}}$ is an abbreviation for $\sum_{(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_l) \in \mathcal{R}}$. To prevent the denominator of (3) from being zero, which may happen when different-class instances $\boldsymbol{x}_j$ and $\boldsymbol{x}_l$ are close to each other, we add a small constant ϵ (ϵ =1e-10) to the denominator; that is, $\tilde{d}_{\boldsymbol{E}}^2(\boldsymbol{x}_i, \boldsymbol{x}_{i,\min}) = ((d_{\boldsymbol{M}}^2(\boldsymbol{x}_i, \boldsymbol{x}_l) - d_{\boldsymbol{M}}^2(\boldsymbol{x}_i, \boldsymbol{x}_j))^2)/(4(d_{\boldsymbol{M}^2}^2(\boldsymbol{x}_j, \boldsymbol{x}_l) + \epsilon))$.

The proposed perturbation loss can be readily included in the objective function of any metric learning methods and is particularly useful to triplet-based methods. When the same triplet set is used for supervising metric learning and deriving adversarial examples, our method can encourage the triplets to meet the distance margin by learning a discriminative metric. For this reason, we adapt LMNN as an example for its wide use and effective classification performance. The objective function of LMNN with the perturbation loss is as follows:

$$\begin{aligned}\min_{\boldsymbol{M}\in\mathbb{S}_+^p} J &= J_{\text{LMNN}} + \lambda J_{\text{P}}\\ J_{\text{LMNN}} &= (1-\mu)\frac{1}{|\mathcal{S}|}\sum_{\mathcal{S}} d_{\boldsymbol{M}}^2(\boldsymbol{x}_i,\boldsymbol{x}_j)\\ &\quad + \mu\frac{1}{|\mathcal{R}|}\sum_{\mathcal{R}}\left[1 + d_{\boldsymbol{M}}^2(\boldsymbol{x}_i,\boldsymbol{x}_j) - d_{\boldsymbol{M}}^2(\boldsymbol{x}_i,\boldsymbol{x}_l)\right]_+ \qquad (5)\end{aligned}$$

where $\sum_{\mathcal{S}}$ stands for $\sum_{(\boldsymbol{x}_i,\boldsymbol{x}_j)\in\mathcal{S}}$. The weight parameter $\lambda > 0$ controls the importance of perturbation loss (J_P) relative to the loss function of LMNN (J_{LMNN}). $\mu \in (0, 1)$ balances the impacts between pulling together target neighbors and pushing away impostors.

We adopt the projected gradient descent algorithm to solve the above optimization problem. The gradient of J_P and J_{LMNN} are given as follows:

$$\begin{aligned}\frac{\partial J_P}{\partial \boldsymbol{M}} &= \frac{1}{|\mathcal{R}|}\sum_{\mathcal{R}}\alpha_{ijl}\left\{\frac{d_{\boldsymbol{M}}^2(\boldsymbol{x}_i,\boldsymbol{x}_l) - d_{\boldsymbol{M}}^2(\boldsymbol{x}_i,\boldsymbol{x}_j)}{2\left(d_{\boldsymbol{M}^2}^2(\boldsymbol{x}_j,\boldsymbol{x}_l)+\epsilon\right)}(\boldsymbol{X}_{ij}-\boldsymbol{X}_{il})\right.\\ &\qquad + \frac{\left(d_{\boldsymbol{M}}^2(\boldsymbol{x}_i,\boldsymbol{x}_l) - d_{\boldsymbol{M}}^2(\boldsymbol{x}_i,\boldsymbol{x}_j)\right)^2}{4\left(d_{\boldsymbol{M}^2}^2(\boldsymbol{x}_j,\boldsymbol{x}_l)+\epsilon\right)^2}\\ &\qquad \left.\times(\boldsymbol{M}\boldsymbol{X}_{jl} + \boldsymbol{X}_{jl}\boldsymbol{M})\right\}\\ \frac{\partial J_{\text{LMNN}}}{\partial \boldsymbol{M}} &= \frac{1-\mu}{|\mathcal{S}|}\sum_{\mathcal{S}}\boldsymbol{X}_{ij} + \frac{\mu}{|\mathcal{R}|}\sum_{\mathcal{R}}\beta_{ijl}(\boldsymbol{X}_{ij}-\boldsymbol{X}_{il}) \qquad (6)\end{aligned}$$

where $\alpha_{ijl} = \mathbb{1}[d_{\boldsymbol{M}}^2(\boldsymbol{x}_i,\boldsymbol{x}_l) > d_{\boldsymbol{M}}^2(\boldsymbol{x}_i,\boldsymbol{x}_j), \tilde{d}_{\boldsymbol{E}}(\boldsymbol{x}_i,\boldsymbol{x}_{i,\min}) \le \tau]$, $\beta_{ijl} = \mathbb{1}[1 + d_{\boldsymbol{M}}^2(\boldsymbol{x}_i,\boldsymbol{x}_j) - d_{\boldsymbol{M}}^2(\boldsymbol{x}_i,\boldsymbol{x}_l) \ge 0]$; $\boldsymbol{X}_{ij} = (\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^T$ and $\boldsymbol{X}_{il}, \boldsymbol{X}_{jl}$ are defined similarly. The gradient of J_P is a sum of two descent directions. The first direction $\boldsymbol{X}_{ij} - \boldsymbol{X}_{il}$ agrees with LMNN, indicating that our method updates the metric toward better discrimination in a weighted manner. The second direction $\boldsymbol{M}\boldsymbol{X}_{jl}+\boldsymbol{X}_{jl}\boldsymbol{M}$ controls the scale of $\boldsymbol{M}$; the metric will descend at a faster pace in the direction of a larger correlation between $\boldsymbol{M}$ and $\boldsymbol{X}_{jl}$. This suggests our method functions as a data-dependent regularization. Let $\boldsymbol{M}^t$ denote the Mahalanobis matrix learned at the tth iteration. The distance matrix will be updated as

$$\boldsymbol{M}^{t+1} = \boldsymbol{M}^t - \gamma\left(\frac{\partial J_{\text{LMNN}}}{\partial \boldsymbol{M}^t} + \lambda\frac{\partial J_P}{\partial \boldsymbol{M}^t}\right)$$

where γ denotes the learning rate. Following [14]'s work, γ is increased by 1% if the loss function decreases and decreased by 50% otherwise. To guarantee the PSD property, we factorize $\boldsymbol{M}^{t+1}$ as $\boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^T$ via eigendecomposition and truncate all negative eigenvalues to zero, i.e., $\boldsymbol{M}^{t+1} = \boldsymbol{V}\max(\boldsymbol{\Lambda}, 0)\boldsymbol{V}^T$.

Remark 2: The proposed perturbation loss is a generic approach to improving robustness against possible perturbation. In Section VII in the supplementary material, we illustrate examples of incorporating the perturbation loss into two different types of triplet-based methods, sparse compositional metric learning (SCML) [15] and proxy neighborhood component analysis (ProxyNCA++) [22]. SCML revises the structure of the Mahalanobis distance by representing it as a sparse and nonnegative combination of rank-one basis elements, which typically results in less number of parameters to be estimated. ProxyNCA++ revises the construction of triplet constraints by replacing nearest instances $\boldsymbol{x}_j$ and $\boldsymbol{x}_l$ with nearest proxy points. The proxies are learned to represent each class, and the resulting method is shown to generalize well on small datasets [23], robust to outliers and noisy labels [24], and improves computational efficiency on large-scale datasets.

Remark 3: Learning a distance metric for extremely high-dimensional data will result in a large number of parameters to be estimated and potentially suffer from overfitting. In order to reduce the input dimensionality, PCA is often applied to preprocess the data prior to metric learning [14], [25]. In Section VI-A in the supplementary material, we extend the proposed method such that the distance metric learned in the low-dimensional PCA subspace could still achieve robustness against perturbation in the original high-dimensional input space. The decision boundary of NN classifier [i.e., the constraint of (1)] is revised in order to take account of the linear transformation matrix induced by the Mahalanobis distance and that of PCA. The proposed extension will be evaluated in Section IV-C.

D. Generalization Benefit

From the perspective of algorithmic robustness [21], enlarging the adversarial margin could potentially improve the generalization ability of triplet-based metric learning methods. The following generalization bound, i.e., the gap between the generalization error $\mathcal{L}$ and the empirical error ℓ_{emp}, follows from the pseudo-robust theorem of [26]. Preliminaries and derivations are given in Section VIII in the supplementary material.

Theorem 1: Let $\boldsymbol{M}^*$ be the optimal solution to (5). Then for any $\delta > 0$, with probability at least $1 - \delta$ we have

$$\begin{aligned}&|\mathcal{L}(\boldsymbol{M}^*) - \ell_{\text{emp}}(\boldsymbol{M}^*)|\\ &\quad \le \frac{\hat{n}(t_s)}{n^3} + B\left(\frac{n^3 - \hat{n}(t_s)}{n^3} + 3\sqrt{\frac{2K\ln 2 + 2\ln 1/\delta}{n}}\right) \qquad (7)\end{aligned}$$

where $\hat{n}(t_s)$ denotes the number of triplets whose adversarial margins are larger than τ, B is a constant denoting the upper bound of the loss function [i.e., (5)], and K denotes the number of disjoint sets that partition the input-label space and equals to $|\mathcal{Y}|(1 + (2/\tau))^p$.

Enlarging the desired adversarial margin τ will affect two quantities in (7), namely K and $\hat{n}(t_s)$. First, since K equals to $|\mathcal{Y}|(1 + (2/\tau))^p$, increasing τ will cause K to decrease at a polynomial rate of the input dimensionality p. Moreover, as the right-hand side of (7) is a function of K ($\mathcal{O}(K^{1/2})$), this means that the upper bound of generalization gap reduces at a rate of $p^{1/2}$. Hence, for datasets with a relative large number of features, a small improvement in the adversarial margin can greatly benefit the generalization ability of the learned metric.

Second, when τ increases, less triplets will satisfy the condition that their adversarial margin is larger than τ; that is, $\hat{n}(t_s)$ decreases with τ. Meanwhile, since $B > 1$, the upper bound is a decreasing function of $\hat{n}(t_s)$. Therefore, enlarging τ leads to an increase in the upper bound. However, the rate of such increase depends on the datasets. For example, if most instances in the dataset are well separated and have a margin in the original input space, enlarging the desired adversarial margin τ will not have a large impact on $\hat{n}(t_s)$, the upper bound, and thus the generalization gap.

In summary, for datasets with many features and most instances being separable, we expect an improvement in the generalization ability of the learned distance metric from enlarging the adversarial margin.

III. Related Work

A. Robust Metric Learning

To make machine learning models more secure and trustworthy, robustness to input perturbations is a crucial dimension [27]. More importantly, designing such robust metric learning algorithms is particularly vital to safety-critical applications, such as healthcare [28], network intrusion detection [29], and surveillance systems based on faces [30], gaits [31], and other biometric traits [32].

Existing approaches to improving the robustness of Mahalanobis distances can be categorized into four main types. The first type of method imposes structural assumption or regularization over $\boldsymbol{M}$ so as to avoid overfitting [25], [33], [34], [35], [36], [37]. Methods with structural assumption are proposed for classifying images and achieve robustness by exploiting the structural information of images; however, such information is generally unavailable in the symbolic datasets that will be studied in this article. Regularization-based methods are proposed to reduce the risk of overfitting to feature noise in the training set. Our proposal, which is aimed to withstand test-time perturbation, does not conflict with these methods and can be combined with them to learn a more effective and robust distance metric; an example is shown in Section IV-C.

The second type of method adopts loss functions that are less sensitive to outlier samples or noisy labels. In most metric learning methods, loss functions are founded on the squared L_2-norm distance for computational efficiency. However, such choice may be sensitive to outliers. To overcome this limitation, several remedies have been proposed, such as using L_1-norm distances [38] and metric based on the signal-to-noise ratio (SNR) [39], or replacing the square function with the maximum correntropy criterion [40].

The third type of method studies robustness to training noise [41], [42]. These methods explicitly model the noise distribution or identify clean latent examples, and consequently, use the expected Mahalanobis distance to adjust the value of the distance margin for each triplet. Our method can also be viewed as imposing a data-dependent and dynamic margin—to achieve the same adversarial margin, triplets that have a higher correlation between $\boldsymbol{x}_l - \boldsymbol{x}_j$ and the metric $\boldsymbol{M}$ should satisfy a larger distance margin. However, the focus of our work is orthogonal to the aforementioned two types of method.

The last type of method generates hard instances through adversarial learning and trains a metric to fare well in the new hard problem [43], [44]. While sharing the aim of improving metric robustness, our method is intrinsically different from them. Their methods approach the task at a data level, where real examples are synthesized based on the criterion of incurring large losses. Our method tackles perturbation at a model level, where a loss function is derived by considering the definition of robustness with respect to the decision maker kNN. By preventing change in the NN in a strict manner, our method is capable of obtaining a certification on adversarial margin.

B. Adversarial Robustness of Deep Metric Learning

More recently, deep metric learning has been investigated intensively, which replaces the linear projection induced by the Mahalanobis distance with deep neural networks. While deep neural networks improve the discriminability between classes, they are found to be nonrobust and vulnerable to adversarial examples. Robust optimization [20], [45] is one of the most effective approaches to improving adversarial robustness, which trains the network to be robust against adversarial perturbations that are mostly constructed via gradient-based optimization; [46] adapts it to deep metric learning by considering the interdependence between data points in pairwise or triplet constraints. Another way to enhance robustness and generalization ability is by attaining a large margin in the input space, which dates back to support vector machines [47] and inspires this work. Due to the hierarchical nonlinear nature of deep networks, the input-space margin cannot be computed exactly and a variety of approximations have been proposed [48], [49], [50], [51]. In this work, we investigate such margin in the framework of metric learning, defines it specifically with respect to the kNN classifier, and provide an exact and analytical solution to the margin. The analytical solution to the margin provides fascinating insights into essential factors for the robustness of distance metrics.

C. Adversarial Robustness of kNN Classifiers

While the notion of adversarial examples applies to kNN classifiers, existing methods for deep neural networks cannot be implemented directly due to the nondifferential nature of the classifier. Papernot *et al.* [52] and Sitawarin and Wagner [53] propose continuous substitutes of kNN, from which gradient-based adversarial examples can be constructed to attack the classifier. Wang *et al.* [54] formulates a series of quadratic programming (QP) problems and proposes an efficient algorithm to search exhaustively over all training samples and compute the minimal adversarial perturbation for the 1-NN classifier. In addition, the dual solution to these QP problems can be used for robustness verification. Yang *et al.* [55] proposes to improve adversarial robustness for kNN by pruning the training set in order to satisfy the condition defined through the robustness radius, i.e., the norm of the minimal adversarial perturbation. Our work also aims to robustify kNN, but achieves it through enlarging the adversarial margin.

IV. Experiments

In this section, we first present two toy examples to illustrate the difference in the learning mechanisms of LMNN and the proposed method dubbed LMNN-PL. Next,

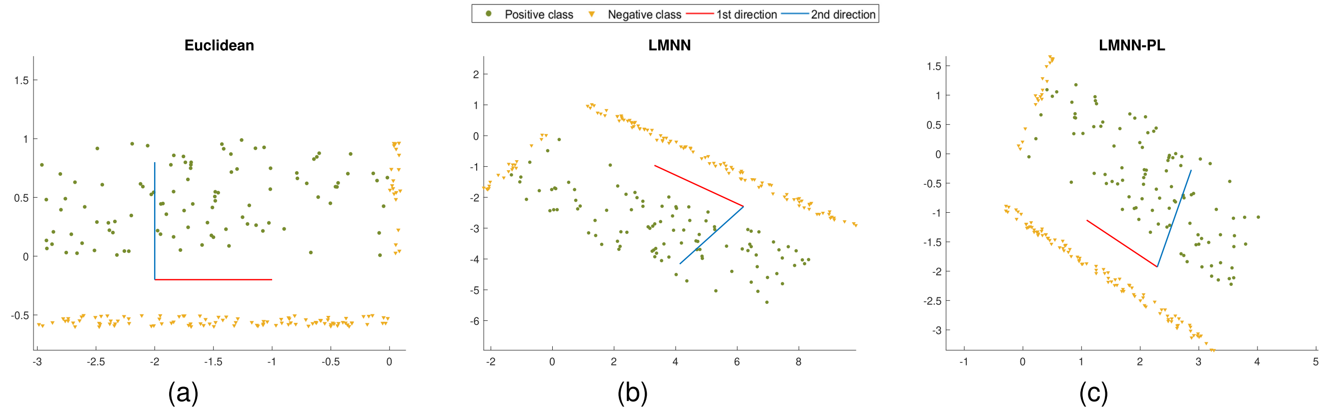


Fig. 2. Comparison of learning mechanisms of LMNN and LMNN-PL when features exhibit different separability. (a) $\boldsymbol{M} = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$ $\bar{d}_E(\boldsymbol{x}_i, \boldsymbol{x}_{i,\min}) = 0.17$. (b) $\boldsymbol{M} = \left[\begin{smallmatrix} 10.2 & 3.5 \\ 3.5 & 7.8 \end{smallmatrix}\right]$ $\bar{d}_E(\boldsymbol{x}_i, \boldsymbol{x}_{i,\min}) = 0.15$. (c) $\boldsymbol{M} = \left[\begin{smallmatrix} 2.1 & 0.6 \\ 0.6 & 3.1 \end{smallmatrix}\right]$ $\bar{d}_E(\boldsymbol{x}_i, \boldsymbol{x}_{i,\min}) = 0.16$.

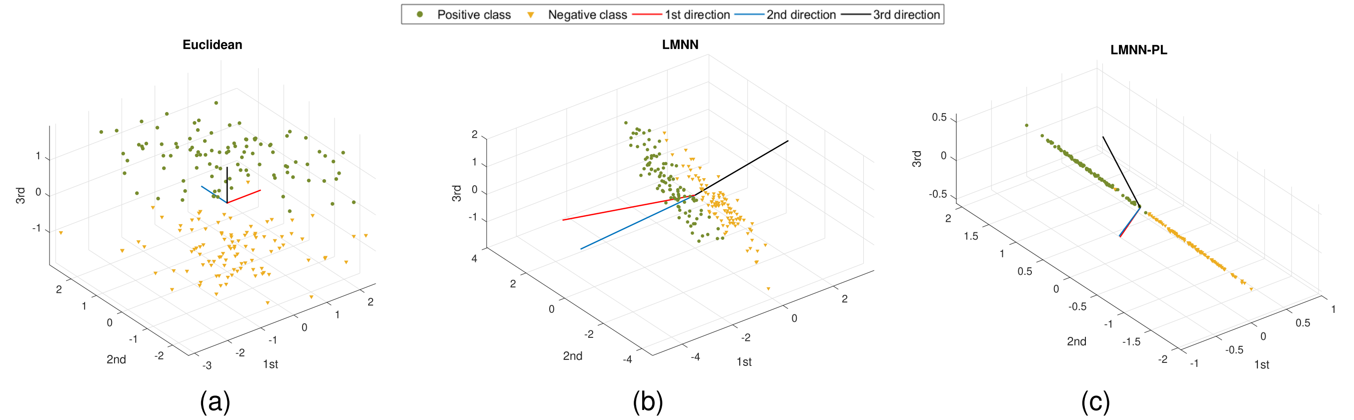


Fig. 3. Comparison of learning mechanisms of LMNN and LMNN-PL when confronting the problem of multicollinearity. (a) $\boldsymbol{M} = \left[\begin{smallmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{smallmatrix}\right]$ $\bar{d}_E(\boldsymbol{x}_i, \boldsymbol{x}_{i,\min}) = 0.83$. (b) $\boldsymbol{M} = \left[\begin{smallmatrix} 22 & 23 & -17 \\ 23 & 28 & -19 \\ -17 & -19 & 14 \end{smallmatrix}\right]$ $\bar{d}_E(\boldsymbol{x}_i, \boldsymbol{x}_{i,\min}) = 0.05$. (c) $\boldsymbol{M} = \left[\begin{smallmatrix} 0.45 & 0.45 & -0.08 \\ 0.45 & 0.45 & -0.06 \\ -0.08 & -0.06 & 0.85 \end{smallmatrix}\right]$ $\bar{d}_E(\boldsymbol{x}_i, \boldsymbol{x}_{i,\min}) = 0.70$.

we compare LMNN-PL with state-of-the-art methods on 16 benchmark datasets (13 low/medium-dimensional and three high-dimensional) and investigate the relationship between adversarial margin, generalization ability, and robustness. Finally, the computational aspect of our method is discussed.

A. Comparisons Between LMNN and LMNN-PL

We design two experiments to compare the metrics learned with the objective of enhancing class discriminability and of certified robustness. In the first example, a 2-D binary classification dataset is simulated, as shown in Fig. 2a. The positive class includes 100 instances drawn uniformly from $[-3, 0]$ in the horizontal (abbr. 1st) direction and $[0, 1]$ in the vertical (abbr. 2nd) direction. The negative class consists of two clusters, where the first cluster includes 100 instances drawn from $U(-3, 0)$ and $U(-0.6, -0.5)$ in the 1st and 2nd directions, respectively, and the second cluster includes 20 instances drawn from $U(0, 0.1)$ and $U(0, 1)$ in the two directions respectively. By design, instances of positive and negative classes can be separated in both directions, while the separability in the 1st direction is much smaller than the 2nd direction. Fig. 2(b) and (c) show the instances in the projected feature space with metrics learned from LMNN and LMNN-PL, respectively; the projection direction is indicated by the unit vector of red and blue lines; and the metric and the average of adversarial margins ($\bar{d}_E(\boldsymbol{x}_i, \boldsymbol{x}_{i,\min})$) are given in the caption. The objective of LMNN is to satisfy the distance margin. Thus, it expands the distance in both directions. Moreover, since the 1st direction has a small separability in the original instance space, this direction is assigned with a larger weight. In contrast, LMNN-PL controls the scale of $\boldsymbol{M}$. Moreover, a notable difference is that the 2nd direction is assigned with a larger weight than the 1st direction, which is again caused by the small separability in the 1st direction. As any perturbation in the 1st direction is highly likely to result in a misclassification, the proposed method diverts more attention to robust features, i.e., the 2nd direction. Due to the easiness of the task, all metrics lead to the same classification accuracy of 99.09% on a separate test set.

In the second example, we simulate a 3-D binary classification dataset, as shown in Fig. 3(a). Each class includes 100 instances. The first two dimensions are drawn from multivariate Gaussian distributions with $\boldsymbol{\mu}_p = [0.45, 0.45]$, $\boldsymbol{\mu}_n = [-0.45, -0.45]$, $\boldsymbol{\Sigma}_p = \boldsymbol{\Sigma}_n = \left[\begin{smallmatrix} 1 & -0.9 \\ -0.9 & 1 \end{smallmatrix}\right]$; the third dimension equals the sum of the first two dimensions, plus white Gaussian noise with standard deviation of 0.01. By design, the dataset exhibits the problem of strong multicollinearity. This issue has little influence on LMNN as the data is nearly separable in all directions. However, it will affect the adversarial margin. Specifically, if the metric assigns equal weights to all dimensions, then the perturbation should be small in all directions so as to guarantee that the perturbed instance stays on the correct side of the decision boundary. In contrast, if the metric assigns weights only to the third dimension, then the perturbation in the first two dimensions will not cause any change in the learned feature space and hence a larger magnitude of perturbation can be tolerated. This expectation is supported by the empirical result in Fig. 3(c), where the distance in the third dimension is more important than the first two dimensions. LMNN achieves an accuracy of 95.50% and our method achieves an accuracy of 96.00%.

In summary, our method learns a discriminative metric, and meanwhile, imposes a data-dependent regularization on the metric. It also achieves larger adversarial margins than LMNN, demonstrating the effectiveness of the proposed perturbation loss.

B. Experiments on UCI Data

1) Data Description and Experimental Setting: In this experiment, we study 13 datasets from UCI machine learning repository [56]. Information on sample size, feature dimension and class information is listed in Table V in the supplementary material. All datasets are preprocessed with mean-centering and standardization, followed by L_2 normalization to unit length. To evaluate the performance, we use 70%–30% training-test partitions and report the average result over 20 rounds of random split. The only exception is the Credit dataset, where we only run the experiment once as the sample size is relatively large.

We evaluate the effectiveness of the proposed perturbation loss by incorporating it into LMNN, SCML, and ProxyNCA++ (abbreviated to PNCA); the resulting methods are denoted by LMNN-PL, SCML-PL, and PNCA-PL, respectively. In addition, we conduct a thorough study by setting LMNN as the backbone and comparing LMNN-PL with two types of methods. First, we consider different regularizers on $\boldsymbol{M}$. Specifically, we replace the regularizer in LMNN from $\sum_{\mathcal{S}} d_{\boldsymbol{M}}^2(\boldsymbol{x}_i, \boldsymbol{x}_j)$ to the log-determinant divergence (LDD) [12], which encourages learning a metric toward the identity matrix, the capped trace norm (CAP) [25], which encourages a low-rank matrix, and the SN, which has been used to improve adversarial robustness of deep neural networks [57]. Second, we compare with the robust metric learning method DRIFT [41], which models the perturbation distribution explicitly.

Hyperparameters of LMNN-PL are tuned via random search [58]. We randomly sample 50 sets of values from the following ranges: $\mu \in U(0.1, 0.9)$, $\tau \in U(0, P_{90\%}\{d_{\boldsymbol{E}}(\boldsymbol{x}_i, \boldsymbol{x}_{i,\min})\})$, $\lambda \in U(0, 4/\tau^2)$. $U(a, b)$ denotes the uniform distribution. $P_{k\%}\{d_{\boldsymbol{E}}(\boldsymbol{x}_i, \boldsymbol{x}_{i,\min})\}$ denotes the kth percentile of $d_{\boldsymbol{E}}(\boldsymbol{x}_i, \boldsymbol{x}_{i,\min})$, where the distance is calculated for all i in the triplet constraints with respect to the Euclidean distance. Setting the upper bound of the desired margin τ via the percentile avoids unnecessary large values, matching our intention to enlarge the adversarial margin primarily for hard instances. The upper bound of the weight parameter λ depends on the realization of τ to ensure that magnitudes of perturbation loss and LMNN loss are at the same level. The optimal hyperparameters from fivefold cross-validation on the training data or a separate validation set are used to learn the metric. SCML-PL and PNCA-PL are tuned in a similar manner. More details on the training procedure of the proposed and other methods are given in Section IX-B in the supplementary material. The MATLAB code for our method is available at http://github.com/xyang6/LMNNPL.

For LMNN-based and SCML-based methods, we use 3NN as the classifier; for PNCA-based methods, we use the nearest prototype classifier. Classification accuracy is used as the evaluation criterion, except for two highly imbalanced datasets (Ecoli and Yeast), G-means is used.

2) Evaluation on Classification Performance: Table I reports the mean value and standard deviation of classification accuracy or G-means for imbalanced datasets (indicated by an asterisk). LMNN-PL outperforms LMNN on 12 out of 13 datasets. Among the methods with LMNN as the backbone, our method achieves the highest accuracy on eight datasets and second highest accuracy on the four datasets. These experimental results demonstrate the benefit of perturbation loss to generalization of the learned metric. Similarly, we see that SCML-CL outperforms or performs equally well with SCML on nine datasets. The advantage of PNCA-PL becomes less distinct as it is superior to PNCA only on seven datasets. However, this is fairly reasonable as the decision boundary formed by very few proxies is much smoother than the one from 3NN and hence the method is less likely to overfit to training data.

3) Investigation Into Robustness: To test robustness, we add zero-mean Gaussian noise with a diagonal covariance matrix and equal variances to test data; the noise intensity is controlled via the SNR and chosen as 5 dB. In addition, considering the small sample size of UCI datasets, we augment test data by adding multiple rounds of random noise until its size reaches 10 000. As shown in Table II, the proposed methods achieve higher classification accuracy or G-means than the corresponding baselines on almost all datasets. Moreover, LMNN-PL is superior to existing regularization techniques or robust metric learning methods on at least nine datasets. These results clearly demonstrate the efficacy of adding perturbation loss for improving robustness against instance perturbation. Additional experiments with other noise types and intensities are reported in Section IX-C in the supplementary material, where we observe similar advantages of the proposed loss.

C. Experiments on High-Dimensional Data

As mentioned in Remark 3, we extend LMNN-PL for high-dimensional data with PCA being used as a preprocessing step. To verify its effectiveness, we test it on the following three datasets.

1) Isolet [56]: The dataset is a spoken letter database and is available from UCI. It includes 7797 instances,

TABLE I

CLASSIFICATION ACCURACY (OR G-MEANS INDICATED BY AN ASTERISK NEXT TO THE DATASET NAME; MEAN±STANDARD DEVIATION) OF 3NN WITH DIFFERENT METRIC LEARNING METHODS ON CLEAN DATASETS

		LMNN-based						SCML-based		PNCA-based	
Dataset	Euclidean	LMNN	LDD	CAP	SN	DRIFT	LMNN-PL	SCML	SCML-PL	PNCA	PNCA-PL
Australian	82.76±2.38	83.70±2.43	84.18±2.37	83.97±2.45	83.77±2.50	**84.47±2.02**	**84.47±1.63**	**84.76±2.08**	84.42±2.18	**86.30±1.82**	86.15±2.24
Breast cancer	97.17±1.33	**97.12±1.25**	96.95±1.51	97.00±1.08	97.05±1.32	96.98±1.16	97.02±1.30	97.00±1.09	**97.07±1.24**	97.02±1.30	**97.05±1.24**
Ecoli*	85.86±8.10	86.42±7.94	85.29±9.78	83.54±10.09	86.44±7.95	86.45±6.54	**87.04±7.38**	85.53±7.06	**86.69±6.58**	84.80±6.84	**85.54±5.60**
Fourclass	75.12±2.35	75.10±2.31	**75.15±2.32**	75.02±2.48	75.10±2.31	75.08±2.34	75.12±2.35	75.10±2.27	**75.12±2.35**	**75.39±2.21**	72.97±5.36
Haberman	72.25±4.41	72.19±3.89	72.42±3.95	71.52±3.54	72.30±4.57	72.02±3.94	**72.64±4.29**	**72.75±3.79**	72.36±4.38	75.28±3.55	**75.67±3.85**
Iris	87.11±4.92	87.11±5.08	**87.67±4.70**	86.67±5.49	87.22±5.24	85.89±4.46	87.33±4.73	86.89±6.40	**87.44±5.31**	**84.44±6.29**	83.22±5.97
Segment	94.79±0.65	95.31±0.89	95.58±0.81	95.51±0.70	95.38±0.83	**95.75±0.65**	95.64±0.83	92.61±6.65	**93.95±1.47**	**94.73±0.93**	94.52±0.99
Sonar	85.16±4.19	86.67±4.10	87.22±3.90	87.22±4.38	86.67±4.04	86.19±4.43	**87.78±3.53**	82.38±4.15	**84.13±4.61**	83.25±5.95	**83.65±4.83**
Voting	93.78±1.76	95.80±1.78	95.80±1.41	95.92±1.45	95.84±1.74	95.31±1.32	**96.15±1.56**	95.84±1.58	**96.26±1.28**	**95.84±1.65**	95.65±1.66
WDBC	96.29±1.61	96.99±1.30	96.96±1.43	96.99±1.51	96.93±1.34	96.70±1.16	**97.13±1.33**	**97.25±1.30**	**97.25±1.52**	**97.37±1.49**	**97.37±0.94**
Wine	95.28±2.36	97.31±1.94	96.67±1.76	96.85±2.26	97.41±1.84	**97.69±1.79**	**97.69±1.89**	**97.69±1.79**	97.22±2.04	97.04±2.71	**97.22±1.95**
Yeast*	70.33±10.50	69.84±10.26	70.26±10.51	70.29±10.52	69.86±10.29	**70.32±10.51**	**70.32±10.51**	68.81±11.35	**69.90±10.35**	66.01±13.21	**69.41±10.36**
Credit	76.40	76.41	76.68	**76.96**	76.50	76.87	76.89	**76.45**	76.29	**81.15**	81.07
# outperform	-	12	11	12	12	12	-	9	-	7	-

For methods with LMNN as the backbone, the best ones are shown in bold and the second best ones are underlined; for methods with SCML or PNCA as the backbone, the best ones are shown in bold. '# outperform' counts the number of datasets where LMNN-PL (SCML-PL, PNCA-PL resp.) outperforms or performs equally well with LMNN-based (SCML, PNCA resp.) methods.

TABLE II

CLASSIFICATION ACCURACY (OR G-MEANS INDICATED BY AN ASTERISK) OF 3NN NOISE-CONTAMINATED DATASETS. GAUSSIAN NOISE WITH AN SNR OF 5 dB IS ADDED TO TEST DATA

		LMNN-based						SCML-based		PNCA-based	
Dataset	Euclidean	LMNN	LDD	CAP	SN	DRIFT	LMNN-PL	SCML	SCML-PL	PNCA	PNCA-PL
Australian	82.28±1.67	82.46±1.58	83.02±1.58	82.36±1.56	82.56±1.54	82.58±1.45	**83.50±1.56**	82.93±1.65	**83.42±1.68**	82.79±2.37	**83.66±2.28**
Breast cancer	96.79±1.05	96.25±1.09	96.69±1.09	96.35±1.02	96.29±1.11	96.66±1.00	**96.71±1.08**	96.40±1.05	**96.65±1.06**	96.20±1.37	**96.77±1.14**
Ecoli*	79.95±7.67	74.96±7.16	79.13±7.67	74.46±9.36	75.19±7.23	77.86±7.76	**80.04±7.51**	76.49±6.78	**78.38±7.92**	75.52±6.08	**77.04±6.20**
Fourclass	69.11±1.12	67.62±1.23	68.77±1.14	67.63±1.12	68.55±1.30	**69.03±1.13**	69.01±1.17	68.07±1.16	**68.86±1.06**	**70.42±2.17**	69.39±4.60
Haberman	69.93±1.88	69.84±1.79	**69.92±1.87**	69.23±2.00	69.90±1.88	69.09±2.49	69.89±1.90	69.65±1.63	**69.88±1.83**	**74.32±3.22**	**74.32±3.10**
Iris	79.75±3.26	78.61±2.97	78.87±3.16	77.79±3.27	78.70±3.08	78.43±3.09	**79.04±3.09**	78.16±3.58	**79.01±3.12**	77.95±4.53	**78.20±3.90**
Segment	88.18±0.64	81.02±3.55	86.15±1.26	85.34±2.47	82.10±3.41	**86.63±1.09**	84.72±2.62	60.18±9.73	**61.33±9.05**	78.27±2.83	**80.28±3.35**
Sonar	83.47±3.21	83.56±4.27	**86.18±2.95**	85.41±2.82	83.52±4.28	84.65±3.30	85.00±3.15	77.01±4.23	**79.49±3.80**	80.74±4.36	**81.74±3.43**
Voting	93.19±1.15	94.00±1.00	94.25±1.14	94.37±1.17	94.06±1.00	93.95±1.12	**94.64±1.21**	93.99±1.15	**94.64±1.09**	92.61±1.64	**93.46±1.77**
WDBC	95.92±1.30	91.71±1.90	**96.30±0.94**	96.16±1.08	92.46±1.80	96.04±0.86	96.11±0.88	95.74±1.30	**96.21±1.16**	96.03±1.54	**96.22±1.15**
Wine	94.20±1.46	93.33±1.63	94.03±1.39	93.97±1.47	93.45±1.70	**94.66±1.15**	94.51±1.20	94.01±1.56	**94.61±1.32**	**94.19±1.94**	93.48±1.48
Yeast*	69.36±10.47	54.13±8.24	68.62±10.43	66.48±10.18	55.49±9.45	69.64±10.49	**69.82±10.44**	55.96±7.64	**60.47±10.39**	61.41±17.97	**63.59±18.33**
Credit	76.28	76.16	76.22	76.05	76.30	**76.37**	76.15	**75.93**	75.55	78.24	**79.13**
# outperform	-	12	9	10	11	9	-	12	-	11	-

grouped into four training sets and one test set. We apply PCA to reduce the feature dimension from 617 to 170, accounting for 95% of total variance. All methods are trained four times, one time on each training set, and evaluated on the pregiven test set.

2) MNIST-2k [59]: The dataset includes the first 2000 training images and first 2000 test images of the MNIST database. PCA is applied to reduce the dimension from 784 to 141, retaining 95% of total variance. All methods are trained and tested once on the pregiven training/test partition.
3) APS Failure [56]: This is a multivariate dataset with a highly imbalanced class distribution. The training set includes 60 000 instances, among which 1000 belong to the positive class. The test set includes 16 000 instances with 375 positive ones. The training set is further split into 40 000 instances for training and 20 000 instances for selecting hyperparameters. All methods are tested once on the test set. Applying PCA reduces the feature dimension from 161 to 79. Due to the large sample size, we only evaluate LMNN and LMNN-PL on this dataset.

In addition to aforementioned methods, we introduce CAP-PL, which comprises the triplet loss of LMNN, the regularizer of CAP, and the proposed perturbation loss. CAP enforces $\boldsymbol{M}$ to be low-rank, which is a suitable constraint for high-dimensional data. With the inclusion of perturbation loss, we expect the learned compact metric to be more robust to perturbation. For a fair comparison, in CAP-PL, we use the same rank and regularization weight as CAP, and tune τ, λ from ten randomly sampled sets of values.

Table III compares the generalization and robustness performance of LMNN, CAP, SCML, and our method; the generalization performance of other methods are inferior to LMNN-PL and reported in Table VIII in the supplementary material. First, on all three original datasets, our method achieves better performance than the baseline methods, validating its efficacy in improving the generalization ability of the learned metric. Second, when the SNR is 20 dB, the average perturbation size is smaller than the average adversarial

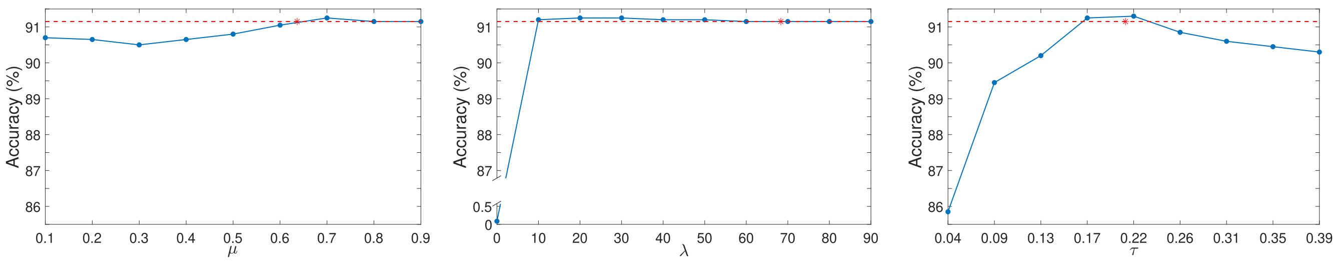


Fig. 4. Sensitivity of LMNN-PL to hyperparameters (indicated by the straight line). Optimal accuracy and parameter value found via CV are indicated by the dashed line and asterisk, respectively.

TABLE III
GENERALIZATION AND ROBUSTNESS OF SELECTED METRIC LEARNING METHODS ON HIGH-DIMENSIONAL DATASETS

Isolet						
Method	Clean	IG,SNR=20 (0.0809)	IG,SNR=5 (0.4233)	AG,SNR=20 (0.0588)	AG,SNR=5 (0.3181)	Adv. margin
LMNN	90.14±4.45	90.09±4.15	86.02±3.48	90.17±4.03	87.81±3.87	0.1095
LMNN-PL	91.08±3.71	91.02±3.77	87.91±3.30	91.05±3.73	89.40±3.76	0.1249
SCML	90.73±4.10	90.33±4.21	86.50±4.18	90.51±4.14	88.50±3.71	0.0683
SCML-PL	90.83±4.16	90.67±4.12	86.55±3.75	90.83±4.16	88.41±4.07	0.0822
CAP	91.05±3.66	91.13±3.85	88.97±4.00	91.10±3.73	89.90±3.87	0.1514
CAP-PL	91.58±3.96	91.52±3.86	89.91±3.74	91.47±3.91	90.65±3.73	0.1559

MNIST						
Method	Clean	IG,SNR=20 (0.0540)	IG,SNR=5 (0.2939)	AG,SNR=20 (0.0649)	AG,SNR=5 (0.3482)	Adv. margin
LMNN	90.55	90.00	88.40	90.10	88.40	0.1528
LMNN-PL	91.15	91.35	90.80	91.45	90.35	0.2235
SCML	88.95	88.75	87.35	88.85	86.45	0.1217
SCML-PL	89.15	89.20	88.50	89.35	88.05	0.1432
CAP	91.65	91.80	91.40	91.80	90.70	0.2219
CAP-PL	92.00	91.90	90.85	91.95	90.65	0.2264

APS Failure						
Method	Clean	IG,SNR=20 (0.0906)	IG,SNR=5 (0.5097)	AG,SNR=20 (0.0996)	AG,SNR=5 (0.5604)	Adv. margin
LMNN	80.69	80.20	75.66	81.18	75.13	0.1773
LMNN-PL	80.89	82.15	74.13	82.33	77.92	0.2583

Columns 3-6 report methods' robustness against isotropic Gaussian noise (IG) and anisotropic Gaussian noise (AG). Values in brackets give the average perturbation size, calculated as the mean value of the L_2-norm of noises ($\|\Delta\boldsymbol{x}_i\|_2$).

TABLE IV
AVERAGE TRAINING TIME (IN SECONDS) OF LMNN-BASED METHODS

	LMNN	LDD	CAP	SN	DRIFT	LMNN-PL
Australian	13.44	0.83	3.07	7.60	1.00	2.15
Segment	27.48	10.45	11.47	24.66	5.12	19.54
Sonar	4.93	4.08	4.65	30.39	0.92	6.75
WDBC	9.38	2.94	5.22	16.54	5.12	8.17
Credit	724.42	34.65	115.22	966.63	130.36	138.63
Isolet	339.57	207.69	176.50	540.26	NA	190.55
MNIST	369.55	68.98	289.18	197.50	37.51	391.04

margin. In this case, our method maintains its superiority. When the SNR is 5 dB, the average perturbation size is larger than the average adversarial margin. Nonetheless, our method produces even larger gain in classification performance for LMNN on all datasets except APS Failure with the Gaussian noise, for SCML on MNIST and on Isolet with the Gaussian noise, for CAP on Isolet. These results suggest that adversarial margin is indeed a contributing factor in enhancing robustness. Third, CAP-PL obtains higher accuracy on both clean and noise-contaminated data than LMNN-PL. This supports our discussion in Section III that regularization and perturbation loss impose different requirements on $\boldsymbol{M}$ and combining them has the potential for learning a more effective distance metric.

D. Computational Cost

We now analyze the computational complexity of LMNN-PL. According to (6), our method requires additional calculations on $d^2_{\boldsymbol{M}^2}(\boldsymbol{x}_j, \boldsymbol{x}_l)$ and $\boldsymbol{M}\boldsymbol{X}_{jl}$. Given n training instances, k target neighbors and p features, the computational complexities of $d^2_{\boldsymbol{M}^2}(\boldsymbol{x}_j, \boldsymbol{x}_l)$ and $\boldsymbol{M}\boldsymbol{X}_{jl}$ are $O(np^2+n^2p)$ and $O(n^2p^2)$, respectively. The total complexity is $O(p^3+n^2p^2+kn^2p)$, same as that of LMNN.

Table IV compares the running time of LMNN-based methods on five UCI datasets that are large in sample size or in dimensionality and two high-dimensional datasets. The computational cost of our method is comparable to LMNN.

E. Parameter Sensitivity

The proposed LMNN-PL includes three hyperparameters – μ for the weight of similarity constraints, λ for the weight of the perturbation loss, and τ for the desired adversarial margin. We investigate their influences on the classification performance by varying one hyperparameter and fixing the other two at their optimal values. Fig. 4 shows the accuracy on MNIST evaluated over the range of the hyperparameter. The performance changes smoothly with respect to μ. It is stable over a wide range of λ. When λ equals 0, LMNN-PL fails to learn a metric and returns a zero matrix. The performance is most affected by τ. Indeed, τ plays the central role in LMNN-PL as it determines the distribution of adversarial margins. A small value of τ has little influence on the objective function as the adversarial margin of most instances may already exceed it before optimization, and a large value may greatly reduce the number of triplets that satisfy the loss condition in the definition of pseudo-robustness [i.e., $\hat{n}(t_s)$ in Theorem 1]. Therefore, we shall strive to search for its optimal value.

V. CONCLUSION

In this article, we propose to enhance the robustness and generalization of distance metrics. This is easily achievable by taking advantage of the linear transformation induced by the

Mahalanobis distance. Specifically, we find an explicit formula for the adversarial margin, which is defined as the Euclidean distance between benign instances and their closest adversarial examples, and advocate to enlarge it through penalizing the perturbation loss designed on the basis of the derivation. Experiments verify that our method effectively enlarges the adversarial margin, sustains classification excellence, and enhances robustness to instance perturbation. The proposed loss term is generic in nature and could be readily embedded in other Mahalanobis-based metric learning methods. In the future, we will consider extending the idea to metric learning methods with nonlinear feature extraction and/or nonlinear metric learning methods.

References

[1] E. Xing, M. Jordan, S. J. Russell, and A. Ng, "Distance metric learning with application to clustering with side-information," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 15, 2002, pp. 1–8.

[2] B. Kulis, "Metric learning: A survey," *Found. Trends Mach. Learn.*, vol. 5, no. 4, pp. 287–364, 2013.

[3] A. Bellet, A. Habrard, and M. Sebban, "Metric learning," *Synth. Lectures Artif. Intell. Mach. Learn.*, vol. 9, no. 1, pp. 1–151, 2015.

[4] D. Li and Y. Tian, "Survey and experimental study on metric learning methods," *Neural Netw.*, vol. 105, pp. 447–462, Sep. 2018.

[5] S. Xiang, F. Nie, and C. Zhang, "Learning a Mahalanobis distance metric for data clustering and classification," *Pattern Recognit.*, vol. 41, no. 12, pp. 3600–3612, 2008.

[6] B. McFee and G. R. Lanckriet, "Metric learning to rank," in *Proc. Int. Conf. Mach. Learn.*, 2010, pp. 1–8.

[7] J. Yang, D. She, Y.-K. Lai, and M.-H. Yang, "Retrieving and classifying affective images via deep metric learning," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 1–8.

[8] M. Kostinger, M. Hirzer, P. Wohlhart, P. M. Roth, and H. Bischof, "Large scale metric learning from equivalence constraints," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2288–2295.

[9] J. Hu, J. Lu, Y.-P. Tan, J. Yuan, and J. Zhou, "Local large-margin multi-metric learning for face and kinship verification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 8, pp. 1875–1891, Aug. 2018.

[10] L. Ma, X. Yang, and D. Tao, "Person re-identification over camera networks using multi-task distance metric learning," *IEEE Trans. Image Process.*, vol. 23, no. 8, pp. 3656–3670, Aug. 2014.

[11] G. Zou, G. Fu, X. Peng, Y. Liu, M. Gao, and Z. Liu, "Person re-identification based on metric learning: A survey," *Multimedia Tools Appl.*, vol. 80, pp. 26855–26888, May 2021.

[12] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, "Information-theoretic metric learning," in *Proc. 24th Int. Conf. Mach. Learn. (ICML)*, 2007, pp. 209–216.

[13] J. T. Kwok and I. W. Tsang, "Learning with idealized kernels," in *Proc. Int. Conf. Mach. Learn.*, 2003, pp. 1–8.

[14] K. Q. Weinberger, J. Blitzer, and L. Saul, "Distance metric learning for large margin nearest neighbor classification," *J. Mach. Learn. Res.*, vol. 10, pp. 207–244, Jul. 2009.

[15] Y. Shi, A. Bellet, and F. Sha, "Sparse compositional metric learning," in *Proc. AAAI Conf. Artif. Intell.*, 2014, pp. 1–7.

[16] K. Song, F. Nie, J. Han, and X. Li, "Parameter free large margin nearest neighbor for distance metric learning," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 1–7.

[17] M. Dong, Y. Wang, X. Yang, and J.-H. Xue, "Learning local metrics and influential regions for classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 6, pp. 1522–1529, Jun. 2020.

[18] M. T. Law, N. Thome, and M. Cord, "Quadruplet-wise image similarity learning," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 249–256.

[19] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 1–9.

[20] C. Szegedy *et al.*, "Intriguing properties of neural networks," 2013, *arXiv:1312.6199*.

[21] H. Xu and S. Mannor, "Robustness and generalization," *Mach. Learn.*, vol. 86, no. 3, pp. 391–423, 2012.

[22] E. W. Teh, T. DeVries, and G. W. Taylor, "ProxyNCA++: Revisiting and revitalizing proxy neighborhood component analysis," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 448–464.

[23] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 4080–4090.

[24] S. Kim, D. Kim, M. Cho, and S. Kwak, "Proxy anchor loss for deep metric learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 3238–3247.

[25] Z. Huo, F. Nie, and H. Huang, "Robust and effective metric learning using capped trace norm: Metric learning via capped trace norm," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 1605–1614.

[26] A. Bellet and A. Habrard, "Robustness and generalization for metric learning," *Neurocomputing*, vol. 151, pp. 259–267, Mar. 2015.

[27] L. Floridi, "Establishing the rules for building trustworthy AI," *Nature Mach. Intell.*, vol. 1, no. 6, pp. 261–262, Jun. 2019.

[28] Q. Suo, W. Zhong, F. Ma, Y. Ye, M. Huai, and A. Zhang, "Multi-task sparse metric learning for monitoring patient similarity progression," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2018, pp. 477–486.

[29] R. Aliakbarisani, A. Ghasemi, and S. Felix Wu, "A data-driven metric learning-based scheme for unsupervised network anomaly detection," *Comput. Electr. Eng.*, vol. 73, pp. 71–83, Jan. 2019.

[30] M. Guillaumin, J. Verbeek, and C. Schmid, "Is that you? Metric learning approaches for face identification," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 498–505.

[31] X. Ben *et al.*, "On the distance metric learning between cross-domain gaits," *Neurocomputing*, vol. 208, pp. 153–164, Oct. 2016.

[32] I. Omara, H. Zhang, F. Wang, A. Hagag, X. Li, and W. Zuo, "Metric learning with dynamically generated pairwise constraints for ear recognition," *Information*, vol. 9, no. 9, p. 215, Aug. 2018.

[33] R. Jin, S. Wang, and Y. Zhou, "Regularized distance metric learning: Theory and algorithm," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 1–9.

[34] D. Lim, G. Lanckriet, and B. McFee, "Robust structural metric learning," in *Proc. 30th Int. Conf. Mach. Learn.*, 2013, pp. 615–623.

[35] M. T. Law, N. Thome, and M. Cord, "Fantope regularization in metric learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1051–1058.

[36] L. Luo and H. Huang, "Matrix variate Gaussian mixture distribution steered robust metric learning," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 1–8.

[37] K. Liu, L. Brand, H. Wang, and F. Nie, "Learning robust distance metric with side information via ratio minimization of orthogonally constrained $\ell_{2,1}$-norm distances," in *Proc. Int. Joint Conf. Artif. Intell.*, 2019, pp. 1–7.

[38] H. Wang, F. Nie, and H. Huang, "Robust distance metric learning via simultaneous ℓ_1-norm minimization and maximization," in *Proc. 31st Int. Conf. Mach. Learn.*, 2014, pp. 1836–1844.

[39] T. Yuan, W. Deng, J. Tang, Y. Tang, and B. Chen, "Signal-to-noise ratio: A robust distance metric for deep metric learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4815–4824.

[40] J. Xu, L. Luo, C. Deng, and H. Huang, "New robust metric learning model using maximum correntropy criterion," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 2555–2564.

[41] H.-J. Ye, D.-C. Zhan, X.-M. Si, and Y. Jiang, "Learning Mahalanobis distance metric: Considering instance disturbance helps," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 3315–3321.

[42] Q. Qian, J. Tang, H. Li, S. Zhu, and R. Jin, "Large-scale distance metric learning with uncertainty," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8542–8550.

[43] S. Chen, C. Gong, J. Yang, X. Li, Y. Wei, and J. Li, "Adversarial metric learning," in *Proc. Int. Joint Conf. Artif. Intell.*, 2018, pp. 1–8.

[44] Y. Duan, W. Zheng, X. Lin, J. Lu, and J. Zhou, "Deep adversarial metric learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2780–2789.

[45] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski, *Robust Optimization*. Princeton, NJ, USA: Princeton Univ. Press, 2009.

[46] T. K. Panum, Z. Wang, P. Kan, E. Fernandes, and S. Jha. (2021). *Adversarial Deep Metric Learning*. [Online]. Available: https://openreview.net/forum?id=Kzg0XmE6mxu

[47] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[48] S. An, M. Hayat, S. H. Khan, M. Bennamoun, F. Boussaid, and F. Sohel, "Contractive rectifier networks for nonlinear maximum margin classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2515–2523.

[49] G. Elsayed, D. Krishnan, H. Mobahi, K. Regan, and S. Bengio, "Large margin deep networks for classification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 1–11.

[50] Z. Yan, Y. Guo, and C. Zhang, "Adversarial margin maximization networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 4, pp. 1129–1139, Apr. 2021.

[51] G. W. Ding, Y. Sharma, K. Y. C. Lui, and R. Huang, "MMA training: Direct input space margin maximization through adversarial training," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–28.

[52] N. Papernot, P. McDaniel, and I. Goodfellow, "Transferability in machine learning: From phenomena to black-box attacks using adversarial samples," 2016, *arXiv:1605.07277*.

[53] C. Sitawarin and D. Wagner, "On the robustness of deep K-nearest neighbors," in *Proc. IEEE Secur. Privacy Workshops (SPW)*, May 2019, pp. 1–7.

[54] L. Wang, X. Liu, J. Yi, Z.-H. Zhou, and C.-J. Hsieh, "Evaluating the robustness of nearest neighbor classifiers: A primal-dual perspective," 2019, *arXiv:1906.03972*.

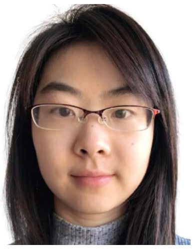
[55] Y.-Y. Yang, C. Rashtchian, Y. Wang, and K. Chaudhuri, "Robustness for non-parametric classification: A generic attack and defense," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2020, pp. 941–951.

[56] D. Dua and C. Graff. (2017). *UCI Machine Learning Repository*. [Online]. Available: http://archive.ics.uci.edu/ml

[57] P. L. Bartlett, D. J. Foster, and M. J. Telgarsky, "Spectrally-normalized margin bounds for neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 6240–6249.

[58] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, pp. 281–305, Feb. 2012.

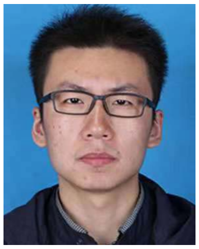
[59] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1548–1560, Aug. 2011. [Online]. Available: http://www.cad.zju.edu.cn/home/dengcai/Data/MLData.html



Xiaochen Yang received the B.Sc. degree in actuarial science from the London School of Economics and Political Science, London, U.K., in 2013, and the Ph.D. degree in statistical science from University College London, London, in 2020.

She is currently a Lecturer with the School of Mathematics and Statistics, University of Glasgow, Glasgow, U.K. Her research interests include statistical classification, metric learning, and hyperspectral image analysis.

Dr. Yang is an Associate Editor of *Neurocomputing*.



Yiwen Guo received the B.E. degree from Wuhan University, Wuhan, China, in 2011, and the Ph.D. degree from Tsinghua University, Beijing, China, in 2016.

He was a Research Scientist at ByteDance AI Lab, Beijing. Prior to this, he was a Staff Research Scientist at Intel Labs China, Beijing. His current research interests include computer vision, pattern recognition, and machine learning.

Mingzhi Dong received the B.Eng. and M.Eng. degrees from the Beijing University of Posts and Telecommunications, Beijing, China, in 2010 and 2013, respectively, and the Ph.D. degree from University College London, London, U.K., in 2019.

He is currently a Post-Doctoral Research Fellow with Fudan University, Shanghai, China. His research interests include metric learning, reinforcement learning, and weak supervised learning.

Jing-Hao Xue (Senior Member, IEEE) received the Dr.Eng. degree in signal and information processing from Tsinghua University, Beijing, China, in 1998, and the Ph.D. degree in statistics from the University of Glasgow, Glasgow, U.K., in 2008.

He is currently a Professor with the Department of Statistical Science, University College London, London, U.K. His research interests include multivariate and high-dimensional data analysis, statistical pattern recognition, machine learning, and image processing.

Dr. Xue is an Associate Editor of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE TRANSACTIONS ON CYBERNETICS, and IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS.