Metric Learning for Categorical and Ambiguous Features: An Adversarial Method

Xiaochen Yang[1,⋆][0000−0002−9299−5951], Mingzhi Dong[1,⋆], Yiwen Guo[2][0000−0002−0709−4877], and Jing-Hao Xue(✉)[1][0000−0003−1174−610X]

[1] University College London, London, UK
xiaochen.yang.16@ucl.ac.uk,mingzhidong@gmail.com,jinghao.xue@ucl.ac.uk
[2] ByteDance AI Lab, Beijing, China
guoyiwen.ai@bytedance.com

Abstract. Metric learning learns a distance metric from data and has significantly improved the classification accuracy of distance-based classifiers such as k-nearest neighbors. However, metric learning has rarely been applied to categorical data, which are prevalent in health and social sciences, but inherently difficult to classify due to high feature ambiguity and small sample size. More specifically, ambiguity arises as the boundaries between ordinal or nominal levels are not always sharply defined. In this paper, we mitigate the impact of feature ambiguity by considering the worst-case perturbation of each instance and propose to learn the Mahalanobis distance through adversarial training. The geometric interpretation shows that our method dynamically divides the instance space into three regions and exploits the information on the "adversarially vulnerable" region. This information, which has not been considered in previous methods, makes our method more suitable than them for small-sized data. Moreover, we establish the generalization bound for a general form of adversarial training. It suggests that the sample complexity rate remains at the same order as that of standard training only if the Mahalanobis distance is regularized with the elementwise 1-norm. Experiments on ordinal and mixed ordinal-and-nominal datasets demonstrate the effectiveness of the proposed method when encountering the problems of high feature ambiguity and small sample size.

Keywords: Metric learning · Categorical data · Adversarial training.

1 Introduction

The k-nearest neighbors (kNN) algorithm is a classical and widely used classification method by virtue of the nonparametric nature, interpretability, and flexibility in defining the distance between instances [6]. As a discriminative distance function can boost kNN's performance, the idea of learning a task-specific metric from the data was pioneered in [33], which formulates the task of learning

⋆ Equal contribution

a generalized Mahalanobis distance as a convex optimization problem. Thereafter, many global [31], local [30], kernelized [25] and deep [16] metric learning methods have been proposed to further improve the discriminability. While these methods are effective, they have rarely been applied to data with ordinal and nominal features.

Ordinal and nominal variables (i.e. features) are subsumed under the data type of categorical variables that have measurement scales consisting of a set of categories [1]. Categorical variables with ordered scales are called ordinal variables and the ones with unordered scales are called nominal variables. For example, when collecting a film survey, the audience review (poor, fair, good, excellent) is an ordinal variable; the genre of favorite films (action, comedy, drama, horror) is a nominal variable. Both types of variables occur frequently in social and health sciences and also arise in education and marketing.

Classifying ordinal and nominal variables faces at least the following three challenges. First, a simple way of representing these variables is to encode them as integers and then treat them as real-valued continuous variables. However, for an ordinal variable, the difference between two integers does not necessarily reflect the distance between the two ordinal levels, and for a nominal variable, the difference between two integers is meaningless for two nominal levels. Another way of representing ordinal and nominal variables is to encode each categorical variable into a set of binary variables, such as through dummy coding. This conversion avoids the above problems, and allows for the modeling of interactions between different levels of the variable. However, it inevitably increases the feature dimension, and the effect is dramatic when each variable has a large number of levels. The second challenge is the ambiguity in ordinal variables. For example, in the example of audience review, the boundaries between levels such as 'good' and 'excellent' are not sharply defined, thereby causing ambiguity. This issue is less common in nominal variables, but it still appears when some categories have overlapping characteristics. Third, for economic and ethical reasons, the categorical data collected in social and health sciences often have a small sample size. This places a restriction on model complexity since a complex model may overfit and generalize poorly to unseen data.

This paper focuses on adapting metric learning methods for ordinal and nominal features that could work on both types of encoded data, i.e. as integer variables and as dummy variables, and address the feature ambiguity and small-sized problems. Firstly, to mitigate the impact of feature ambiguity, we propose to consider the worst-case perturbation of each instance within a deliberately designed constraint set and learn the distance metric via adversarial training. The constraint set takes into account the discrete nature of nominal variables and the ordering nature of ordinal variables. Secondly, we provide a geometric interpretation of the proposed formulation, which suggests that our method dynamically divides the instance space into three regions, namely support region, adversarially vulnerable region, and adversarially robust region. Compared with classical metric learning methods which only uses information on the support region, our method additionally uses information on instances from the adversar-

ially vulnerable region, thereby coping better with the small sample size problem. Thirdly, we prove the generalization bound for a general form of adversarial training. It guarantees that, when regularizing the Mahalanobis distance with the elementwise 1-norm, the sample complexity rate of the proposed method remains at the same order as that of classical methods. Finally, the method is tested on datasets with all ordinal variables and with a mixture of ordinal and nominal variables. It surpasses state-of-the-art methods in cases of high feature ambiguity and small sample size.

2 Related Work

This section briefly reviews distance metrics for categorical data and metric learning methods that consider feature uncertainty.

Distance metrics for categorical data Various distance or similarity measures are proposed for categorical data, mostly for nominal data, in an unsupervised setting. The most common measure is *overlap*, which defines the similarity between two instances $\boldsymbol{x}_1, \boldsymbol{x}_2$ on the ith feature to be 1 if their values are equal and 0 otherwise. Summing up the similarities over all features defines the distance between $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. Based on overlap, many probabilistic or frequency-based measures have been proposed to assign different weights on matches or mismatches, as well as taking into account the occurrence of other feature values [2]. Another class of measures are based on entropy, where the distance contribution of each categorical level depends on the amount of information it holds. Entropy-based measures have been extended in [36] to quantify the order relation of ordinal variables.

In a supervised setting, non-learning approaches use the label information to determine the discriminative power of each feature and adjust the feature weights in distance calculation accordingly [9]. Learning approaches learn the distance between each pair of categorical levels or a mapping function from each level to a real value by minimizing the classification error [8, 32]. More recently, large margin-based metric learning methods have been adapted for ordinal and nominal variables [23, 37]. Building on the assumption that an ordinal variable represents a continuous latent variable that falls into an interval of values, [23] jointly learns the Mahalanobis distance, thresholds of intervals, and parameters of the latent variable distribution. As the number of thresholds is determined by the number of variables and levels within them, the method may involve a large number of parameters and suffer from overfitting. [37] represents the categorical data by computing the interaction between levels, between variables, and between variables and classes, followed by learning the Mahalanobis distance in a kernel space. However, it ignores the natural ordering of ordinal variables.

Metric learning with uncertainty In most metric learning methods, a Mahalanobis distance is optimized such that similar instances become closer with respect to the new metric and dissimilar instances become farther away. As the optimization process is guided by the side information, its effectiveness degrades in the presence

of label noise, outlier samples, and feature uncertainty. Compared with outliers (or influential points in the statistics literature) which account for a small proportion of instances but severely influence the model, feature uncertainty, possibly ensued from ambiguity in the definition of set boundaries, measurement and quantization errors, and data processing of repeated measurements, normally appears as small perturbations but potentially pollutes a large number of instances [26]. Many robust metric learning methods have been proposed to tackle the above problems [29, 27, 28], and here, we only discuss those on feature uncertainty.

One way to handle feature uncertainty is to build an explicit model of perturbation [34, 22]. [34] assumes a perturbation distribution of each instance, replaces the Mahalanobis distance by its expected value, and iteratively learns the distribution and distance metric by minimizing the number of violations of triplet constraints. The method essentially adjusts the constraint on distance margin for each triplet according to its reliableness. Another approach is to learn a distance metric that is less sensitive to feature uncertainty via adversarial training [7, 12]. The method involves two stages. The confusion stage generates adversarial pairs that incur large losses, and the discrimination stage optimizes the distance metric based on these augmented pairs. Originating from robust optimization [17, 24], adversarial training has received considerable attention in recent years as an effective approach to achieving robustness to adversarial examples [20]. In addition, adversarial training is shown to improve the classification accuracy when there is only a limited number of instances available to train the model [3]. While our method shares a similar principle, it differs from existing adversarial metric learning methods in two respects. Firstly, we take the subsequent classification mechanism into consideration when searching for the worst-case perturbation. Derived from triplet constraints, the perturbation is capable of altering the decision of NN classifier. Secondly and more importantly, the loss function in our proposal is designed specifically for ordinal and nominal features with an explicit consideration of their discrete and ordering nature.

3 Methodology

In this section, we propose to model feature ambiguity as a perturbation to the instance and learn the Mahalanobis distance via adversarial training. After introducing notations, we will present the method and its optimization algorithm, followed by a geometric interpretation and a generalization analysis.

3.1 Preliminaries

Let $\boldsymbol{z}^n = \{\boldsymbol{z}_i = (\boldsymbol{x}_i, y_i), i = 1, \ldots, n\}$ denote the training set, where $\boldsymbol{x}_i \in \mathcal{X}$ is the ith training instance associated with label $y_i \in \mathcal{Y} = \{1, \ldots, C\}$; $\boldsymbol{z}_i \in \mathcal{Z}$ is independently and identically distributed according to an unknown distribution $\mathcal{D}$. Suppose each instance includes p features, p^{ord} of which are ordinal variables and $p^{\text{nom}} = p - p^{\text{ord}}$ are nominal variables. Ordinal variables can be encoded as consecutive integers or as a set of binary values. In the integer case, a variable with

p_r levels takes values from $\{1, 2, \ldots, p_r\}$, and the mapping should follow the order relation. In other words, for ordinal levels $O_1 \prec O_2 \prec \cdots \prec O_{p_r}$ with an order relation $\prec$, there is a mapping function $\mathcal{O}$ such that $\mathcal{O}(O_q) = q, q = 1, \ldots, p_r$. In the binary-valued case, ordinal variables are encoded via the *OrderedPartitions* method [15, 23]. For example, an ordinal variable with 3 levels will be encoded as [1,0,0], [1,1,0] and [1,1,1]. Nominal variables are encoded via the *1-of-K* encoding scheme. For example, a nominal variable with 3 levels will be encoded as [1,0,0], [0,1,0] and [0,0,1]. Let P denote the feature dimension after encoding, which equals $p^{\text{ord}} + \sum_{r=1}^{p^{\text{nom}}} p_r$ if ordinal variables are encoded as integers and equals $\sum_{r=1}^{p} p_r$ if they are encoded as a set of binary values.

In this paper, we focus on learning the Mahalanobis distance from triplet-based side information. For any two instances $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathbb{R}^P$, the generalized (squared) Mahalanobis distance is defined as

$$d^2_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i - \boldsymbol{x}_j)^T \boldsymbol{M} (\boldsymbol{x}_i - \boldsymbol{x}_j)$$

where $\boldsymbol{M} \in \mathbb{S}^P_+$ is a $P \times P$ real-valued positive semidefinite (PSD) matrix. A classical triplet-based metric learning method is the large margin nearest neighbors (LMNN) algorithm [31]. It pulls k nearest same-class instances closer and pushes away differently labeled instances by a fixed margin through optimizing the following objective function:

$$\min_{\boldsymbol{M} \in \mathbb{S}^P_+} (1-\mu) \sum_{(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{S}} d^2_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_j) + \mu \sum_{(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_l) \in \mathcal{R}} \left[1 + d^2_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_j) - d^2_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_l)\right]_+, \quad (1)$$

where $[a]_+ = \max(a, 0)$ for $a \in \mathbb{R}$; μ is the trade-off parameter; and

$$\begin{aligned} \mathcal{S} &= \big\{(\boldsymbol{x}_i, \boldsymbol{x}_j) : \boldsymbol{x}_j \in \{k\text{NNs with the same class label of } \boldsymbol{x}_i\}\big\}, \\ \mathcal{R} &= \big\{(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_l) : (\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{S}, y_i \neq y_l\big\}. \end{aligned} \quad (2)$$

$\boldsymbol{x}_j$ is termed the target neighbor of $\boldsymbol{x}_i$ and $\boldsymbol{x}_l$ is termed the impostor.

3.2 Metric Learning with Adversarial Training (MLadv)

The objective function of LMNN (Eq. 1) can be interpreted as minimizing a linear combination between the *empirical risk* $\frac{1}{n}\sum_{i=1}^n \ell(\boldsymbol{x}, y; \boldsymbol{M})$ and the regularizer on $\boldsymbol{M}$; the hinge loss ℓ of $[1 + d^2_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_j) - d^2_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_l)]_+$ separates target neighbors and impostors by a unit margin, and the regularizer is chosen as $\sum_{(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{S}} d^2_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_j)$. To address the issue of feature ambiguity faced by ordinal and nominal variables, we propose to model the unknown ambiguity as a perturbation of $\boldsymbol{x}_i$. Instead of the hinge loss, we consider the worst-case loss within a certain perturbation range and minimize the *adversarial empirical risk*:

$$\min_{\boldsymbol{M}} \frac{1}{n} \sum_{i=1}^n \max_{\boldsymbol{\delta} \in \Delta} \ell(\boldsymbol{x} + \boldsymbol{\delta}, y; \boldsymbol{M}). \quad (3)$$

$\boldsymbol{\delta}$ denotes the perturbation and its form is specified by set Δ. The optimal solution to the inner maximization problem is termed the worst-case perturbation and denoted by $\boldsymbol{\delta}^\star$.

To fit ordinal and nominal variables, we need to incorporate their properties when defining the perturbation set Δ. A typical choice of Δ is the set of ℓ_p-bounded perturbation, i.e. $\|\boldsymbol{\delta}\|_p \leq \varepsilon$, with $p = 1, 2, \infty$. However, a non-integer real-valued ε is not suitable for ordinal and nominal variables as it ignores the discrete nature of nominal variables and the ordering nature of ordinal variables. Therefore, we restrict $\boldsymbol{\delta}$ via the following two conditions: i) $\|\boldsymbol{\delta}\|_\infty = 1$; and ii) $\|\boldsymbol{\delta}\|_1 \leq \varepsilon, \varepsilon \in \mathbb{N}$. Since the loss function is linear in $\boldsymbol{\delta}$ as shown in Eq. 1 of Appendix A, the first condition guarantees that the perturbed instance remains as an integer or a binary value. More crucially, the magnitude of one aligns with the source of feature ambiguity, which arises from non-rigorously defined set boundaries. In the example of film survey, the perturbation from 'good' to 'fair' matches the real-world decision-making process whereas replacing 'good' by 'bad' dramatically changes the original information. The second condition controls the level of perturbation. Integrating these two conditions, the perturbation $\boldsymbol{\delta}$ can change at most ε features of each instance.

To train the Mahalanobis distance, we form triplet constraints from both original and perturbed instances [14], and apply different loss functions to these triplets. For the original triplets, we adopt the loss function of LMNN and change the unit distance margin to an adjustable quantity τ. As we shall discuss in Sec. 3.4, τ determines how the instance space is divided into the support region and the adversarially vulnerable region. If the distance margin is satisfied by the triplet $(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_l)$, we will proceed to add perturbation to the instance $\boldsymbol{x}_i$. For the perturbed triplets, we adopt the perceptron loss [18]. Although the perceptron loss is rarely used in metric learning due to the lack of distance margin, it is sensible in our setting since the perturbation itself can serve as a margin in the instance space.

Integrating the above design of perturbation set and loss functions, we propose the following objective function for metric learning through adversarial training (MLadv):

$$\begin{aligned}\min_{\boldsymbol{M}\in\mathbb{S}_+^d} \lambda\|\boldsymbol{M}\|_1 + \frac{\mu}{|\mathcal{R}|}\sum_{(\boldsymbol{x}_i,\boldsymbol{x}_j,\boldsymbol{x}_l)\in\mathcal{R}} \Big[\tau + d_{\boldsymbol{M}}^2(\boldsymbol{x}_i,\boldsymbol{x}_j) - d_{\boldsymbol{M}}^2(\boldsymbol{x}_i,\boldsymbol{x}_l)\Big]_+ \\ + \frac{1-\mu}{|\mathcal{R}|}\sum_{(\boldsymbol{x}_i,\boldsymbol{x}_j,\boldsymbol{x}_l)\in\mathcal{R}} \mathbb{1}[d_{\boldsymbol{M}}^2(\boldsymbol{x}_i,\boldsymbol{x}_l) > d_{\boldsymbol{M}}^2(\boldsymbol{x}_i,\boldsymbol{x}_j) + \tau] \\ \cdot \Big[\max_{\boldsymbol{\delta}_i:\|\boldsymbol{\delta}_i\|_\infty=1, \|\boldsymbol{\delta}_i\|_1\leq\varepsilon} \{d_{\boldsymbol{M}}^2(\boldsymbol{x}_i+\boldsymbol{\delta}_i,\boldsymbol{x}_j) - d_{\boldsymbol{M}}^2(\boldsymbol{x}_i+\boldsymbol{\delta}_i,\boldsymbol{x}_l)\}\Big]_+,\end{aligned} \tag{4}$$

where $|\mathcal{R}|$ denote the numbers of triplets in the set $\mathcal{R}$; $\mathbb{1}[\cdot]$ is the indicator function which equals 1 if the condition is satisfied and 0 otherwise. The elementwise 1-norm (hereinafter abbreviated to L_1-norm), i.e. $\|\boldsymbol{M}\|_1 = \|\mathrm{vec}(\boldsymbol{M})\|_1 = \sum_{m,n=1}^P \boldsymbol{M}_{mn}$, is used to regularize the complexity of the distance matrix. As proved in Sec. 3.5, this choice of regularizer is essential to guarantee that the number of samples required for the adversarially trained metric to generalize has the same order as that for the standard metric. $\lambda > 0$ is a trade-off parameter between the regularization term and the loss function, and $\mu \in [0, 1]$ balances between the

influence from original instances and perturbed instances. The triplet set $\mathcal{R}$ is constructed in the same way as LMNN, i.e. according to Eq. 2.

3.3 Optimization Algorithm

According to the Danskin's theorem [21], the gradient of the maximum of a differentiable function is given by the gradient of the function evaluated at the maximum point, i.e.

$$\nabla_{\boldsymbol{M}} \max_{\boldsymbol{\delta} \in \Delta} \ell(\boldsymbol{x} + \boldsymbol{\delta}, \boldsymbol{y}; \boldsymbol{M}) = \nabla_{\boldsymbol{M}} \ell(\boldsymbol{x} + \boldsymbol{\delta}^\star, \boldsymbol{y}; \boldsymbol{M}),$$

where $\boldsymbol{\delta}^\star = \arg\max_{\boldsymbol{\delta} \in \Delta} \ell(\boldsymbol{x} + \boldsymbol{\delta}, \boldsymbol{y}; \boldsymbol{M})$; ∇ denotes the gradient. Therefore, we solve the optimization problem (Eq. 4) by first deriving a closed-form solution to the inner maximization problem and then updating $\boldsymbol{M}$ via the proximal gradient descent algorithm.

The solution to the worst-case perturbation $\boldsymbol{\delta}_i^\star$ can be obtained as follows: let $\boldsymbol{\delta}_i^\star = \arg\max_{\boldsymbol{\delta}_i : \|\boldsymbol{\delta}_i\|_\infty = 1, \|\boldsymbol{\delta}_i\|_1 \leq \varepsilon} \{d^2_{\boldsymbol{M}}(\boldsymbol{x}_i + \boldsymbol{\delta}_i, \boldsymbol{x}_j) - d^2_{\boldsymbol{M}}(\boldsymbol{x}_i + \boldsymbol{\delta}_i, \boldsymbol{x}_l)\}$, then

$$\boldsymbol{\delta}^\star_{i,[k]} = \begin{cases} \operatorname{sign}(\boldsymbol{M}_{k\cdot}(\boldsymbol{x}_l - \boldsymbol{x}_j)) & \text{if } k \in \underset{a=1,\cdots,P}{\arg\max_\varepsilon} |\boldsymbol{M}_{a\cdot}(\boldsymbol{x}_l - \boldsymbol{x}_j)| \\ 0 & \text{otherwise} \end{cases}, \tag{5}$$

where $\boldsymbol{\delta}^\star_{i,[k]}$ denotes the kth element of the vector $\boldsymbol{\delta}_i^\star$; $\boldsymbol{M}_{k\cdot}$ denotes the kth row of $\boldsymbol{M}$; $\arg\max_\varepsilon$ denotes the set of largest ε elements of a vector; $\operatorname{sign}(\boldsymbol{v})$ applies the sign function to each element of the vector $\boldsymbol{v}$ and $|\boldsymbol{v}|$ calculates elementwise absolute values. Detailed derivation is given in Appendix A.

Since the L_1-norm regularization introduces a non-smooth function, the proximal gradient descent algorithm is adopted to optimize $\boldsymbol{M}$ in three steps. In the gradient descent step, $\boldsymbol{M}$ is updated as

$$\begin{aligned} \boldsymbol{M}^{t+\frac{1}{3}} &= \boldsymbol{M}^t - \eta^t \nabla \boldsymbol{M}|_{\boldsymbol{M}^t} \\ \nabla \boldsymbol{M} &= \frac{\mu}{|\mathcal{R}|} \sum_{\mathcal{R}} \alpha_{ijl}(\boldsymbol{X}_{ij} - \boldsymbol{X}_{il}) + \frac{1-\mu}{|\mathcal{R}|} \sum_{\mathcal{R}} (1 - \alpha_{ijl}) \alpha^\star_{ijl}(\boldsymbol{X}^\star_{ij} - \boldsymbol{X}^\star_{il}) \end{aligned} \tag{6}$$

where $\sum_{\mathcal{R}}$ is an abbreviation for $\sum_{(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_l) \in \mathcal{R}}$; $\alpha_{ijl} = \mathbb{1}[\tau + d^2_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq d^2_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_l)]$, $\alpha^\star_{ijl} = \mathbb{1}[d^2_{\boldsymbol{M}}(\boldsymbol{x}^\star_i, \boldsymbol{x}_j) \geq d^2_{\boldsymbol{M}}(\boldsymbol{x}^\star_i, \boldsymbol{x}_l)]$; $\boldsymbol{x}^\star_i = \boldsymbol{x}_i + \boldsymbol{\delta}^\star_i$; $\boldsymbol{X}_{ij} = (\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^T$, $\boldsymbol{X}^\star_{ij} = (\boldsymbol{x}^\star_i - \boldsymbol{x}_j)(\boldsymbol{x}^\star_i - \boldsymbol{x}_j)^T$, and $\boldsymbol{X}_{il}, \boldsymbol{X}^\star_{il}$ are defined similarly. The learning rate η^t decays during training according to the exponential function $\exp(-0.99(1 + 0.01t))$. Next, we compute the proximal mapping for the L_1-norm regularization, which is equivalent to applying the soft-thresholding operator to $\boldsymbol{M}^{t+\frac{1}{3}}$:

$$\boldsymbol{M}^{t+\frac{2}{3}}_{mn} = \operatorname{sign}(\boldsymbol{M}^{t+\frac{1}{3}}_{mn})[|\boldsymbol{M}^{t+\frac{1}{3}}_{mn}| - \lambda\eta^t]_+. \tag{7}$$

Finally, $\boldsymbol{M}$ is projected onto the cone of PSD matrices via eigendecomposition:

$$\begin{aligned} \boldsymbol{M}^{t+\frac{2}{3}} &= \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^T \\ \boldsymbol{M}^{t+1} &= \boldsymbol{V} \max(\boldsymbol{\Lambda}, 0) \boldsymbol{V}^T. \end{aligned} \tag{8}$$

The optimization algorithm for the proposed method is summarized in Algorithm 1 of Appendix C.1.

We now analyze the computational complexity of the proposed method. MLadv has the same computational complexity as LMNN in calculating the distance of each triplet, performing gradient descent, and projecting onto the PSD cone; their total complexity equals $O(P^3 + nP^2 + |\mathcal{R}| \cdot P)$, where P is the feature dimension after variable encoding, n is the number of training instances, and $|\mathcal{R}|$ is the number of triplet constraints. The extra cost results from the sorting operation used to find the worst-case perturbation and the soft-thresholding operation used to perform the L_1-norm regularization. The time complexity of the sorting step is $O(P^2 \log P)$ and that of the soft-thresholding step is $O(P^2)$. Overall, the time complexity of MLadv per iteration is $O(P^3 + P^2 \log P + nP^2 + |\mathcal{R}| \cdot P)$.

3.4 Geometric Interpretation

We now provide a geometric interpretation for better understanding the effect of perturbation.

To start with, we rewrite the gradient of Eq. 6 by plugging in the worst-case perturbation derived in Eq. 5:

$$\frac{\mu}{|\mathcal{R}|} \sum_{\mathcal{R}} \mathbb{1}[d^2_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_l) \leq \tau + d^2_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_j)](\boldsymbol{X}_{ij} - \boldsymbol{X}_{il}) \tag{9}$$

$$+\frac{1-\mu}{|\mathcal{R}|} \sum_{\mathcal{R}} \mathbb{1}[d^2_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_j) + \tau < d^2_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_l) \leq d^2_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_j) + 2\|\boldsymbol{M}\boldsymbol{x}_{lj}\|_{1,[\varepsilon]}](\boldsymbol{X}^\star_{ij} - \boldsymbol{X}^\star_{il}),$$

where $\|\boldsymbol{M}\boldsymbol{x}_{lj}\|_{1,[\varepsilon]} = \sum \max_\varepsilon |\boldsymbol{M}(\boldsymbol{x}_l - \boldsymbol{x}_j)|$ is the sum of ε largest absolute values in the vector $[\boldsymbol{M}_{1\cdot}(\boldsymbol{x}_l - \boldsymbol{x}_j), \cdots, \boldsymbol{M}_{P\cdot}(\boldsymbol{x}_l - \boldsymbol{x}_j)]$.

Eq. 9 shows that, while LMNN and its variants learn the metric only on triplets where the impostor lies insufficiently far away from the instance, i.e. the difference in squared distances (DD) $d^2_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_l) - d^2_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is less than or equal to the required margin τ, the proposed method not only uses these information but also selectively exploits triplets that satisfy the margin constraint. In particular, the new selection criterion considers the correlation between the distance metric and $(\boldsymbol{x}_l - \boldsymbol{x}_j)$: if the correlation is high, i.e. the value of $\|\boldsymbol{M}\boldsymbol{x}_{lj}\|_{1,[\varepsilon]}$ is large, it is more likely this triplet will incur a loss and hence contribute to the gradient.

Fig. 1 illustrates the above discussion with two figures. In both figures, we show all instances in the linearly mapped feature space induced by the Mahalanobis distance, and consider different positions of $\boldsymbol{x}_i$ with respect to fixed target neighbor $\boldsymbol{x}_j$ and impostor $\boldsymbol{x}_l$. The left figure illustrates which triplets are used in LMNN and MLadv for calculating the gradient; for simplicity, the learned $\boldsymbol{M}$ is a scaled Euclidean distance. For $\boldsymbol{x}_{i_1}$, both methods use the triplet $(\boldsymbol{x}_{i_1}, \boldsymbol{x}_j, \boldsymbol{x}_l)$ since DD is less than τ. For $\boldsymbol{x}_{i_2}$ and $\boldsymbol{x}_{i_3}$, the methods differ. $(\boldsymbol{x}_{i_2}, \boldsymbol{x}_j, \boldsymbol{x}_l)$ and $(\boldsymbol{x}_{i_3}, \boldsymbol{x}_j, \boldsymbol{x}_l)$ satisfy the margin constraint and hence are not used in LMNN. However, they are used in our MLadv as $\boldsymbol{x}_{i_2}$ and $\boldsymbol{x}_{i_3}$ may be misclassified in the presence of perturbation; the perturbation sets with $\varepsilon = 1$ and $\varepsilon = 2$ are indicated by the blue line and blue square, respectively. When $\varepsilon = 1$, $\boldsymbol{x}_{i_2}$ may be

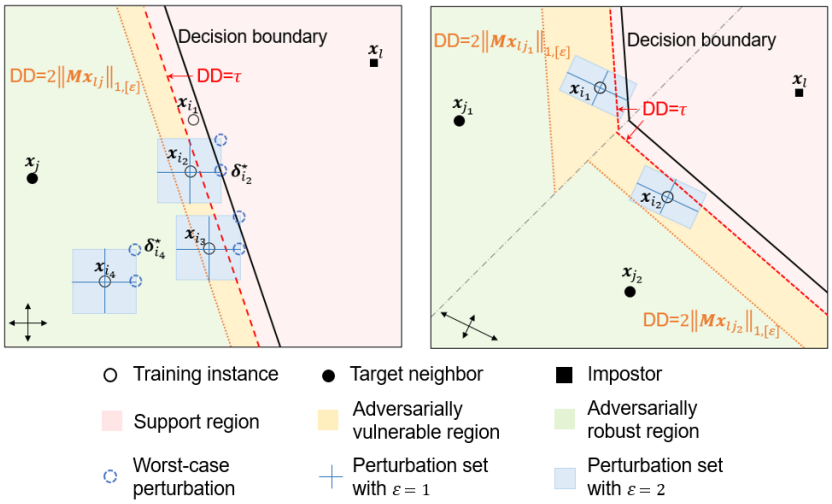


Fig. 1: Illustration of MLadv. Instances are shown in the linearly mapped feature space induced by an isotropic $\boldsymbol{M}$ (*left*) and an anisotropic $\boldsymbol{M}$ (*right*). Left: LMNN learns $\boldsymbol{M}$ based on instances from the support region where the difference in squared distances (DD) $d^2_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_l) - d^2_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is not greater than τ. MLadv learns on additional instances from the adversarially vulnerable region where the instance may be misclassified after adding the worst-case perturbation. The regions are divided by two hyperplanes that are parallel to the decision boundary with DD of τ and $2\|\boldsymbol{M}\boldsymbol{x}_{lj}\|_{1,[\varepsilon]}$, respectively. Right: Instances lying above the gray dash-dotted line select $\boldsymbol{x}_{j_1}$ as NN and should be separated farther away from the decision boundary due to the high correlation between $\boldsymbol{M}$ and $\boldsymbol{x}_l - \boldsymbol{x}_{j_1}$.

misclassified as the worst-case perturbation $\boldsymbol{\delta}^\star_{i_2}$ can drag the instance across the decision boundary; when $\varepsilon = 2$, both $\boldsymbol{x}_{i_2}$ and $\boldsymbol{x}_{i_3}$ may be misclassified. For $\boldsymbol{x}_{i_4}$, both methods ignore the triplet $(\boldsymbol{x}_{i_4}, \boldsymbol{x}_j, \boldsymbol{x}_l)$ since $\boldsymbol{x}_{i_4}$ remains far away from the decision boundary even after adding $\boldsymbol{\delta}^\star_{i_4}$.

The right figure presents the general case with an anisotropic $\boldsymbol{M}$ and multiple target neighbors, and illustrates the interaction between DD, $\boldsymbol{x}_l - \boldsymbol{x}_j$, and $\boldsymbol{M}$. Even though the DDs of $(\boldsymbol{x}_{i_1}, \boldsymbol{x}_{j_1}, \boldsymbol{x}_l)$ and $(\boldsymbol{x}_{i_2}, \boldsymbol{x}_{j_2}, \boldsymbol{x}_l)$ are the same, $\boldsymbol{x}_{i_1}$ is not robust against the worst-case perturbation whereas $\boldsymbol{x}_{i_2}$ is. The reason is that $\boldsymbol{M}$ expands the horizontal distance, as indicated by the arrows at the bottom-left corner, and has a higher correlation with $\boldsymbol{x}_l - \boldsymbol{x}_{j_1}$ compared to $\boldsymbol{x}_l - \boldsymbol{x}_{j_2}$. This suggests that, for an instance to be invariant to the worst-case perturbation, the requirement of DD is determined locally with respect to $\boldsymbol{x}_l - \boldsymbol{x}_j$ and dynamically with respect to $\boldsymbol{M}$.

In summary, as points with the same DD form a separating hyperplane that is orthogonal to the line joining $\boldsymbol{x}_j$ and $\boldsymbol{x}_l$, the proposed method essentially divides the instance space into three regions according to the hyperplanes with DD of τ and $2\|\boldsymbol{M}\boldsymbol{x}_{lj}\|_{1,[\varepsilon]}$. It then makes use of instances from the support

region and adversarially vulnerable region for learning the metric. The additional information from the latter region is particularly important for datasets with a small sample size.

3.5 Theoretical Analysis

In this section, we provide the generalization bound for metric learning trained in the adversarial setting. In essence, with the same form of loss function, adversarial training incurs a larger loss than standard training due to the addition of perturbation. Therefore, it is expected that the sample complexity would be higher in order to achieve the same generalization performance.

We start by defining some notations. The adversarial loss is defined as

$$\tilde{\ell}_{\boldsymbol{M}}(\boldsymbol{z}_i, \boldsymbol{z}_j, \boldsymbol{z}_l) = \mathbb{1}[y_i = y_j \neq y_l][\tau + \max_{\boldsymbol{\delta}_i : \|\boldsymbol{\delta}_i\|_\infty \leq \varepsilon} \{d^2_{\boldsymbol{M}}(\boldsymbol{x}_i + \boldsymbol{\delta}_i, \boldsymbol{x}_j) - d^2_{\boldsymbol{M}}(\boldsymbol{x}_i + \boldsymbol{\delta}_i, \boldsymbol{x}_l)\}]_+. \tag{10}$$

The generalization bound studies the gap between the adversarial population risk $\tilde{R}(\boldsymbol{M}) = \mathbb{E}_{(\boldsymbol{z}_i, \boldsymbol{z}_j, \boldsymbol{z}_l) \sim \mathcal{D}}[\tilde{\ell}_{\boldsymbol{M}}(\boldsymbol{z}_i, \boldsymbol{z}_j, \boldsymbol{z}_l)]$ and the adversarial empirical risk $\tilde{R}_n(\boldsymbol{M}) = \frac{1}{n(n-1)(n-2)} \sum_{i \neq j \neq l} \tilde{\ell}_{\boldsymbol{M}}(\boldsymbol{z}_i, \boldsymbol{z}_j, \boldsymbol{z}_l)$. Let $\boldsymbol{M}_{\boldsymbol{z}}$ denote the optimal solution to the learning problem:

$$\min_{\boldsymbol{M} \in \mathbb{S}^+_P} \tilde{R}_n(\boldsymbol{M}) + \lambda \|\boldsymbol{M}\|_1. \tag{11}$$

The generalization bound of $\boldsymbol{M}_{\boldsymbol{z}}$ is given by the following theorem.

Theorem 1. *Let $\boldsymbol{M}_{\boldsymbol{z}}$ be the solution to the problem (11). Then, for any $0 < \delta < 1$, with probability $1 - \delta$ we have that*

$$\begin{aligned} \tilde{R}(\boldsymbol{M}_{\boldsymbol{z}}) - \tilde{R}_n(\boldsymbol{M}_{\boldsymbol{z}}) \leq & \frac{32\tau(x^2_{max} + \varepsilon x_{max})\sqrt{e \log P}}{\lambda \sqrt{n}} \\ & + \tau\Big[1 + \frac{x^2_{max} + 2\varepsilon x_{max}}{\lambda}\Big]\sqrt{\frac{2\ln(1/\delta)}{n}} + \frac{4\tau}{\sqrt{n}}, \end{aligned} \tag{12}$$

where $x_{max} = \sup_{\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}} \|\boldsymbol{x} - \boldsymbol{x}'\|_\infty$.

Theorem 1 is established based on the Rademacher complexity [5, 35] and U-statistics [19]; proof is given in Appendix B.

We make three remarks here. First, by definition, the perturbation size is relatively small compared to $x_{\max}$, and therefore, $\varepsilon x_{\max} < x^2_{\max}$. This suggests that adversarial training does not largely increase the sample complexity. Second, as shown in the proof, if $\boldsymbol{M}$ is regularized via the Frobenius norm, the sample complexity required by adversarial training will be higher than the standard training at a rate of $O(\sqrt{P})$. To avoid the sublinear dependence of sample complexity on feature dimension, we use the L_1-norm as the regularizer. Third, Theorem 1 provides a general guarantee on the generalization performance of triplet-based metric learning trained in the adversarial setting. The adversarial loss defined in Eq. 10 with $\varepsilon = 1$ unifies the two loss functions defined in our learning objective (Eq. 4). In other words, the generalization gap of our learned metric is bounded as given in Theorem 1.

4 Experiments

In this section, we first conduct experiments on a discretized dataset to evaluate the proposed method when facing the problems of small sample size and feature ambiguity. Then, we compare it with state-of-the-art methods on datasets with all ordinal variables and mixed ordinal-and-nominal variables.

4.1 Parameter Settings

The proposed method includes four hyperparameters, namely weight of original instances μ, regularization parameter λ, distance margin τ, and perturbation size ε. Their values are identified via the random search strategy [4]. We sample 100 sets of values and select the one that gives the highest accuracy on the validation set. The range of each hyperparameter is as follows: $\mu \in [0, 1]$, $\lambda \in \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$, $\tau \in [0, \max \|\boldsymbol{x}_{lj}\|_1]$, $\varepsilon \in \{0, 1, \ldots, p^{\text{ord}} + 2p^{\text{nom}}\}$. The upper bound of τ is inspired by Eq. 9 with $\boldsymbol{M}$ initialized as the Euclidean distance. The upper bound of ε is chosen based on the fact that perturbing one ordinal level to its adjacent level or a nominal level to another level causes at most $p^{\text{ord}} + 2p^{\text{nom}}$ changes in encoded features. In addition, the initial learning rate is tuned for each dataset before optimizing the hyperparameters. We search its value from $\{10^{-2}, 10^{-1}, \ldots, 10^{2}\}$ while holding $\mu, \tau = 1$ (i.e. replicating LMNN). The MATLAB code for our method is available at `http://github.com/xyang6/MLadv`.

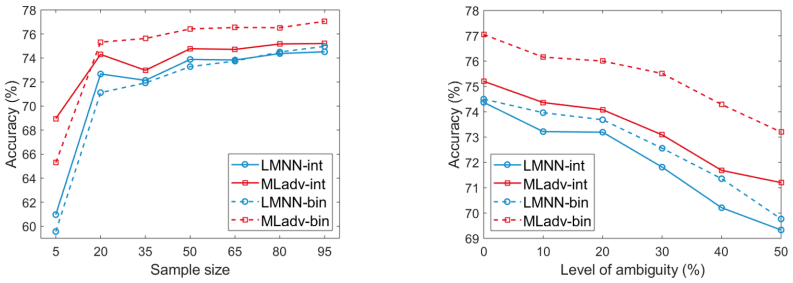
Triplet constraints are constructed from 3 target neighbors and 10 nearest impostors calculated under the Euclidean distance. 3NN is used as the classifier.

4.2 Experiments with Discretized Features

The goal of this experiment is to understand the potential of the proposed method for data with a small training set and ambiguous features. Our experiment is based on the UCI dataset Magic, which has 10 real-valued features, 19020 instances, and 2 classes. All features are first discretized into ordinal features with five equal-frequency levels, and then encoded as integers (denoted as 'int') or as a set of binary values (denoted as 'bin'). We compare LMNN and the proposed method on both types of data.

Learning from Small Training Sets In this study, we build the training set by randomly selecting $5, 20, \ldots, 95$ instances from each class; the validation and test sets each include 9000 instances. The experiment is implemented 20 times and the mean accuracy is shown in Fig. 2a; quantitative results, including the standard deviation, are provided in Appendix C.3.

First, our MLadv outperforms LMNN over the whole range of training sample size, no matter what the encoding scheme is. Second, we see a clear advantage of MLadv over LMNN when the training set is small. Third, we notice that our MLadv performs better with binary encoding than integer encoding, when the sample size is larger than 20.



(a) effect of training sample size (b) effect of feature ambiguity

Fig. 2: Evaluation of LMNN and MLadv on the discretized dataset Magic. Ordinal variables are encoded as integers ('int') or a set of binary values ('bin').

Learning under Feature Ambiguity We move on to evaluate the method when encountering feature ambiguity. The experimental setting is same as before; the training sample size is selected as 80. To simulate ambiguity, for each feature, we select $10\%, 20\%, \ldots, 50\%$ instances whose ground-truth real values are closest to the discretization threshold, and change their ordinal level to the adjacent level.

Fig. 2b shows the classification accuracy in this study. MLadv improves LMNN consistently over a wide range of ambiguity levels, and the performance gain becomes slightly larger as the ambiguity level increases.

Visualization of Training Process Our geometric interpretation suggests that MLadv considers additional triplets from the adversarially vulnerable region, which would be particularly valuable in the small-sized problem. In Fig. 3, we present the training process of MLadv at different iterations. The multidimensional scaling (MDS) is used to embed the learned distance between 20 instances into two dimensions [10]. Sizes of green circles and yellow circles are proportional to the number of triplets that do not satisfy the distance margin (i.e. second term of Eq. 4) and the number of triplets that incur a loss after adding the worst-case perturbation (i.e. third term of Eq. 4), respectively.

At the beginning of training, as instances of the same class are not well separated from instances of the different class, almost all triplets violate the distance margin constraints. Therefore, the metric is learned mostly from the original instances (as indicated by most points being in green circles). After 10 iterations, the majority of instances are closer to target neighbors than to impostors, but they are not robust to the worst-case perturbation (as indicated by a large number of yellow circles). Our method will continue using their information for metric learning. After 200 iterations, sizes of yellow circles become smaller, indicating that the learned metric becomes more robust. At the end of training, while some instances still violate the margin constraint, a large number of instances are surrounded by instances of the same class and locate far away from instances of the different class.

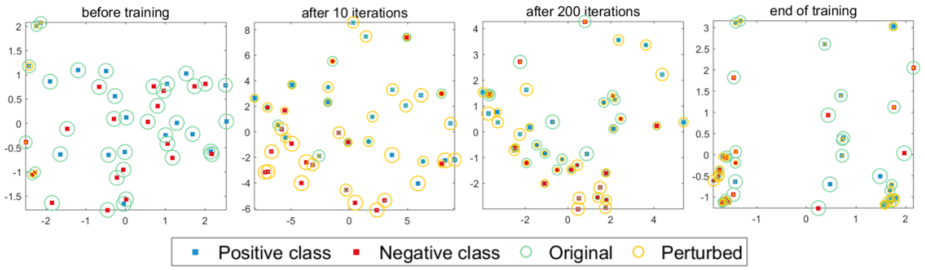


Fig. 3: Demonstration of the training process of MLadv on Magic with binary encoding. Figures show the 2D embedding of the learned distance via MDS. Sizes of green circles and yellow circles are proportional to the number of triplets violating the distance margin constraint and incurring a loss after adding the worst-case perturbation, respectively. As the training progresses, the metric becomes more robust against the perturbations and the difference between the intra-class distance and the inter-class distance becomes more remarkable.

4.3 Experiments on Real Datasets

The goal of this experiment is to compare the proposed method with robust metric learning methods and ordinal metric learning methods under the conditions when the training sample size is small or the feature ambiguity is present. As ambiguities in categorical levels occur more frequently in ordinal variables than in nominal variables, our experiments only study datasets with all ordinal variables or with a mixture of ordinal and nominal variables.

Datasets and Experimental Settings We use 6 datasets from UCI machine learning repository [11] and WEKA workbench [13]. Information on feature type, feature dimension, sample size and class information is listed in Table 1. Here, we explain the last column of ambiguity, which is assigned based on our understanding of the data. The degree of ambiguity is inherent in the data and may be inferred from the data source. Lecturer and Social Worker collect subjective ratings and assessments respectively, and hence may include a high level of ambiguity. Hayer-Roth and Lymphography are social data and medical data respectively; ambiguity is also likely to exist in these data. Car and Nursery are derived from a hierarchical decision model; their ambiguity levels are expected to be relatively low as there is an underlying rule behind these data.

Each dataset is randomly split into the training, validation, and test sets. To simulate a small-sample environment, we set their proportions as 20%,40%,40% for all datasets except for the large dataset Nursery. For Nursery, 100 samples are selected as the training set, and the remaining samples are equally split into the validation and test sets. We repeat the random split 20 times, and report the mean value and standard deviation of classification accuracy.

We compare the proposed method with LMNN and three closely related methods. DRIFT [34] and AML [6] are robust metric learning methods that

Table 1: Characteristics of the datasets

Dataset	Abb.	Feature Type	#Instances	p^{ord}	p^{nom}	#Classes	Ambiguity
Car	CA	ordinal	1728	6	0	4	low
Nursery	NU	ordinal+nominal	12960	6	2	4	low
Hayes-Roth	HR	ordinal+nominal	132	2	2	3	medium
Lymphography	LY	ordinal+nominal	148	3	15	4	medium
Lecturer	LE	ordinal	1000	4	0	5	high
Social Worker	SW	ordinal	1000	10	0	4	high

Table 2: Classification accuracy (mean value±standard deviation) of 3NN with different metric learning methods. The best methods are shown in bold and the second best ones are underlined. The mean accuracy averaged over all datasets is shown at the last row.

	LMNN-int	LMNN-bin	DRIFT	AML	Ord-LMNN	**MLadv-int**	**MLadv-bin**
	low level of ambiguity						
CA	89.94±1.31	92.24±0.89	90.24±1.17	88.84±1.13	**93.94±1.43**	90.13±1.32	92.90±1.13
NU	85.73±1.68	86.11±1.78	86.01±2.02	79.83±3.11	**87.54±1.45**	86.65±2.12	86.67±1.48
	medium level of ambiguity						
HR	71.83±10.72	76.42±6.80	71.34±10.37	65.98±7.85	75.12±9.55	74.51±10.06	**78.58±5.94**
LY	78.51±7.15	79.91±6.65	83.16±6.40	68.25±17.57	74.56±9.17	82.37±3.58	**83.33±4.85**
	high level of ambiguity						
LE	55.08±2.55	54.83±2.53	55.64±3.00	55.61±2.28	53.03±3.27	**55.90±2.70**	55.61±3.00
SW	50.00±2.61	50.58±2.30	50.73±2.93	50.50±2.07	48.68±2.82	51.10±2.15	**51.91±3.09**
Avg	71.85	73.35	72.85	68.17	72.14	73.44	**74.84**

are designed to handle feature uncertainty for real-valued data. Ord-LMNN [23] adapts LMNN to ordinal variables by assuming a latent variable for each ordinal variable, with the uniform prior tested in our experiment. Training procedures of these methods are specified in Appendix C.2.

Results and Discussions Table 2 reports the classification accuracy of 3NN with the Mahalanobis distance learned from different methods. First, we see that the proposed method outperforms the baseline method LMNN, regardless of the encoding scheme. Second, we compare MLadv with the existing ordinal metric learning method Ord-LMNN. Ord-LMNN considers the order relation of ordinal variables and is effective on datasets Balance Scale and Car. However, as the method estimates the distributional parameters for each feature, its effectiveness highly depends on the data quality. When the ambiguity level is high, the accuracy of Ord-LMNN becomes even worse than the baseline whereas our method remains competitive. Third, the robust metric learning method DRIFT achieves a high accuracy when the feature ambiguity is high. However, as the method ignores the properties of ordinal and nominal variables, its performance is inferior to our method. Overall, our method achieves the best or second-best performance on each dataset and has the highest mean accuracy.

We make a final remark on the encoding scheme and practicability of the proposed method. On most datasets, MLadv-bin is superior to MLadv-int. We

hypothesize that, as binary encoding gives a higher feature dimension, the expressive power of the metric increases and hence may improve the discriminability. While the two encoding schemes are evaluated separately in our experiment, they could be determined at the step of choosing the initial learning rate in practical applications. In other words, there is no need to tune hyperparameters twice. Except for Lymphography, this early decision can always find the optimal method between MLadv-int and MLadv-bin.

5 Conclusions and Future Work

In this paper, we propose that adversarial training with a deliberately designed perturbation set can enhance triplet-based metric learning methods in mitigating the problems of high feature ambiguity and small sample size faced by ordinal and mixed ordinal-and-nominal data. Experiments on real datasets verify the efficacy of our method. We also discuss the effect of adversarial training from both geometrical and theoretical perspectives. In the future, we intend to generalize the method to a mix of categorical and continuous features. Moreover, metric learning comprises a loss function and a regularizer, and this paper tailors the loss function to incorporate the properties of categorical features. Our future work will consider designing regularizers that are specific to categorical features.

References

1. Agresti, A.: Categorical data analysis, vol. 482. John Wiley & Sons (2003)
2. Alamuri, M., Surampudi, B.R., Negi, A.: A survey of distance/similarity measures for categorical data. In: IJCNN (2014)
3. Babbar, R., Schölkopf, B.: Data scarcity, robustness and extreme multi-label classification. Machine Learning **108**(8-9), 1329–1351 (2019)
4. Bergstra, J., Bengio, Y.: Random search for hyperparameter optimization. Journal of Machine Learning Research **13**, 281–305 (2012)
5. Cao, Q., Guo, Z.C., Ying, Y.: Generalization bounds for metric and similarity learning. Machine Learning **102**(1), 115–132 (2016)
6. Chen, G.H., Shah, D., et al.: Explaining the success of nearest neighbor methods in prediction. Foundations and Trends® in Machine Learning **10**(5-6), 337–588 (2018)
7. Chen, S., Gong, C., Yang, J., Li, X., Wei, Y., Li, J.: Adversarial metric learning. In: IJCAI (2018)
8. Cheng, V., Li, C.H., Kwok, J.T., Li, C.K.: Dissimilarity learning for nominal data. Pattern Recognition **37**(7), 1471–1477 (2004)
9. Cost, S., Salzberg, S.: A weighted nearest neighbor algorithm for learning with symbolic features. Machine Learning **10**(1), 57–78 (1993)
10. Cox, M.A., Cox, T.F.: Multidimensional scaling. In: Handbook of data visualization, pp. 315–347. Springer (2008)
11. Dua, D., Graff, C.: UCI machine learning repository (2017), `http://archive.ics.uci.edu/ml`
12. Duan, Y., Zheng, W., Lin, X., Lu, J., Zhou, J.: Deep adversarial metric learning. In: CVPR (2018)

13. Frank, E., Hall, M.A., Witten, I.H.: The WEKA workbench. online appendix for "data mining: Practical machine learning tools and techniques" (2016)
14. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: ICLR (2015)
15. Gutierrez, P.A., Perez-Ortiz, M., Sanchez-Monedero, J., Fernandez-Navarro, F., Hervas-Martinez, C.: Ordinal regression methods: survey and experimental study. IEEE Transactions on Knowledge and Data Engineering **28**(1), 127–146 (2015)
16. Hoffer, E., Ailon, N.: Deep metric learning using triplet network. In: International Workshop on Similarity-Based Pattern Recognition (2015)
17. Lanckriet, G.R., Ghaoui, L.E., Bhattacharyya, C., Jordan, M.I.: A robust minimax approach to classification. Journal of Machine Learning Research **3**, 555–582 (2002)
18. LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., Huang, F.: A tutorial on energy-based learning. Predicting structured data (2006)
19. Luo, L., Xu, J., Deng, C., Huang, H.: Robust metric learning on grassmann manifolds with generalization guarantees. In: AAAI (2019)
20. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: ICLR (2018)
21. Mangasarian, O.L.: Nonlinear programming. SIAM (1994)
22. Qian, Q., Tang, J., Li, H., Zhu, S., Jin, R.: Large-scale distance metric learning with uncertainty. In: CVPR (2018)
23. Shi, Y., Li, W., Sha, F.: Metric learning for ordinal data. In: AAAI (2016)
24. Shivaswamy, P.K., Bhattacharyya, C., Smola, A.J.: Second order cone programming approaches for handling missing and uncertain data. Journal of Machine Learning Research **7**, 1283–1314 (2006)
25. Torresani, L., Lee, K.c.: Large margin component analysis. In: NeurIPS (2007)
26. Tsang, S., Kao, B., Yip, K.Y., Ho, W.S., Lee, S.D.: Decision trees for uncertain data. IEEE Transactions on Knowledge and Data Engineering **23**(1), 64–78 (2009)
27. Wang, D., Tan, X.: Robust distance metric learning in the presence of label noise. In: AAAI (2014)
28. Wang, D., Tan, X.: Robust distance metric learning via bayesian inference. IEEE Transactions on Image Processing **27**(3), 1542–1553 (2017)
29. Wang, H., Nie, F., Huang, H.: Robust distance metric learning via simultaneous ℓ_1-norm minimization and maximization. In: ICML (2014)
30. Wang, J., Kalousis, A., Woznica, A.: Parametric local metric learning for nearest neighbor classification. In: NeurIPS (2012)
31. Weinberger, K.Q., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. Journal of Machine Learning Research **10**, 207–244 (2009)
32. Xie, J., Szymanski, B., Zaki, M.: Learning dissimilarities for categorical symbols. In: International Workshop on Feature Selection in Data Mining (2010)
33. Xing, E.P., Jordan, M.I., Russell, S.J., Ng, A.Y.: Distance metric learning with application to clustering with side-information. In: NeurIPS (2003)
34. Ye, H.J., Zhan, D.C., Si, X.M., Jiang, Y.: Learning mahalanobis distance metric: Considering instance disturbance helps. In: IJCAI (2017)
35. Yin, D., Kannan, R., Bartlett, P.: Rademacher complexity for adversarially robust generalization. In: ICML (2019)
36. Zhang, Y., Cheung, Y.m., Tan, K.C.: A unified entropy-based distance metric for ordinal-and-nominal-attribute data clustering. IEEE Transactions on Neural Networks and Learning Systems (2019)
37. Zhu, C., Cao, L., Liu, Q., Yin, J., Kumar, V.: Heterogeneous metric learning of categorical data with hierarchical couplings. IEEE Transactions on Knowledge and Data Engineering **30**(7), 1254–1267 (2018)

Metric Learning for Categorical and Ambiguous Features: An Adversarial Method (Supplementary Material)

Xiaochen Yang[1,⋆][0000−0002−9299−5951], Mingzhi Dong[1,⋆], Yiwen Guo[2][0000−0002−0709−4877], and Jing-Hao Xue(✉)[1][0000−0003−1174−610X]

[1] University College London, London, UK
xiaochen.yang.16@ucl.ac.uk,mingzhidong@gmail.com,jinghao.xue@ucl.ac.uk
[2] ByteDance AI Lab, Beijing, China
guoyiwen.ai@bytedance.com

Abstract. Appendix A derives the solution to the worst-case perturbation. Appendix B proves the generalization bound. Appendix C includes the optimization algorithm of the proposed MLadv, experimental settings of MLadv and comparative methods, and quantitative results of experiments with discretized features.

A Solution to the Worst-Case Perturbation $\boldsymbol{\delta}_i^\star$

Let $\boldsymbol{\delta}_i^\star = \arg\max_{\boldsymbol{\delta}_i:\|\boldsymbol{\delta}_i\|_\infty=1,\|\boldsymbol{\delta}_i\|_1\leq\varepsilon}\{d^2_{\boldsymbol{M}}(\boldsymbol{x}_i+\boldsymbol{\delta}_i,\boldsymbol{x}_j)-d^2_{\boldsymbol{M}}(\boldsymbol{x}_i+\boldsymbol{\delta}_i,\boldsymbol{x}_l)\}$, the closed-form solution to $\boldsymbol{\delta}_i^\star$ is derived by simplifying the objective function as follows:

$$\begin{aligned} &\arg\max_{\boldsymbol{\delta}_i}\{d^2_{\boldsymbol{M}}(\boldsymbol{x}_i+\boldsymbol{\delta}_i,\boldsymbol{x}_j)-d^2_{\boldsymbol{M}}(\boldsymbol{x}_i+\boldsymbol{\delta}_i,\boldsymbol{x}_l)\} \\ \Leftrightarrow\ &\arg\max_{\boldsymbol{\delta}_i}\{d^2_{\boldsymbol{M}}(\boldsymbol{x}_i,\boldsymbol{x}_j)-d^2_{\boldsymbol{M}}(\boldsymbol{x}_i,\boldsymbol{x}_l)+2\boldsymbol{\delta}_i^T\boldsymbol{M}(\boldsymbol{x}_l-\boldsymbol{x}_j)\} \\ \Leftrightarrow\ &\arg\max_{\boldsymbol{\delta}_i}\boldsymbol{\delta}_i^T\boldsymbol{M}(\boldsymbol{x}_l-\boldsymbol{x}_j) \end{aligned} \tag{1}$$

Under the constraint $\|\boldsymbol{\delta}_i\|_\infty = 1$, we have $\boldsymbol{\delta}_i^\star = \mathrm{sign}(\boldsymbol{M}(\boldsymbol{x}_l-\boldsymbol{x}_j))$. With the additional constraint $\|\boldsymbol{\delta}_i\|_1 \leq \varepsilon, \varepsilon\in\mathbb{N}$, we have

$$\boldsymbol{\delta}^\star_{i,[k]} = \begin{cases} \mathrm{sign}(\boldsymbol{M}_{k\cdot}(\boldsymbol{x}_l-\boldsymbol{x}_j)) & \text{if } k\in\underset{a=1,\cdots,P}{\arg\max_\varepsilon}|\boldsymbol{M}_{a\cdot}(\boldsymbol{x}_l-\boldsymbol{x}_j)| \\ 0 & \text{otherwise} \end{cases}, \tag{2}$$

where $\boldsymbol{\delta}^\star_{i,[k]}$ denotes the kth element of the vector $\boldsymbol{\delta}_i^\star$; $\boldsymbol{M}_{k\cdot}$ denotes the kth row of $\boldsymbol{M}$; $\arg\max_\varepsilon$ denotes the largest ε elements of a vector.

B Proof of Generalization Bound

The generalization bound is proved following the works of [2, 8, 5]. With the same form of loss function, adversarial training incurs a larger loss than standard

⋆ Equal contribution

training due to the addition of perturbation. Therefore, we need to show that the adversarial loss and the Rademacher complexity of the adversarial loss function class are both bounded.

For completeness, we list all notations used in the proof as follows.
Inner product, vector norm, and matrix norm: $\langle \boldsymbol{X}, \boldsymbol{Y} \rangle = \text{trace}(\boldsymbol{X}^T \boldsymbol{Y})$ denotes the Frobenius inner product of matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$. $\|\boldsymbol{v}\|_1$ and $\|\boldsymbol{v}\|_2$ denote the L_1-norm and L_2-norm of a vector $\boldsymbol{v}$, respectively; $\|\boldsymbol{M}\|_1 = \sum_{mn} \boldsymbol{M}_{mn}$ and $\|\boldsymbol{M}\|_F$ denote the (elementwise) L_1-norm and the Frobenius norm of a matrix $\boldsymbol{M}$, respectively. Given any matrix norm $\|\cdot\|$, its dual norm is defined as $\|\boldsymbol{M}\|_* = \sup\{\langle \boldsymbol{M}, \boldsymbol{X} \rangle : \|\boldsymbol{X}\| \leq 1\}$.
Adversarial loss of triplet-based metric learning problems with ℓ_∞-bounded perturbations:

$$\tilde{\ell}_{\boldsymbol{M}}(\boldsymbol{z}_i, \boldsymbol{z}_j, \boldsymbol{z}_l) = \mathbb{1}[y_i = y_j \neq y_l][\tau + \max_{\boldsymbol{\delta}_i : \|\boldsymbol{\delta}_i\|_\infty \leq \varepsilon} \{d^2_{\boldsymbol{M}}(\boldsymbol{x}_i + \boldsymbol{\delta}_i, \boldsymbol{x}_j) - d^2_{\boldsymbol{M}}(\boldsymbol{x}_i + \boldsymbol{\delta}_i, \boldsymbol{x}_l)\}]_+ \quad (3)$$

Adversarial population risk:

$$\tilde{R}(\boldsymbol{M}) = \mathbb{E}_{(\boldsymbol{z}_i, \boldsymbol{z}_j, \boldsymbol{z}_l) \sim \mathcal{D}}[\tilde{\ell}_{\boldsymbol{M}}(\boldsymbol{z}_i, \boldsymbol{z}_j, \boldsymbol{z}_l)]$$

Adversarial empirical risk:

$$\tilde{R}_n(\boldsymbol{M}) = \frac{1}{n(n-1)(n-2)} \sum_{i \neq j \neq l} \tilde{\ell}_{\boldsymbol{M}}(\boldsymbol{z}_i, \boldsymbol{z}_j, \boldsymbol{z}_l)$$

Rademacher complexity [7]: For any function class $\mathcal{F}$, given a sample set $\boldsymbol{z}^n$ of size n, the empirical Rademacher complexity of $\mathcal{F}$ with respect to $\boldsymbol{z}^n$ is defined as:

$$\hat{\mathcal{R}}_n(\mathcal{F}) = \frac{1}{n} \mathbb{E}_{\boldsymbol{\sigma}} \Big[\sup_{f \in \mathcal{F}} \sum_{i=1}^{n} \sigma_i f(\boldsymbol{z}_i) \Big],$$

where $\sigma_1, \ldots, \sigma_n$ are Rademacher variables, independently and identically distributed (i.i.d.) according to $\mathbb{P}(\sigma_i = 1) = \mathbb{P}(\sigma_i = -1) = \frac{1}{2}$. The Rademacher complexity is the expectation of the empirical Rademacher complexity over all samples of size n drawn according to $\mathcal{D}$: $\mathcal{R}_n(\mathcal{F}) = \mathbb{E}_{\boldsymbol{z}^n \sim \mathcal{D}}[\hat{\mathcal{R}}_n(\mathcal{F})]$.

We propose to learn the distance metric by optimizing the following objective function:

$$\min_{\boldsymbol{M} \in \mathbb{S}_P^+} \tilde{R}_n(\boldsymbol{M}) + \lambda \|\boldsymbol{M}\|_1. \quad (4)$$

The optimal solution to Eq. 4 is denoted as $\boldsymbol{M}_{\boldsymbol{z}}$. Since $\tilde{R}_n(\boldsymbol{M}_{\boldsymbol{z}}) + \lambda \|\boldsymbol{M}_{\boldsymbol{z}}\|_1 \leq \tilde{R}_n(\boldsymbol{0}) + \lambda \|\boldsymbol{0}\|_1 \leq \tau$, where $\boldsymbol{0}$ denotes the zero matrix, we can restrict the parameter space of $\boldsymbol{M}$ as:

$$\mathcal{H} = \{\boldsymbol{M} : \boldsymbol{M} \in \mathbb{S}_P^+, \|\boldsymbol{M}\|_1 \leq \frac{\tau}{\lambda}\}.$$

The following lemma shows that the adversarial loss is bounded.

Lemma 1. *The adversarial loss of Eq. 3 is upper bounded:*

$$\sup_{\boldsymbol{z}_i,\boldsymbol{z}_j,\boldsymbol{z}_l\in\mathcal{Z}}\sup_{\boldsymbol{M}\in\mathcal{H}}\tilde{\ell}_{\boldsymbol{M}}(\boldsymbol{z}_i,\boldsymbol{z}_j,\boldsymbol{z}_l)\leq\tau\Big[1+\frac{2\varepsilon x_{max}}{\lambda}+\frac{x_{max}^2}{\lambda}\Big], \tag{5}$$

where $x_{max}=\sup_{\boldsymbol{x},\boldsymbol{x}'\in\mathcal{X}}\|\boldsymbol{x}-\boldsymbol{x}'\|_\infty$.

Proof.

$$\begin{aligned}
&\tilde{\ell}_{\boldsymbol{M}}(\boldsymbol{z}_i,\boldsymbol{z}_j,\boldsymbol{z}_l)\\
\leq&[\tau+\max_{\boldsymbol{\delta}_i:\|\boldsymbol{\delta}_i\|_\infty\leq\varepsilon}\{d_{\boldsymbol{M}}^2(\boldsymbol{x}_i+\boldsymbol{\delta}_i,\boldsymbol{x}_j)-d_{\boldsymbol{M}}^2(\boldsymbol{x}_i+\boldsymbol{\delta}_i,\boldsymbol{x}_l)\}]_+\\
=&[\tau+d_{\boldsymbol{M}}^2(\boldsymbol{x}_i+\varepsilon\mathrm{sign}(\boldsymbol{M}(\boldsymbol{x}_l-\boldsymbol{x}_j)),\boldsymbol{x}_j)-d_{\boldsymbol{M}}^2(\boldsymbol{x}_i+\varepsilon\mathrm{sign}(\boldsymbol{M}(\boldsymbol{x}_l-\boldsymbol{x}_j)),\boldsymbol{x}_l)]_+\\
=&[\tau+d_{\boldsymbol{M}}^2(\boldsymbol{x}_i,\boldsymbol{x}_j)-d_{\boldsymbol{M}}^2(\boldsymbol{x}_i,\boldsymbol{x}_l)+2\varepsilon\,\mathrm{sign}(\boldsymbol{M}(\boldsymbol{x}_l-\boldsymbol{x}_j))^T\boldsymbol{M}(\boldsymbol{x}_l-\boldsymbol{x}_j)]_+\\
\leq&\tau+\langle\boldsymbol{M},\boldsymbol{X}_{ij}\rangle+2\varepsilon\langle\boldsymbol{M},(\boldsymbol{x}_l-\boldsymbol{x}_j)\mathrm{sign}(\boldsymbol{M}(\boldsymbol{x}_l-\boldsymbol{x}_j))^T\rangle\\
\overset{(a)}{\leq}&\tau+\|\boldsymbol{M}\|_1\|\boldsymbol{X}_{ij}\|_\infty+2\varepsilon\|\boldsymbol{M}\|_1\|(\boldsymbol{x}_l-\boldsymbol{x}_j)\mathrm{sign}(\boldsymbol{M}(\boldsymbol{x}_l-\boldsymbol{x}_j))^T\|_\infty\\
\Rightarrow\sup_{\boldsymbol{z}_i,\boldsymbol{z}_j,\boldsymbol{z}_l\in\mathcal{Z}}&\sup_{\boldsymbol{M}\in\mathcal{H}}\tilde{\ell}_{\boldsymbol{M}}(\boldsymbol{z}_i,\boldsymbol{z}_j,\boldsymbol{z}_l)\overset{(b)}{\leq}\tau+\frac{\tau}{\lambda}x_{\max}^2+2\varepsilon\frac{\tau}{\lambda}x_{\max}
\end{aligned} \tag{6}$$

Remark: Step (a) of the above proof makes use of the dual norm of $\|\boldsymbol{M}\|_1$, and step (b) bounds $\|(\boldsymbol{x}_l-\boldsymbol{x}_j)\mathrm{sign}(\boldsymbol{M}(\boldsymbol{x}_l-\boldsymbol{x}_j))^T\|_\infty$ by $x_{\max}$. If we regularize $\boldsymbol{M}$ via the Frobenius norm, the dual norm will be the Frobenius norm, and $\|(\boldsymbol{x}_l-\boldsymbol{x}_j)\mathrm{sign}(\boldsymbol{M}(\boldsymbol{x}_l-\boldsymbol{x}_j))^T\|_F\leq\sqrt{P}\sup_{\boldsymbol{x},\boldsymbol{x}'\in\mathcal{X}}\|\boldsymbol{x}_l-\boldsymbol{x}_j\|_2$. This sublinear dependence of the loss function on the feature dimension is unavoidable, even after normalizing all instances to have a unit length with respect to the L_2-norm.

The following lemma shows that the Rademacher complexity of the adversarial loss function class is bounded.

Lemma 2. *Let* $\mathcal{R}_n=\frac{1}{n}\mathbb{E}_{\boldsymbol{z}^n,\boldsymbol{\sigma}}[\sup_{M\in\mathcal{H}}\sum_{i=1}^n\sigma_i\tilde{\ell}_{\boldsymbol{M}}(\boldsymbol{z}_i,\boldsymbol{z}_i',\boldsymbol{z}_i'')]$, *where* $\boldsymbol{z}_i',\boldsymbol{z}_i''$ *are independent of* $\boldsymbol{z}_i$. *Then,*

$$\mathcal{R}_n\leq\frac{8\tau(x_{max}^2+\varepsilon x_{max})\sqrt{e\log P}}{\lambda\sqrt{n}}+\frac{\tau}{\sqrt{n}}. \tag{7}$$

Proof. The proof builds on the contraction lemma of the Rademacher complexity [4] and the Khinchin–Kahane inequality (Lemma 9 of [2]).

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{z}^n,\boldsymbol{\sigma}}\sup_{\boldsymbol{M}\in\mathcal{H}}\sum_{i=1}^n\sigma_i\tilde{\ell}_{\boldsymbol{M}}(\boldsymbol{z}_i,\boldsymbol{z}_i',\boldsymbol{z}_i'')\\
=&\mathbb{E}_{\boldsymbol{z}^n,\boldsymbol{\sigma}}\sup_{\boldsymbol{M}\in\mathcal{H}}\sum_{i=1}^n\sigma_i\mathbb{1}[y=y'\neq y''][\tau+d_{\boldsymbol{M}}^2(\boldsymbol{x}_i+\boldsymbol{\delta}_i^\star,\boldsymbol{x}_i')-d_{\boldsymbol{M}}^2(\boldsymbol{x}_i+\boldsymbol{\delta}_i^\star,\boldsymbol{x}_i'')]_+\\
\leq&\mathbb{E}_{\boldsymbol{z}^n,\boldsymbol{\sigma}}\sup_{\boldsymbol{M}\in\mathcal{H}}\Big|\sum_{i=1}^n\sigma_i\big([\tau+d_{\boldsymbol{M}}^2(\boldsymbol{x}_i+\boldsymbol{\delta}_i^\star,\boldsymbol{x}_i')-d_{\boldsymbol{M}}^2(\boldsymbol{x}_i+\boldsymbol{\delta}_i^\star,\boldsymbol{x}_i'')]_+-\tau\big)\Big|+\mathbb{E}_{\boldsymbol{\sigma}}\Big|\sum_{i=1}^n\sigma_i\tau\Big|\\
\overset{(a)}{\leq}&2\mathbb{E}_{\boldsymbol{z}^n,\boldsymbol{\sigma}}\sup_{\boldsymbol{M}\in\mathcal{H}}\Big|\sum_{i=1}^n\sigma_i\big[d_{\boldsymbol{M}}^2(\boldsymbol{x}_i+\boldsymbol{\delta}_i^\star,\boldsymbol{x}_i')-d_{\boldsymbol{M}}^2(\boldsymbol{x}_i+\boldsymbol{\delta}_i^\star,\boldsymbol{x}_i'')\big]\Big|+\mathbb{E}_{\boldsymbol{\sigma}}\Big|\sum_{i=1}^n\sigma_i\tau\Big|
\end{aligned}$$

$$
\begin{aligned}
&\leq 4\mathbb{E}_{\boldsymbol{z}^n,\boldsymbol{\sigma}} \sup_{\boldsymbol{M}\in\mathcal{H}} \Big| \sum_{i=1}^{n} \sigma_i d_{\boldsymbol{M}}^2(\boldsymbol{x}_i, \boldsymbol{x}_i') \Big| \\
&\quad + 4\varepsilon \mathbb{E}_{\boldsymbol{z}^n,\boldsymbol{\sigma}} \sup_{\boldsymbol{M}\in\mathcal{H}} \Big| \sum_{i=1}^{n} \sigma_i \operatorname{sign}(\boldsymbol{M}(\boldsymbol{x}_i'' - \boldsymbol{x}_i'))^T \boldsymbol{M}(\boldsymbol{x}_i'' - \boldsymbol{x}_i') \Big| + \mathbb{E}_{\boldsymbol{\sigma}} \Big| \sum_{i=1}^{n} \sigma_i \tau \Big| \\
&\leq \frac{4\tau}{\lambda} \mathbb{E}_{\boldsymbol{z}^n,\boldsymbol{\sigma}} \Big\| \sum_{i=1}^{n} \sigma_i (\boldsymbol{x}_i - \boldsymbol{x}_i')(\boldsymbol{x}_i - \boldsymbol{x}_i')^T \Big\|_\infty \\
&\quad + \frac{4\tau\varepsilon}{\lambda} \mathbb{E}_{\boldsymbol{z}^n,\boldsymbol{\sigma}} \Big\| \sum_{i=1}^{n} \sigma_i (\boldsymbol{x}_i'' - \boldsymbol{x}_i') \operatorname{sign}(\boldsymbol{M}(\boldsymbol{x}_i'' - \boldsymbol{x}_i'))^T \Big\|_\infty + \tau \mathbb{E}_{\boldsymbol{\sigma}} \Big| \sum_{i=1}^{n} \sigma_i \Big| \qquad (8)
\end{aligned}
$$

Step (a) is obtained by applying the Talagrand's contraction lemma.

Each term in the last line of inequality (8) can be bounded by applying the Khinchin–Kahane inequality. Here, we show the bound of the second term; bounds of the first and third terms are derived in [2] and results are listed below for completeness.

$$
\begin{aligned}
&\mathbb{E}_{\boldsymbol{z}^n,\boldsymbol{\sigma}} \Big\| \sum_{i=1}^{n} \sigma_i (\boldsymbol{x}_i'' - \boldsymbol{x}_i') \operatorname{sign}(\boldsymbol{M}(\boldsymbol{x}_i'' - \boldsymbol{x}_i'))^T \Big\|_\infty \\
&\leq \mathbb{E}_{\boldsymbol{z}^n,\boldsymbol{\sigma}} \Big\| \sum_{i=1}^{n} \sigma_i (\boldsymbol{x}_i'' - \boldsymbol{x}_i') \operatorname{sign}(\boldsymbol{M}(\boldsymbol{x}_i'' - \boldsymbol{x}_i'))^T \Big\|_q \quad \text{for any } 1 < q < \infty \\
&= \mathbb{E}_{\boldsymbol{z}^n,\boldsymbol{\sigma}} \Big[\sum_{k_1,k_2=1}^{P} \Big| \sum_{i=1}^{n} \sigma_i (\boldsymbol{x}_{i,[k_1]}'' - \boldsymbol{x}_{i,[k_1]}') \operatorname{sign}(\boldsymbol{M}_{k_2\cdot}(\boldsymbol{x}_i'' - \boldsymbol{x}_i')) \Big|^q \Big]^{\frac{1}{q}} \\
&\leq \mathbb{E}_{\boldsymbol{z}^n} \Big[\sum_{k_1,k_2=1}^{P} \mathbb{E}_{\boldsymbol{\sigma}} \Big| \sum_{i=1}^{n} \sigma_i (\boldsymbol{x}_{i,[k_1]}'' - \boldsymbol{x}_{i,[k_1]}') \operatorname{sign}(\boldsymbol{M}_{k_2\cdot}(\boldsymbol{x}_i'' - \boldsymbol{x}_i')) \Big|^q \Big]^{\frac{1}{q}} \\
&\overset{(b)}{\leq} \mathbb{E}_{\boldsymbol{z}^n} \Big[\sum_{k_1,k_2=1}^{P} (q-1)^{\frac{q}{2}} \Big(\mathbb{E}_{\boldsymbol{\sigma}} \Big| \sum_{i=1}^{n} \sigma_i (\boldsymbol{x}_{i,[k_1]}'' - \boldsymbol{x}_{i,[k_1]}') \operatorname{sign}(\boldsymbol{M}_{k_2\cdot}(\boldsymbol{x}_i'' - \boldsymbol{x}_i')) \Big|^2 \Big)^{\frac{q}{2}} \Big]^{\frac{1}{q}} \\
&= \mathbb{E}_{\boldsymbol{z}^n} \Big[\sum_{k_1,k_2=1}^{P} (q-1)^{\frac{q}{2}} \Big(\sum_{i=1}^{n} \big(\boldsymbol{x}_{i,[k_1]}'' - \boldsymbol{x}_{i,[k_1]}'\big)^2 \big(\operatorname{sign}(\boldsymbol{M}_{k_2\cdot}(\boldsymbol{x}_i'' - \boldsymbol{x}_i'))\big)^2 \Big)^{\frac{q}{2}} \Big]^{\frac{1}{q}} \\
&\leq q^{\frac{1}{2}} \mathbb{E}_{\boldsymbol{z}^n} \Big[\sum_{k_1,k_2=1}^{P} \Big(\sum_{i=1}^{n} \sup_{\boldsymbol{x},\boldsymbol{x}'\in\mathcal{X}} \|\boldsymbol{x} - \boldsymbol{x}'\|_\infty^2 \Big)^{\frac{q}{2}} \Big]^{\frac{1}{q}} \\
&= q^{\frac{1}{2}} P^{\frac{2}{q}} \sup_{\boldsymbol{x},\boldsymbol{x}'\in\mathcal{X}} \|\boldsymbol{x} - \boldsymbol{x}'\|_\infty \sqrt{n} \\
&\overset{(c)}{=} 2 \sup_{\boldsymbol{x},\boldsymbol{x}'\in\mathcal{X}} \|\boldsymbol{x} - \boldsymbol{x}'\|_\infty \sqrt{en \log P}
\end{aligned}
$$

$$
\mathbb{E}_{\boldsymbol{z}^n,\boldsymbol{\sigma}} \Big\| \sum_{i=1}^{n} \sigma_i (\boldsymbol{x}_i - \boldsymbol{x}_i')(\boldsymbol{x}_i - \boldsymbol{x}_i')^T \Big\|_\infty \leq 2 \sup_{\boldsymbol{x},\boldsymbol{x}'\in\mathcal{X}} \|\boldsymbol{x} - \boldsymbol{x}'\|_\infty^2 \sqrt{en \log P}
$$

$$
\mathbb{E}_{\boldsymbol{\sigma}} \Big| \sum_{i=1}^{n} \sigma_i \Big| \leq \sqrt{n} \qquad (9)
$$

In step (b), we apply the Khinchin–Kahane inequality with $2 < q < \infty$. In step (c), we set $q = 4 \log P$. Putting results of (9) into the inequality (8) gives the bound of (7).

We now prove the generalization bound of $\boldsymbol{M}_{\boldsymbol{z}}$.

Theorem 3. *Let $\boldsymbol{M}_{\boldsymbol{z}}$ be the solution to the problem (4). Then, for any $0 < \delta < 1$, with probability $1 - \delta$ we have that*

$$\begin{aligned}\tilde{R}(\boldsymbol{M}_{\boldsymbol{z}}) - \tilde{R}_n(\boldsymbol{M}_{\boldsymbol{z}}) \leq & \frac{32\tau(x_{max}^2 + \varepsilon x_{max})\sqrt{e \log P}}{\lambda\sqrt{n}} \\ & + \tau\Big[1 + \frac{x_{max}^2 + 2\varepsilon x_{max}}{\lambda}\Big]\sqrt{\frac{2\ln(1/\delta)}{n}} + \frac{4\tau}{\sqrt{n}},\end{aligned} \tag{10}$$

where $x_{max} = \sup_{\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}} \|\boldsymbol{x} - \boldsymbol{x}'\|_\infty$.

Proof.
Step 1: We bound the difference between $\tilde{R}(\boldsymbol{M}_{\boldsymbol{z}}) - \tilde{R}_n(\boldsymbol{M}_{\boldsymbol{z}})$ and $\mathbb{E}_{\boldsymbol{z}^n} \sup_{\boldsymbol{M} \in \mathcal{H}} [\tilde{R}(\boldsymbol{M}) - \tilde{R}_n(\boldsymbol{M})]$ via the McDiarmid's inequality [6].

First, we observe that $\tilde{R}(\boldsymbol{M}_{\boldsymbol{z}}) - \tilde{R}_n(\boldsymbol{M}_{\boldsymbol{z}}) \leq \sup_{\boldsymbol{M} \in \mathcal{H}} [\tilde{R}(\boldsymbol{M}) - \tilde{R}_n(\boldsymbol{M})]$. Next, we apply the McDiarmid's inequality to bound the difference between $\sup_{\boldsymbol{M} \in \mathcal{H}} [\tilde{R}(\boldsymbol{M}) - \tilde{R}_n(\boldsymbol{M})]$ and $\mathbb{E}_{\boldsymbol{z}^n} \sup_{\boldsymbol{M} \in \mathcal{H}} [\tilde{R}(\boldsymbol{M}) - \tilde{R}_n(\boldsymbol{M})]$, where $\mathbb{E}_{\boldsymbol{z}^n}$ denotes the expectation with respect to the training sample set $\boldsymbol{z}^n$. An essential condition of the McDiarmid's inequality is that the function $\sup_{\boldsymbol{M} \in \mathcal{H}} [\tilde{R}(\boldsymbol{M}) - \tilde{R}_n(\boldsymbol{M})]$ has bounded differences, which is shown as follows. Let $\boldsymbol{z}^n = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_{k-1}, \boldsymbol{z}_k, \boldsymbol{z}_{k+1}, \ldots, \boldsymbol{z}_n)$ and $\boldsymbol{z}^{n\prime} = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_{k-1}, \boldsymbol{z}'_k, \boldsymbol{z}_{k+1}, \ldots, \boldsymbol{z}_n)$ be two training sample sets that differ in one sample. Combining the result of Lemma 1, we have the following inequality:

$$\begin{aligned}& \Big| \sup_{\boldsymbol{M} \in \mathcal{H}} [\tilde{R}(\boldsymbol{M}; \boldsymbol{z}^n) - \tilde{R}_n(\boldsymbol{M}; \boldsymbol{z}^n)] - \sup_{\boldsymbol{M} \in \mathcal{H}} [\tilde{R}(\boldsymbol{M}; \boldsymbol{z}^{n\prime}) - \tilde{R}_n(\boldsymbol{M}; \boldsymbol{z}^{n\prime})] \Big| \\ \leq & \Big| \sup_{\boldsymbol{M} \in \mathcal{H}} \tilde{R}_n(\boldsymbol{M}; \boldsymbol{z}^n) - \sup_{\boldsymbol{M} \in \mathcal{H}} \tilde{R}_n(\boldsymbol{M}; \boldsymbol{z}^{n\prime}) \Big| \\ = & \frac{1}{n(n-1)(n-2)} \sup_{\boldsymbol{M} \in \mathcal{H}} \sum_{k \neq j \neq l} |\tilde{\ell}_{\boldsymbol{M}}(\boldsymbol{z}_k, \boldsymbol{z}_j, \boldsymbol{z}_l) - \tilde{\ell}_{\boldsymbol{M}}(\boldsymbol{z}'_k, \boldsymbol{z}_j, \boldsymbol{z}_l)| \\ \leq & \frac{1}{n(n-1)(n-2)} \sup_{\boldsymbol{M} \in \mathcal{H}} \sum_{k \neq j \neq l} (|\tilde{\ell}_{\boldsymbol{M}}(\boldsymbol{z}_k, \boldsymbol{z}_j, \boldsymbol{z}_l)| + |\tilde{\ell}_{\boldsymbol{M}}(\boldsymbol{z}'_k, \boldsymbol{z}_j, \boldsymbol{z}_l)|) \\ \leq & \frac{2}{n} \sup_{\boldsymbol{M} \in \mathcal{H}} \tilde{\ell}_{\boldsymbol{M}}(\boldsymbol{z}_k, \boldsymbol{z}_j, \boldsymbol{z}_l) \\ \leq & \frac{2\tau}{n}\Big[1 + \frac{2\varepsilon x_{\max}}{\lambda} + \frac{x_{\max}^2}{\lambda}\Big].\end{aligned} \tag{11}$$

Applying the McDiarmid's inequality to the term $\sup_{\boldsymbol{M} \in \mathcal{H}} [\tilde{R}(\boldsymbol{M}) - \tilde{R}_n(\boldsymbol{M})]$, with probability $1 - \delta$ there holds

$$\begin{aligned}& \sup_{\boldsymbol{M} \in \mathcal{H}} [\tilde{R}(\boldsymbol{M}) - \tilde{R}_n(\boldsymbol{M})] \\ \leq & \mathbb{E}_{\boldsymbol{z}^n} \sup_{\boldsymbol{M} \in \mathcal{H}} [\tilde{R}(\boldsymbol{M}) - \tilde{R}_n(\boldsymbol{M})] + \tau\Big[1 + \frac{2\varepsilon x_{\max}}{\lambda} + \frac{x_{\max}^2}{\lambda}\Big]\sqrt{\frac{2\ln(1/\delta)}{n}}.\end{aligned} \tag{12}$$

Step 2: We bound the expectation term $\mathbb{E}_{\boldsymbol{z}^n} \sup_{\boldsymbol{M}\in\mathcal{H}}[\tilde{R}(\boldsymbol{M}) - \tilde{R}_n(\boldsymbol{M})]$ by reducing the analysis of non-i.i.d. triplets to i.i.d. random variables via the U-statistic [3] and symmetrizing with the introduction of Rademacher variables [1].

First, based on Lemma 4 of [5], we can derive the following inequality:

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{z}^n} \sup_{\boldsymbol{M}\in\mathcal{H}} [\tilde{R}(\boldsymbol{M}) - \tilde{R}_n(\boldsymbol{M})] \\
=&\mathbb{E}_{\boldsymbol{z}^n} \sup_{\boldsymbol{M}\in\mathcal{H}} \Big[\tilde{R}(\boldsymbol{M}) - \frac{1}{n(n-1)(n-2)} \sum_{i\neq j\neq l} \tilde{\ell}_{\boldsymbol{M}}(\boldsymbol{z}_i, \boldsymbol{z}_j, \boldsymbol{z}_l)\Big] \\
\leq&\mathbb{E}_{\boldsymbol{z}^n} \sup_{\boldsymbol{M}\in\mathcal{H}} \Big[\tilde{R}(\boldsymbol{M}) - \frac{1}{\lfloor \frac{n}{3} \rfloor} \sum_{i=1}^{\lfloor \frac{n}{3} \rfloor} \tilde{\ell}_{\boldsymbol{M}}(\boldsymbol{z}_i, \boldsymbol{z}_{\lfloor \frac{n}{3} \rfloor+i}, \boldsymbol{z}_{\lfloor \frac{n}{3} \rfloor+i})\Big],
\end{aligned} \tag{13}$$

where $\lfloor\cdot\rfloor$ denotes the floor function. For simplicity, define $\bar{R}_n(\boldsymbol{M}) = \frac{1}{\lfloor \frac{n}{3} \rfloor} \sum_{i=1}^{\lfloor \frac{n}{3} \rfloor} \tilde{\ell}_{\boldsymbol{M}}(\boldsymbol{z}_i, \boldsymbol{z}_{\lfloor \frac{n}{3} \rfloor+i}, \boldsymbol{z}_{\lfloor \frac{n}{3} \rfloor+i})$.

Next, we symmetrize by replacing $\tilde{R}(\boldsymbol{M})$ with $\mathbb{E}_{\bar{\boldsymbol{z}}^n}[\bar{R}_n(\boldsymbol{M})]$. Let $\bar{\boldsymbol{z}}^n = (\bar{\boldsymbol{z}}_1, \ldots, \bar{\boldsymbol{z}}_n)$ denote another training set, where $\bar{\boldsymbol{z}}_i$'s are independent of each other and independent of $\boldsymbol{z}_i$'s. Then,

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{z}^n} \sup_{\boldsymbol{M}\in\mathcal{H}} [\tilde{R}(\boldsymbol{M}) - \bar{R}_n(\boldsymbol{M})] =& \mathbb{E}_{\boldsymbol{z}^n} \sup_{\boldsymbol{M}\in\mathcal{H}} \big[\mathbb{E}_{\bar{\boldsymbol{z}}^n}[\bar{R}_n(\boldsymbol{M}; \bar{\boldsymbol{z}}^n)] - \bar{R}_n(\boldsymbol{M}; \boldsymbol{z}^n)\big] \\
\leq& \mathbb{E}_{\boldsymbol{z}^n, \bar{\boldsymbol{z}}^n} \sup_{\boldsymbol{M}\in\mathcal{H}} [\bar{R}_n(\boldsymbol{M}; \bar{\boldsymbol{z}}^n) - \bar{R}_n(\boldsymbol{M}; \boldsymbol{z}^n)].
\end{aligned} \tag{14}$$

Finally, we symmetrize again by introducing the Rademacher variables.

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{z}^n, \bar{\boldsymbol{z}}^n} \sup_{\boldsymbol{M}\in\mathcal{H}} [\bar{R}_n(\boldsymbol{M}; \bar{\boldsymbol{z}}^n) - \bar{R}_n(\boldsymbol{M}; \boldsymbol{z}^n)] \\
=&\mathbb{E}_{\boldsymbol{z}^n, \bar{\boldsymbol{z}}^n} \frac{1}{\lfloor \frac{n}{3} \rfloor} \sup_{\boldsymbol{M}\in\mathcal{H}} \sum_{i=1}^{\lfloor \frac{n}{3} \rfloor} [\tilde{\ell}_{\boldsymbol{M}}(\bar{\boldsymbol{z}}_i, \bar{\boldsymbol{z}}_{\lfloor \frac{n}{3} \rfloor+i}, \bar{\boldsymbol{z}}_{\lfloor \frac{n}{3} \rfloor+i}) - \tilde{\ell}_{\boldsymbol{M}}(\boldsymbol{z}_i, \boldsymbol{z}_{\lfloor \frac{n}{3} \rfloor+i}, \boldsymbol{z}_{\lfloor \frac{n}{3} \rfloor+i})] \\
=&\mathbb{E}_{\boldsymbol{z}^n, \bar{\boldsymbol{z}}^n, \boldsymbol{\sigma}} \frac{1}{\lfloor \frac{n}{3} \rfloor} \sup_{\boldsymbol{M}\in\mathcal{H}} \sum_{i=1}^{\lfloor \frac{n}{3} \rfloor} \sigma_i [\tilde{\ell}_{\boldsymbol{M}}(\bar{\boldsymbol{z}}_i, \bar{\boldsymbol{z}}_{\lfloor \frac{n}{3} \rfloor+i}, \bar{\boldsymbol{z}}_{\lfloor \frac{n}{3} \rfloor+i}) - \tilde{\ell}_{\boldsymbol{M}}(\boldsymbol{z}_i, \boldsymbol{z}_{\lfloor \frac{n}{3} \rfloor+i}, \boldsymbol{z}_{\lfloor \frac{n}{3} \rfloor+i})] \\
\leq&\mathbb{E}_{\boldsymbol{z}^n, \bar{\boldsymbol{z}}^n, \boldsymbol{\sigma}} \frac{1}{\lfloor \frac{n}{3} \rfloor} \sup_{\boldsymbol{M}\in\mathcal{H}} \sum_{i=1}^{\lfloor \frac{n}{3} \rfloor} \sigma_i \tilde{\ell}_{\boldsymbol{M}}(\bar{\boldsymbol{z}}_i, \bar{\boldsymbol{z}}_{\lfloor \frac{n}{3} \rfloor+i}, \bar{\boldsymbol{z}}_{\lfloor \frac{n}{3} \rfloor+i}) \\
&- \mathbb{E}_{\boldsymbol{z}^n, \bar{\boldsymbol{z}}^n, \boldsymbol{\sigma}} \frac{1}{\lfloor \frac{n}{3} \rfloor} \sup_{\boldsymbol{M}\in\mathcal{H}} \sum_{i=1}^{\lfloor \frac{n}{3} \rfloor} \sigma_i \tilde{\ell}_{\boldsymbol{M}}(\boldsymbol{z}_i, \boldsymbol{z}_{\lfloor \frac{n}{3} \rfloor+i}, \boldsymbol{z}_{\lfloor \frac{n}{3} \rfloor+i}) \\
=&2\mathbb{E}_{\boldsymbol{z}^n, \boldsymbol{\sigma}} \frac{1}{\lfloor \frac{n}{3} \rfloor} \sup_{\boldsymbol{M}\in\mathcal{H}} \sum_{i=1}^{\lfloor \frac{n}{3} \rfloor} \sigma_i \tilde{\ell}_{\boldsymbol{M}}(\boldsymbol{z}_i, \boldsymbol{z}_{\lfloor \frac{n}{3} \rfloor+i}, \boldsymbol{z}_{\lfloor \frac{n}{3} \rfloor+i})
\end{aligned} \tag{15}$$

Step 3: Substituting the result of Lemma 2 into the inequality (15) and combining with inequalities (12),(13),(14) prove the theorem.

C Implementation and Additional Experimental Results of MLadv and Comparative Methods

C.1 Optimization Algorithm of MLadv

The optimization algorithm of the proposed MLadv is listed in Algorithm 1. All numbered equations refer to the equations in the main text of the paper.

Algorithm 1: Metric learning with Adversarial Training

Input: triplet set $\mathcal{R}$, parameters $\lambda, \mu, \tau, \varepsilon$, maximum number of iterations T
Output: $\boldsymbol{M}^T$
Initialization: $\boldsymbol{M}^0 = \boldsymbol{I}$;
for $t = 1$ *to* T **do**
  Compute worst-case perturbation $\boldsymbol{\delta}_i^\star$ according to Eq. 5;
  Perform gradient descent on $\boldsymbol{M}$ according to Eq. 6;
  Perform proximal mapping on $\boldsymbol{M}$ according to Eq. 7;
  Project $\boldsymbol{M}$ onto the PSD cone according to Eq. 8;

C.2 Experimental Settings of MLadv and Comparative Methods

MLadv $\boldsymbol{M}$ is initialized as the identity matrix. The learning rate η is initialized to 0.1 and decays during training according to the exponential function $\exp(-0.99(1+0.01t))$. The training stops if the relative change in the objective function is smaller than the threshold of 1e-7 or reaches the maximum number of iterations of 5000. As shown in Fig. 1, MLadv converges before reaching the maximum iteration number.

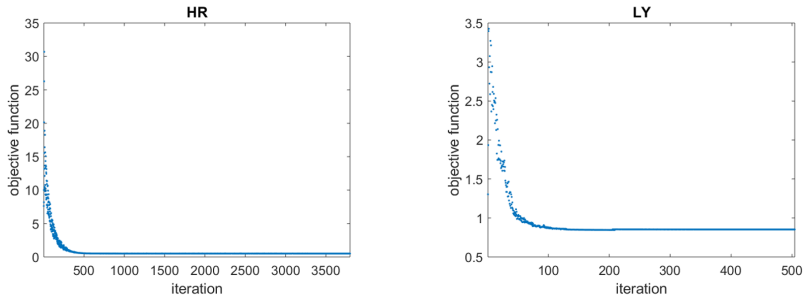


Fig. 1: Convergence curves of datasets HR and LY.

Comparative Methods The comparative methods are implemented by using the official codes provided by the authors. Trade-off parameters are selected based on the validation performance. For LMNN, we choose μ from $\{0.1, 0.2, \ldots, 0.9\}$; for DRIFT and AML, we search the grid suggested by the authors; for Ord-LMNN, we search λ from $\{0.4, 1, \ldots, 4\}$ and τ from $\{0.5, 1, \ldots, 3.5\}$. All other parameters are set as default.

C.3 Quantitative Results of Experiments with Discretized Features

Tables 1 and 2 are supplements to Fig. 2 of the main text of the paper, which list the mean value and standard deviation of classification accuracy on the discretized dataset Magic.

Table 1: Effect of training sample size on the classification accuracy (mean value±standard deviation) of LMNN and MLadv.

sample size	LMNN-int	MLadv-int	LMNN-bin	MLadv-bin
5	60.97±9.91	68.96±5.32	59.56±6.52	65.29±5.08
20	72.67±4.11	74.30±3.08	71.12±4.28	75.32±3.03
35	72.14±2.26	72.98±2.18	71.93±2.07	75.62±2.51
50	73.88±1.80	74.77±2.16	73.28±2.07	76.42±1.86
65	73.83±2.21	74.72±1.91	73.75±1.94	76.54±1.62
80	74.37±1.72	75.17±1.50	74.50±1.87	76.53±1.76
95	74.52±1.27	75.20±1.22	74.97±1.40	77.05±1.64

Table 2: Effect of ambiguity level on the classification accuracy (mean value±standard deviation) of LMNN and MLadv.

level of ambiguity	LMNN-int	MLadv-int	LMNN-bin	MLadv-bin
0%	74.37±1.72	75.20±1.22	74.50±1.87	77.05±1.64
10%	73.22±1.59	74.36±1.34	73.96±1.58	76.16±1.35
20%	73.19±1.65	74.08±1.78	73.69±2.10	76.00±1.40
30%	71.81±2.07	73.09±1.85	72.56±2.37	75.51±1.70
40%	70.21±2.63	71.69±2.34	71.36±2.63	74.29±2.54
50%	69.33±3.06	71.20±2.53	69.77±2.84	73.20±2.53

References

1. Bartlett, P.L., Mendelson, S.: Rademacher and gaussian complexities: Risk bounds and structural results. Journal of Machine Learning Research **3**, 463–482 (2002)
2. Cao, Q., Guo, Z.C., Ying, Y.: Generalization bounds for metric and similarity learning. Machine Learning **102**(1), 115–132 (2016)

3. Clémençon, S., Lugosi, G., Vayatis, N., et al.: Ranking and empirical minimization of U-statistics. The Annals of Statistics **36**(2), 844–874 (2008)
4. Ledoux, M., Talagrand, M.: Probability in Banach Spaces: isoperimetry and processes. Springer Science & Business Media (2013)
5. Luo, L., Xu, J., Deng, C., Huang, H.: Robust metric learning on grassmann manifolds with generalization guarantees. In: AAAI. vol. 33, pp. 4480–4487 (2019)
6. McDiarmid, C.: On the method of bounded differences. Surveys in combinatorics **141**(1), 148–188 (1989)
7. Mohri, M., Rostamizadeh, A., Talwalkar, A.: Foundations of machine learning. MIT press (2018)
8. Yin, D., Kannan, R., Bartlett, P.: Rademacher complexity for adversarially robust generalization. In: ICML. pp. 7085–7094 (2019)