Robust Distance Metric Learning via Bayesian Inference

Dong Wang and Xiaoyang Tan

***Abstract*— Distance metric learning (DML) has achieved great success in many computer vision tasks. However, most existing DML algorithms are based on point estimation, and thus are sensitive to the choice of training examples and tend to be over-fitting in the presence of label noise. In this paper, we present a robust DML algorithm based on Bayesian inference. In particular, our method is essentially a Bayesian extension to a previous classic DML method—large margin nearest neighbor classification and we use stochastic variational inference to estimate the posterior distribution of the transformation matrix. Furthermore, we theoretically show that the proposed algorithm is robust against label noise in the sense that an arbitrary point with label noise has bounded influence on the learnt model. With some reasonable assumptions, we derive a generalization error bound of this method in the presence of label noise. We also show that the DML hypothesis class in which our model lies is probably approximately correct-learnable and give the sample complexity. The effectiveness of the proposed method[1] is demonstrated with state of the art performance on three popular data sets with different types of label noise.**

***Index Terms*— Distance metric learning, bayesian inference, label noise, generalization error.**

I. Introduction

THE performance of many computer vision tasks, e.g., face retrieval, kinship verification and object classification strongly rely on the metric that used to measure the distances or similarities between data. For this reason, a lot of recent work has shown the interest of distance metric learning (DML) [2]–[9]. Most of these methods aim to learn a Mahalanobis distance metric which pulls together samples from the same class while pushing away those from different classes.

Although DML has achieved great success, one emerging challenge of DML is label noise [10] which may lead to serious performance deterioration [11]. Those examples with error labels could come from many sources, but one of the most common sources is due to the fact that nowadays people popularly use crowdsourcing or other techniques to harvest data from internet, especially in the field of computer vision and machine learning [12].

To address this issue, many robust DML methods have been proposed. One type of methods is adding a regularizer to the objective of DML [13]–[18]. However, although regularizer helps to avoid overfitting when the training set is small or corrupted with feature noise, it helps less in the presence of label noise. This is due to that the effect of label noise is much bigger than feature noise, and with increasing amount of training data the influence of the regularizer becomes weaker. Another type of methods is using various label noise tolerant loss functions to improve the robustness against label noise, such as: robust Fisher discriminant analysis [19], L1-norm distance metric learning [20], robust neighborhood component analysis [21]. However, those objectives are usually nonconvex or nonsmooth which would largely increase the computation cost and therefore can not be applied to big dataset. Furthermore, all these methods are based on point estimation, and thus are sensitive to the choice of training examples and tends to be over-fitting especially when training set is small and/or noisy.

Bayesian learning is also good choice for robust learning [22], [23]. Yang *et al.* [24] presents a Bayesian DML model which takes the prior distribution of the transformation matrix into account and estimate the posterior distribution via variational inference. This method can achieve state-of-the-art performance under small sample size. However, it models each sample independently by pairwise constraint, which limits its efficiency in learning.

In this work, we follow [24] to learn a robust DML model via Bayesian inference. Our method is essentially a Bayesian extension to a previous classic DML method — LMNN (large margin nearest neighbor classification [1]). In stead of using pairwise constraints as in [24], we adopt the popular large margin constraint to obtain a more robust distance metric. To apply this method on big dataset, we present an efficient training approach based on stochastic variational inference (SVI) [25]. We also conduct thorough theoretical analysis on the proposed method, showing that it is robust

Manuscript received June 29, 2017; revised November 11, 2017; accepted December 3, 2017. Date of publication December 11, 2017; date of current version January 5, 2018. This work was supported in part by the National Science Foundation of China under Grant 61672280, Grant 61373060, and Grant 61732006, in part by the National Key Research and Development Program of China under Grant 2017YFB0802300, in part by Jiangsu 333 Project under Grant BRA2017377, in part by the Pre-Research Fund of EDD, Qing Lan Project, and in part by the Funding of Jiangsu Innovation Program for Graduate Education under Grant KYLX15_0320. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Xiaochun Cao. *(Corresponding author: Xiaoyang Tan.)*

D. Wang is with the Department of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China.

X. Tan is with the Department of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China, and also with the Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing 211106, China (e-mail: x.tan@nuaa.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2017.2782366

[1] A MATLAB implementation of this method is made available at http://parnec.nuaa.edu.cn/xtan/Publication.htm

against label noise as any point with label noise has bounded influence on the learnt model. Although a few theoretical results of the regularized loss minimization framework exit in literatures (e.g., [13], [26]–[28]), the effect of label noise on the generalization error is usually missing in such analysis. We derive a generalization error bound of the proposed method and show that the DML hypothesis class of our model is PAC-learnable in the presence of label noise. This is different from previous VC-dimension based analysis on label noise [29]–[31] which mainly focus on classification with bounded loss (0-1 loss) and has not been applied on the issue of distance metric learning.

In what follows we first give the formulation of the proposed Bayesian BLMNN method and detail the SVI training process in Sec. II. We then give a thorough theoretical analysis of the proposed method in Sec. III and demonstrate its effectiveness on three real-world datasets in Sec. VI. We conclude this paper in Section. VII.

II. The Proposed Method

A. Preliminary

Assuming that we have a dataset of N data points in R^D, denoted as $\{(x_i, y_i)|\ i = 1, 2, \ldots, N\}$, where y_i is the label of the i-th data point x_i. In distance metric learning, we aim to learn a Mahalanobis matrix—$A \in R^{D\times D}$ using some form of supervision information. Mahalanobis distance metric measures the squared distance between two data points x_i and x_j as follows,

$$d_{ij}^2 = (x_i - x_j)^T A (x_i - x_j) \tag{1}$$

Note that $A \succeq 0$ is a positive semi-definite matrix and can be decomposed to $U \cdot U^T$ ($U = [u_1, u_2, \ldots, u_M] \in R^{D\times M}$, $M \leq D$). Let $y_{ij} = 1$ indicates a similar pair with $y_i = y_j$, and $y_{ij} = 0$ a dissimilar pair.

The Large Margin Nearest Neighbor (LMNN) method aims to learn a distance metric by which similar pairs should be separated from dissimilar pairs with a margin:

$$L = \sum_{ijl\in S} \max(1 + d_{ij}^2 - d_{il}^2, 0) + C_{\text{reg}}\|A\|_F^2 \tag{2}$$

where C_{reg} is a hyper-parameter, and the training set S contains $|S|$ independent triplets (ijl) satisfying $y_{ij}(1 - y_{il}) = 1$, $1 + d_{ij}^2 - d_{il}^2 > 0$[2] and $j, l \in N_i$ where N_i denotes the set of neighbors of x_i.

B. Bayesian LMNN

In Bayesian modeling, we need introduce a prior distribution $p(A)$ for the transformation matrix A, and define the likelihood function $p(S|A)$, then estimate the posterior distribution $p(A|S)$ given the training data S. For the Bayesian LMNN model, we can choose a Gaussian prior for the parameter A, and define the likelihood function according to the large margin principle [1] as,

$$\begin{aligned} p(S|A) &= \prod_{(ijl)\in S} p(x_i, x_j, x_l, y_i, y_j, y_l|A) \\ &= C \prod_{(ijl)\in S} \exp\left\{-2 \cdot \max(1 + d_{ij}^2 - d_{il}^2, 0)\right\} \end{aligned} \tag{3}$$

where C is a normalizing constant. With Bayes rule, the posterior distribution of the distance metric parameters A satisfies $p(A|S) \propto p(S|A)p(A)$, which is the product of likelihood function $p(S|A)$ and prior distribution $p(A)$. In what follows, we give the details of our schemes on how to deal with these distributions in the context of robust distance metric learning.

To work with Bayesian inference, we use a number of Gaussian distributions to approximate each single likelihood ([32], see appendix A for details):

$$\begin{aligned} &p(x_i, x_j, x_l, y_i, y_j, y_l|A) \\ &= \int_0^\infty \frac{1}{\sqrt{2\pi\lambda_{ijl}}} \exp\left\{-\frac{1}{2}\frac{(1 + d_{ij}^2 - d_{il}^2 + \lambda_{ijl})^2}{\lambda_{ijl}}\right\} d\lambda_{ijl} \end{aligned} \tag{4}$$

where λ_{ijl} is an induced parameter. And we reformulate the distance function d_{ij}^2 in a linear form, that is

$$\begin{aligned} d_{ij}^2 &= \text{tr}[A(x_i - x_j)(x_i - x_j)^T] \\ &= \gamma^T x_{ij} \end{aligned} \tag{5}$$

where the new variables γ and x_{ij} are respectively the vectorized version of matrix A and $(x_i - x_j)(x_i - x_j)^T$. To this end, our objective boils down to find the optimal γ, and in what follows, we will replace the likelihood function $p(x_i, x_j, x_l, y_i, y_j, y_l|A)$ (eq. (3)) with its equivalent form $p(x_i, x_j, x_l, y_i, y_j, y_l|\gamma)$.

C. Training via Stochastic Variational Inference

To estimate the posterior distribution of parameter γ, a proper prior distribution is also needed. In this work we use the Gaussian prior: $p(\gamma) = \mathcal{N}(\gamma|\mu_0, V_0)$. Moreover, since the parameter γ is coupled with λ_{ijl} in the likelihood (eq. (4)), we adopt a factorized variational distribution $q(\gamma, \lambda) = q(\gamma)\prod_{ijl} q(\lambda_{ijl})$ to approximate the groundtruth distribution $p(\gamma, \lambda|S, \mu_0, V_0)$. Our objective is therefore to minimize the KL distance between the variational distribution $q(\gamma, \lambda)$ and the groundtruth distribution $p(\gamma, \lambda|S, \mu_0, V_0)$,

In standard VI (please see Appendix B for details), to train the Bayesian LMNN model, we need iteratively update $\bar{\gamma}$ and $\bar{\lambda}_{ijl}$ until converged. However, Variational Inference (VI) is not a scalable algorithm, because in each iteration we must update all local variational variables (i.e. λ_{ijl}) before updating global parameter (i.e. $\bar{\gamma}$). Moreover, due to that the dimension of parameter γ is D^2, the computation cost of computing the inverse of the covariance matrix is $o(D^6)$, which is too large to train on big dataset. In [24] Yang et al. presented an eigen approximation-based method which learns a diagonal matrix instead of a full matrix A, so as to reduce the computation cost to $o(D^3)$.

[2] In LMNN, those perfect triplets that satisfying $y_{ij}(1 - y_{il}) = 1, 1 + d_{ij}^2 - d_{il}^2 \leq 0$ don't need to be trained, therefore we simply throw them away from the training set S.

To address this issue, we use stochastic variational inference (SVI) [25] to train the BLMNN model, which allows to update global parameter immediately after updating any one of the local variables. Moreover, with some mathematical manipulation, the computation cost in each step can decrease to $o(D^2)$ from $o(D^6)$.

Formally, denote $q(\gamma^t) = \mathcal{N}(\gamma^t|\bar{\gamma}^t, V_\gamma^t)$ as the distribution of γ learnt in the t-th step. In each step, we randomly choose a triplet (ijl) and compute the variational parameter λ_{ijl}, which is actually a generalized inverse Gaussian distribution ($\mathcal{GIG}$) [32],

$$q^*(\lambda_{ijl}) = \mathcal{GIG}(\lambda_{ijl}|\frac{1}{2}, 1, (1+(\bar{\gamma}^{t-1})^T x_{ijl})^2) \quad (6)$$

$$\bar{\lambda}_{ijl} = 1 + |1 + (\bar{\gamma}^{t-1})^T x_{ijl}| \quad (7)$$

where for convenience we denote:

$$x_{ijl} = x_{ij} - x_{il} \quad (8)$$

We then introduce an intermediate distribution $q(\gamma') = \mathcal{N}(\gamma'|\bar{\gamma}', V_\gamma')$ which is calculated as:

$$\begin{aligned} V_\gamma' &= (V_0^{-1} + |S| x_{ijl} \lambda_{ijl}^{-1} x_{ijl}^T)^{-1} \\ \bar{\gamma}' &= V_\gamma' [V_0^{-1}\mu_0 - |S| x_{ijl}(1 + {\lambda_{ijl}}^{-1})] \end{aligned} \quad (9)$$

where $|S|$ is the number of triplets. Let us choose $V_0 = \delta^2 I$ and follow the Matrix Inversion Lemma,[3] then we have

$$\begin{aligned} V_\gamma' &= \delta^2 I - \delta^2 b_{ijl}^{-1} x_{ijl} x_{ijl}^T \\ \bar{\gamma}' &= \mu_0 - c_{ijl} x_{ijl} \end{aligned} \quad (11)$$

where

$$\begin{aligned} b_{ijl} &= \delta^{-2}|S|^{-1}\lambda_{ijl} + \|x_{ijl}\|_2^2 \\ c_{ijl} &= \delta^2 |S|(1 + {\lambda_{ijl}}^{-1}) + b_{ijl}^{-1} x_{ijl}^T \mu_0 \\ &\quad - \delta^2 b_{ijl}^{-1} |S| \|x_{ijl}\|_2^2 (1 + {\lambda_{ijl}}^{-1}) \end{aligned} \quad (12)$$

We then update $q(\gamma^t) = \mathcal{N}(\gamma^t|\bar{\gamma}^t, V_\gamma^t)$ by,

$$\begin{aligned} \bar{\gamma}^t &= (1-\rho_t)\bar{\gamma}^{t-1} + \rho_t \bar{\gamma}' \\ V_\gamma^t &= (1-\rho_t) V_\gamma^{t-1} + \rho_t V_\gamma' \end{aligned} \quad (13)$$

where the sequence of step sizes ρ_t needs to satisfy:

$$\rho_t > 0, \quad \sum_t \rho_t = \infty, \quad \sum_t \rho_t^2 < \infty \quad (14)$$

Hence, one can see that the computational cost of learning the parameter A (or γ) is $o(|S|D^2)$ which is the same as standard LMNN. The whole pipeline of the training procedure is summarized in Alg. 1.

[3]Matrix Inversion Lemma:

$$(P + UQ^{-1}U^T)^{-1} = P^{-1} - P^{-1}U(Q + U^T P^{-1} U)^{-1} U^T P^{-1} \quad (10)$$

Algorithm 1 Bayesian LMNN

Input:

Input: Training set $\{(x_i, y_i)|\ i = 1, 2, ..., N\}$, prior distribution $\mathcal{N}(\gamma|\mu_0, V_0)$, the number of training steps T and step size $\rho_t, (t = 1, 2, 3...T)$.

Output:

posterior distribution $\mathcal{N}(\gamma|\bar{\gamma}, V_\gamma)$

—— Training Stage

1: Construct the triplet training set S.
2: Initialize $q(\gamma^0)$ with $\mathcal{N}(\gamma|\mu_0, V_0)$; compute x_{ijl} with eq. (8) for all $(ijl \in S)$; then initialize $\bar{\lambda}_{ijl}$ with eq. (7).
3: for t = 1 to T
4: randomly sample a triplet (ijl) in S
5: compute $\bar{\lambda}_{ijl}$ with eq. (7).
6: compute the intermediate $q(\gamma')$ with eq. (9)
7: update $q(\gamma^t)$ with eq. (13)
8: end
9: Return $q(\gamma^T)$.

D. Prediction

For prediction we are interested in the posterior distribution of the point-to-point distance d_{ij}^2 for a new couple of data (i, j) according to the learnt distribution of γ, which is a Gaussian distribution as shown above. Particularly, according to eq. (5) we have,

$$\begin{aligned} d_{ij}^2 &\sim \mathcal{N}(d_{ij}^2|\bar{d}_{ij}, \sigma_{ij}^2) \\ \bar{d}_{ij} &= (x_{ij})^T \bar{\gamma} \\ \sigma_{ij}^2 &= (x_{ij})^T V_\gamma x_{ij} \end{aligned} \quad (15)$$

where $\bar{\gamma}$ and V_γ are the learnt parameters via Alg. 1.

If a K-Nearest Neighbor or a kernel machine is adopted as the classifier, we just need to compute the similarity of any two data points rather than the whole distribution. In these cases, one could simply use the MAP (maximum a posterior) value—$\bar{d}_{ij}$ to estimate the d_{ij}^2.

III. Theoretic Analysis

In this section, we will give a thorough analysis of the proposed method, including the robustness against label noise, the generalization error and the sample complexity in the presence of label noise.

A. Robustness Against Label Noise

In real-world applications, there may exit various types of label noise [11] and in this work, we focus on the type with the following property,

Definition 1 (Label Noisy Triplet): A triplet (ijl) of data points are called label noisy triplet if (1) $j, l \in N_i$ (N_i denotes the set of neighbors of x_i), (2) $y_{ij}(1 - y_{il}) = 1$ and (3) in the input space $d_{ij}^2 - d_{il}^2 \geq C_d$, where $C_d > 0$ is a threshold.

Intuitively, in a local area of some feature space, the distance between similar points d_{ij}^2 in a triplet should be smaller or at least not much bigger than that of dissimilar points d_{il}^2,

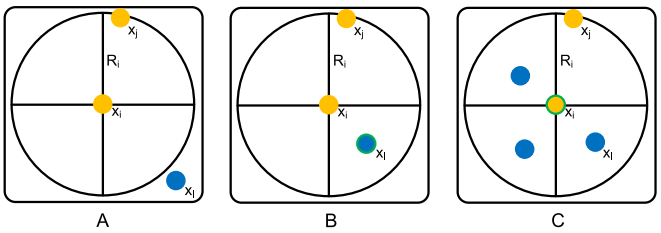

Fig. 1. Illustration of label noisy triplets. The color of a data point indicates its class label, while a data point with a green circle indicates that it is with label noise. (A) is a normal triplet with the distance between similar points d_{ij}^2 smaller than that of dissimilar points d_{il}^2; (B) is a noisy triplet in which the label of l is wrong; (C) is also a noisy triplet that the label of anchor point i is wrong and thus there are many dissimilar points around it.

otherwise the triplet would be regarded as containing points with label noise. Fig. 1 gives an illustration of this idea, where samples i and j have the same class label while the class labels of j and l are different.

Furthermore, a good DML algorithm should have the robustness property that adding an arbitrary label noisy data to the training set would not largely change the learnt model. More formally:

Definition 2 (β-Robustness Against Label Noise): Let S be a training set and z' a label noisy data.[4] Then a learning algorithm $\mathcal{A}$ is β-robust against label noise if the hypothesis it returns with training set S and $\{S, z'\}$ (denoted as $\bar{\gamma}^S$, $\bar{\gamma}^{S,z'}$ respectively) satisfy:

$$\|\bar{\gamma}^S - \bar{\gamma}^{S,z'}\|_2 \leq \beta \tag{16}$$

Note that this definition focuses on the maximum effect of one label noisy data to a learning algorithm, therefore it does not care whether or not the training set S is clean. In the following, we begin with two lemmas to show that the proposed learning algorithm (Alg. 1) owns this type of robustness.

Lemma 1: Assume the distance metric parameter $\bar{\gamma}$ satisfies $\bar{\gamma} \in R^M$ and $\|\bar{\gamma}\|_\infty \leq B$, then a label noisy triplet (ijl), as defined in Definition. 1, should satisfy $\|x_{ijl}\|_2 \geq \frac{C_d}{\sqrt{M}B}$, where x_{ijl} is given by eq. (8). (please see Appendix C for details of this proof.)

Lemma 2: When using Alg. 1 to train a BLMNN model on a big dataset ($|S| \to \infty$), the learnt model $\bar{\gamma}$ ($\bar{\gamma} \in R^M$ and $\|\bar{\gamma}\|_\infty \leq B$) is not sensitive to those label noisy triplets (ijl) with $\|x_{ijl}\|_2 \gg 0$. Furthermore, let $f(x_{ijl})$ denote the change of $\|\bar{\gamma}\|_2$ by a triplet (ijl) in one training step, that is $f(x_{ijl}) = \|\bar{\gamma}^t - \bar{\gamma}^{t-1}\|_2$, then we have

$$f(x_{ijl}) \leq 2\sqrt{M}B + \frac{3}{\|x_{ijl}\|_2} \tag{17}$$

(please see Appendix D for details of this proof.)

Fig. 2 illustrates the graph of function $f(x_{ijl})$ in a simple case of $M = 2$ and $B = 1$. The figure explicitly shows how x_{ijl} influences the magnitude of parameter $\bar{\gamma}$. We see that $f(x_{ijl})$ is "active" only in a small region around $\vec{0}$. In other words, $\bar{\gamma}'$ (in eq. (9)) is not sensitive to those (ijl) whose $\|x_{ijl}\|_2$ is large.

[4]In the LMNN based algorithms, z (or z') indicates a triplet.

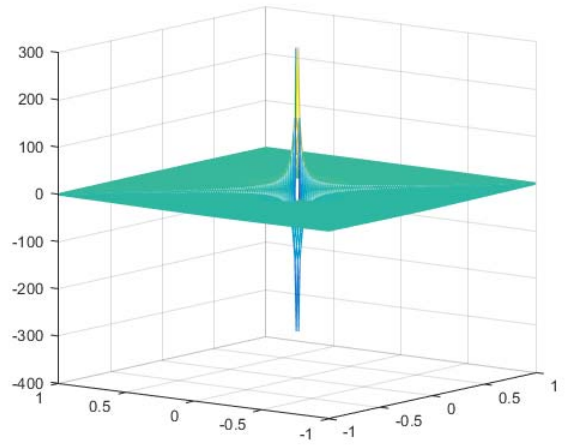

Fig. 2. Illustration the graph of function $f(x_{ijl})$ in a simple case of $\dim(\bar{\gamma}) = 2$ and $B = 1$.

We summarize this result in the following theorem:

Theorem 1: Alg. 1 is a β-robust algorithm to learn an optimal model from the hypothesis class $\mathcal{H} = \{\bar{\gamma} | \bar{\gamma} \in R^M$ and $\|\bar{\gamma}\|_\infty \leq B\}$, that adding an arbitrary label noisy triplet z' into the training set S, its effect on the learnt model— $\|\bar{\gamma}^S - \bar{\gamma}^{S,z'}\|_2$ is bounded with certain β. (please see Appendix E for details of this proof.)

Note that LMNN is not a label noise robust algorithm. From eq. (2) we can get:

$$\frac{\partial L}{\partial \gamma} = \sum_{ijl} x_{ijl} + 2C_{\text{reg}}\gamma \tag{18}$$

where the magnitude of the gradient could be largely increased by label noisy triplets with $\|x_{ijl}\|_2 \gg 0$. This also illustrates that point estimation is sensitive to label noise.

B. Generalization Error in the Presence of Label Noise

With the robustness property of our method, we derive a generalization error bound of the DML hypothesis class $\mathcal{H} = \{\bar{\gamma} | \bar{\gamma} \in R^M$ and $\|\bar{\gamma}\|_\infty \leq B\}$.

For convenience, we denote $L_{\mathcal{D}}(\bar{\gamma}^S) = E_{z\sim\mathcal{D}}[L(\bar{\gamma}^S, z)]$ as the generalization error on distribution $\mathcal{D}$, in which $\bar{\gamma}^S \in \mathcal{H}$ is learnt on a noisy training set $S = \{S_C, S_N\}$ where we denote S_C as the clean subset that only containing normal triplets while S_N as the label noisy subset that containing those label noisy triplets. We also let $\bar{\gamma}^{S_C}$, $\bar{\gamma}^{S_N}$ be the model learnt on the clean subset S_C and the label noisy subset S_N respectively.

Then the generalization error $L_{\mathcal{D}}(\bar{\gamma}^S)$ can be decomposed as:

$$L_{\mathcal{D}}(\bar{\gamma}^S) = L_{\mathcal{D}}(\bar{\gamma}^{S_C}) + (L_{\mathcal{D}}(\bar{\gamma}^S) - L_{\mathcal{D}}(\bar{\gamma}^{S_C})) \tag{19}$$

where the first term $L_{\mathcal{D}}(\bar{\gamma}^{S_C})$ is the generalization error of the model $\bar{\gamma}^{S_C}$ learnt on clean data, while the second term $L_{\mathcal{D}}(\bar{\gamma}^S) - L_{\mathcal{D}}(\bar{\gamma}^{S_C})$ indicates the effect of label noise. Note that in standard LMNN (as eq. (2)), the first term can be upper bounded [13], [33], but the second term may be very big or even unbounded. However, in our method, this error can also be bounded, and we have the following theorem:

Theorem 2: Let L be a C_L-lipschitz loss function as given in eq. (2), then on any training set S with noise level ξ ($\xi = \frac{|S_N|}{|S_C|+|S_N|}$), with probability of at least $1-\delta$, the generation error of $\bar{\gamma}^S$ in $\mathcal{H} = \{\bar{\gamma}\,|\bar{\gamma} \in R^M \text{ and } \|\bar{\gamma}\|_\infty \leq B\}$ via Alg. 1 is:

$$L_{\mathcal{D}}(\bar{\gamma}^S) \leq L_S(\bar{\gamma}^S) + \frac{2C_L C_d}{\sqrt{|S_C|}} + C_\beta \frac{\xi}{1-\xi} + C_m \sqrt{\frac{2\ln(2/\delta)}{|S_C|}} \tag{20}$$

where $L_S(\bar{\gamma}^S)$ is the empirical error of model $\bar{\gamma}^S$ on training set S, and we define $C_\beta = C_L\beta|S_C|$, $C_m = \max_{\bar{\gamma}\in H,(ijl)\sim D} L(\bar{\gamma}, x_{ijl})$. (please see Appendix F for details of this proof.)

We also derive a similar bound for LMNN by adding a constraint to the data that each triplet (ijl) should satisfy $\|x_{ijl}\|_2 \leq \frac{C_R}{\sqrt{MB}}$, which bounds the worst case of label noise. Then the generalization error of LMNN (eq. (2)) is:

$$L_{\mathcal{D}}(\bar{\gamma}^S) \leq L_S(\bar{\gamma}^S) + \frac{2C_L C_R}{\sqrt{|S|}} + C_m \sqrt{\frac{2\ln(2/\delta)}{|S|}} \tag{21}$$

Compare the two bounds in eq. (20) and eq. (21), one can see that when the noise level ξ is very small, the bound of our method can be much tighter than LMNN. This is because that $C_\beta \frac{\xi}{1-\xi} \to 0$ and the $L2$ norm of clean triplets $\|x_{ijl}\|_2$ is usually much smaller than label noisy ones, i.e., $C_d \ll C_R$.

C. Sample Complexity in the Presence of Label Noise

With the robustness of our method, we also show the PAC-learnability of hypothesis class $\mathcal{H} = \{\bar{\gamma}\,|\bar{\gamma} \in R^M \text{ and } \|\bar{\gamma}\|_\infty \leq B\}$ under label noise and we give the sample complexity. Here the definition of PAC-learnability follows from [30],

Definition 3 (PAC-Learnability in the Presence of Label Noise): A hypothesis class $\mathcal{H}$ is PAC-learnable in the presence of label noise if there exists a noise level $\xi_H \in (0,1)$, a function $n_{HN}(\epsilon,\delta,\xi_H) : (0,1)^3 \to \mathcal{N}$ and an algorithm $\mathcal{A}$, such that for any $\epsilon,\delta \in (0,1)$, and for all distribution $\mathcal{D}$ on $\mathcal{Z}$, when running the learning algorithm on $|S| \geq n_{HN}(\epsilon,\delta,\xi_H)$ i.i.d. examples generated by $\mathcal{D}$ with noise level $\xi \leq \xi_H$, the algorithm returns a hypothesis γ with the property that, with probability of at least $1-\delta$ (over the choice of the examples), $|L_{\mathcal{D}}(\bar{\gamma}^S) - L_{\mathcal{D}}(\bar{\gamma}^*)| \leq \epsilon$, where $\bar{\gamma}^*$ is the optimal hypothesis on $\mathcal{D}$.

The following theorem shows that $\mathcal{H} = \{\bar{\gamma}\,|\bar{\gamma} \in R^M \;\&\; \|\bar{\gamma}\|_\infty \leq B\}$ with Alg. 1 is PAC-learnable in the presence of label noise.

Theorem 3: A hypothesis class $\mathcal{H}$ is PAC-learnable in the presence of label noise if there exists a learning algorithm $\mathcal{A}$ satisfying: (a) $\mathcal{H}$ is PAC-learnable via $\mathcal{A}$ without label noise. (b) $\mathcal{A}$ is β-robust against label noise. Furthermore, the sample complexity is,

$$n_{HN}(\epsilon,\delta,\xi_H) = n_H(\epsilon - C_\beta \frac{\xi_H}{1-\xi_H}, \delta) \cdot \frac{1}{1-\xi_H} \tag{22}$$

where $C_\beta = C_L\beta|S_C|$. (please see Appendix G for details of this proof.)

Therefore, for any $\epsilon \in (C_\beta \frac{\xi_H}{1-\xi_H}, 1)$, $\delta \in (0,1)$, when running the algorithm $\mathcal{A}$ on $|S| \geq n_{HN}(\epsilon,\delta,\xi_H)$ i.i.d. examples generated by $\mathcal{D}$ with noise level $\xi \leq \xi_H$, we have, with probability of at least $1-\delta$, $|L_{\mathcal{D}}(\bar{\gamma}^S) - L_{\mathcal{D}}(\bar{\gamma}^*)| \leq \epsilon$.

This implies that 1) as the noise level $\xi_H \to 0$ and the training set size $|S| \to \infty$, the learnt model $\bar{\gamma}^S$ can approximate the optimal model $\bar{\gamma}^*$ (on $\mathcal{H}$) at a very small error ϵ; and 2) the noise level ξ_H that the algorithm can tolerate must be smaller than $\frac{\epsilon}{C_\beta+\epsilon}$ due to $\epsilon_N < \epsilon$.

As the sample complexity $n_H(\epsilon,\delta)$ of $\mathcal{H}$ in the clean data case is $C_S \cdot \epsilon^{-d_H}$ where C_S and d_H are positive constants that depend on $\mathcal{H}$ and the confidence $1-\delta$. Under noise level ξ, the sample complexity can increase to $n_H(\epsilon,\delta)\cdot\exp(d_H C_\beta \xi)$. (please see Appendix G for details.) Hence, one can see that with the robust learning algorithm, the DML hypothesis class is PAC-learnable in the presence of label noise, but the sample complexity can be largely raised by label noise.

D. Summary

We consider a more complicated type of label noise depending on the feature space. We show that our Bayesian method is naturally robust against this type of label noise. Note that this type of label noise is more general than that considered in [34], in which several types of label noise can all be considered as special cases. Although previous work has given some theoretical results of the regularized loss minimization problem, e.g., [13], [26]–[28], they do not consider the effect of label noise. The theoretical results in this work show that a robust DML algorithm should down weight those data with label noise, our work can be seen as a useful complement to them under such conditions.

IV. Other Variants of LMNN

To further investigate the robustness of our variational Bayes based BLMNN model, we compare it with two other variants of LMNN, one is weighted-LMNN, the other is sampling based BLMNN.

A. Weighted LMNN

Weighted-LMNN (wLMNN) is a variant of LMNN by introducing the normalization mechanism:

$$L = \sum_{ijl\in S} w_{ijl} \cdot \max(1 + d_{ij}^2 - d_{il}^2, 0) + C_{\text{reg}}\|A\|_F^2 \tag{23}$$

where the weight $w_{ijl} = \frac{1}{|d_{ij}^2 - d_{il}^2|}$ indicates the possibility of a triplet (ijl) to be label noise. This model is also optimized via gradient descend as standard LMNN. In the experiments, we would show that also with the weighting mechanism, its performance is still not as good as our method, which illustrates that the robustness of Bayesian framework is more than the data normalization.

B. Sampling Based BLMNN

Besides VI, another option to train the BLMNN model is to use MCMC/Gibbs sampling as in [35],

$$\bar{\gamma} = \int_\gamma \gamma \cdot p(\gamma\,|S) d\gamma \approx \frac{1}{C}\sum_{t=1}^{T} \gamma_t \cdot p(S|\gamma_t) \tag{24}$$

where γ_t is sampled from the prior distribution $p(\gamma) = \mathcal{N}(\gamma|\mu_0, V_0)$. The normalizing constant C can also be estimated as:

$$C = \int_{\gamma} p(S|\gamma)p(\gamma)d\gamma \approx \sum_{t=1}^{T} p(S|\gamma_t) \tag{25}$$

However, despite the simpleness, the sampling approach is not robust as the SVI based training. Note that $p(S|\gamma_t)p(\gamma_t)$ can be regarded as the probability we select γ_t as $\bar{\gamma}$. Noisy triplets would decrease the probability of finding the optimal $\bar{\gamma}$. More importantly, the effect of label noise could be very big or even unbounded.

V. Implementation Details

A. Tapering

One problem with the procedure of iterated estimating of distance metric learning using the stochastic variational inference is the lack of assured positive definiteness. This issue is usually addressed in the literature by 1) regularizing the distance metric matrix with some restriction on its energy (e.g., choosing a proper prior distribution) or 2) performing some kind of matrix shrinking or denoising such that its behavior can be stabilized. One representative technique of the latter category is by tapering some elements of a matrix to zero if they are beyond a certain range [36]. This can be implemented, for example, by doing a coordinate-wisely multiplication over it with a positive definite symmetric matrix. In this work, after getting the learnt metric A (or γ) from Algorithm. 1, we decompose A into $U\Lambda U^T$ via eigen decomposition where $U = [u_1, u_2, \ldots u_D]$ are the eigen vectors and $\Lambda = \text{diag}[r_1, r_2, \ldots r_D]$ are the eigen values. Then we taper the elements $r_i < 0$ of Λ to zero, and define the new metric $A' = \sum_{r_i>0} r_i u_i u_i^T$. Then A' is the final obtained PSD metric.

B. Parameter Setting

Parameter settings are mainly related to variational inference, including the parameters of the prior distribution $N(\gamma|\mu_0, V_0)$ and local variational approximation parameters λ_{ijl} (eq. (7)). We set μ_0 to $\epsilon\vec{1}$ where $\vec{1}$ is all 1's vector and ϵ is a small scalar (e.g. 0.01). This choice of μ_0 is equivalent to initialize BLMNN with PCA. Besides, we set V_0 to δI, where δ is also a small value (e.g. 0.01). This helps to preserve the stability of γ (eq. (9)), one important property related to overfitting. Then we initialize λ_{ijl} with eq. (7).

VI. Experiments

A. Settings

Our experiments are conducted on three real-world datasets, i.e. the MNIST dataset [43], the ImageNet dataset [44] and the MS-Celeb dataset [45]. To verify the effectiveness of the proposed method, we take the Principal Component Analysis [37] (PCA) method as a baseline since it is an unsupervised method that is completely irrelevant to the issue of label noise, and compare the proposed Bayesian LMNN (denoted as BLMNN) method with three types of methods:

1) **State of the art DML methods**: including Neighborhood Component Analysis (NCA) [38], Metric Learning for Nearest Class Mean (NCM) [39], Diversity Regularized Metric Learning (DDML) [40], Large Scale Similarity Learning (LSSL) [5], Geometric Mean Metric Learning (GMML) [41], Distance Metric Learning With Latent Variables (LADF) [8] and Distance Metric With Label Consistency (MLLC) [42]
2) **Robust DML methods**: including L1-norm distance metric learning (L1-DML) [20], pairwise constrained Bayesian DML [24] (BML), robust neighborhood component analysis (RNCA) [21]
3) **Variants of LMNN**: including LMNN [1] and its too variants: weighted-LMNN (as eq. (23), denoted as wLMNN) and the sampling based BLMNN (as eq. (24), denoted as BLMNN(S)).

To ensure fair comparison, we first use PCA to project all the data into a 100-dim subspace and force all these methods to learn full rank transformation matrix. The classifier we use is 3-NN. In addition, the performance of all the compared methods is based on the original implementation kindly provided by the corresponding authors, and the related hyper-parameters are fine-tuned through cross-validation.

B. Handwritten Digit Recognition With Random Label Noise

The MNIST dataset [43] is a popular benchmark for the task of handwritten digit recognition. We randomly sample 3000 examples (300 images per class) as training set and use the standard MNIST test set (10K images) as test set. We use an unsupervised feature extractor — CSVDDNet [46] to extract feature representation for each image, and we inject 5% to 30% random label noise by randomly flipping the labels of a given portion of data points, while keeping the test sets clean. Each experiment is repeated for ten times, and both the mean and the standard deviation of the classification accuracy are reported. To evaluate the performance of the compared methods, we also conducted pairwise one-tail statistical test under significance level 0.05.

Tab. I shows how these algorithms perform under random label noise. When there is no label noise, almost all DML methods help to make an improvement in accuracy. However, it can be seen that the performance of all the methods declines with the increasing of noise level. This is due to that label noise can mislead DML algorithms in a way that it pulls data from different class together while keeps those from the same class away. When label noise $\geq$ 15%, there is a statistically significant difference between the second best performer and our BLMNN method at a significance level of 0.05. Particularly, we have the following observations:

1) Comparison With the State of the Art DML Algorithms: One can see that the state of the art DML methods (such as LSSL, GMML, LADF and MLLC) achieve very good performance when the data are clean. However their performance decreases significantly with increasing number of data points with label noise injected, which shows that those DML methods are not robust against label noise in general. And in these difficult cases,

TABLE I

CLASSIFICATION PERFORMANCE (%) ON MNIST DATASET WITH VARYING DEGREE OF LABEL NOISE. (THE ASTERISKS INDICATE A STATISTICALLY SIGNIFICANT DIFFERENCE BETWEEN THE SECOND BEST PERFORMER AND THE PROPOSED METHOD AT A SIGNIFICANCE LEVEL OF 0.05)

label noise (%)	0	5	10	15	20	30
PCA [37]	97.25± 0.06	95.23± 0.10	95.01± 0.14	91.11± 0.19	88.91± 0.23	80.74± 0.34
NCA (2006) [38]	98.15± 0.10	96.31± 0.16	94.35± 0.22	90.06± 0.26	87.86± 0.31	78.84± 0.49
NCM (2012) [39]	98.09± 0.10	95.44± 0.18	94.41± 0.23	90.18± 0.27	88.21± 0.32	79.27± 0.50
DMML (2016) [40]	97.96± 0.12	96.04± 0.16	95.77± 0.24	92.05± 0.29	90.49± 0.34	81.53± 0.52
LSSL (2016) [5]	98.65± 0.11	**97.71± 0.17**	96.31± 0.23	93.47± 0.28*	91.02± 0.32*	82.37± 0.50*
GMML (2016) [41]	98.08± 0.10	96.61± 0.16	95.43± 0.21	92.42± 0.26	88.80± 0.30	80.55± 0.48
LADF (2017) [8]	98.12± 0.10	96.42± 0.16	95.74± 0.22	91.94± 0.27	89.26± 0.32	80.04± 0.51
MLLC (2017) [42]	98.24± 0.11	96.55± 0.17	96.07± 0.23	92.51± 0.29	89.51± 0.35	80.09± 0.52
L1-DML (2014) [20]	98.00± 0.10	96.67± 0.15	95.12± 0.22	92.55± 0.27	89.78± 0.31	81.39± 0.49
BML (2007) [24]	97.95± 0.10	96.58± 0.16	95.08± 0.23	92.17± 0.27	89.68± 0.31	81.63± 0.49
RNCA (2014) [21]	98.15± 0.10	96.73± 0.17	95.05± 0.22	92.26± 0.27	89.77± 0.31	82.31± 0.49
LMNN (2005) [1]	**98.66± 0.11**	97.65± 0.18	94.87± 0.23	92.48± 0.28	88.32± 0.33	79.65± 0.62
wLMNN	98.54± 0.10	96.25± 0.16	95.42± 0.23	92.31± 0.27	89.17± 0.33	80.86± 0.62
BLMNN(S)	98.11± 0.11	97.25± 0.16	95.27± 0.21	92.24± 0.26	89.63± 0.32	81.29± 0.52
BLMNN (ours)	98.58± 0.10	97.44± 0.18	**96.37± 0.22**	**94.34± 0.27**	**91.85± 0.31**	**84.43± 0.50**

the proposed method significantly outperforms the compared ones.

2) Comparison With Robust DML Algorithms: Our BLMNN method outperforms all the robust DML methods compared here, such as L1-DML, Robust NCA, and BML, especially when label noise $\geq$ 20%. This reveals that the Bayesian estimation method is more robust than point estimation (L1, RNCA) under high level label noise. In addition, the observation that BLMNN outperforms BML illustrates the benefits of large margin constraints for metric learning, which allows our method to effectively exploit the local structure of data.

3) Comparison With Variants of LMNN: The performance of our BLMNN also outperforms LMNN and its two variants as the noise level increases. This shows that our Bayesian extension to the LMNN is meaningful. And the robustness is main from the Variational Bayes framework rather than the weighting/normalization mechanism. In addition, we see that the variational training version of BLMNN works significantly better than the sampling version (i.e., BLMNN(S)) consistently. As pointed out in Section 4, one possible reason is that the triplets with label noise could have big negative influence on the quality of MCMC/Gibbs sampling.

C. Natural Image Classification With Simulated Label Noise

We also evaluate the performance of our method on the ImageNet [44] dataset, which contains over 1.2 million color images of totally 1,000 categories. We sample a subset of 10,000 images from ILSVRC2012 (10 categories with 1000 images per category) as the training set and use the ILSVRC2012 validation set as test set by discarding those out of the training categories. In this dataset, we do not inject random label noise, instead we use a pretrained model — ResNet-50 [47] (deep residual network) to simulate realistic label noise. The top 1 classification accuracy of ResNet-50 on the entire ILSVRC2012 training set is 88.0%. Hence, these error predictions of ResNet-50 can be considered as more realistic label noise. We also use ResNet-50 to extract feature representation for both training and test data.

TABLE II

CLASSIFICATION PERFORMANCE (%) OF VARIOUS METHODS WITH/WITHOUT LABEL NOISE ON THE IMAGENET DATASET

Algorithms	w.o. label noise	w. label noise
PCA [37]	84.8	80.6
NCM (2012) [39]	85.8	78.6
DMML (2016) [40]	87.0	81.6
LSSL (2016) [5]	86.8	81.8
GMML (2016) [41]	87.2	81.4
LADF (2017) [8]	86.8	81.6
MLLC (2017) [42]	86.4	81.8
L1-DML (2014) [20]	85.8	81.0
BML (2007) [24]	85.4	81.2
RNCA (2014) [21]	85.8	82.0
LMNN (2005) [1]	**88.0**	81.4
wLMNN	87.6	82.2
BLMNN(S)	85.6	81.6
BLMNN (ours)	87.2	**83.4**

Tab. II gives the results. The We can see that if the groundtruth of label information is used for model training, our method performs slightly worse than the LMNN method. However, if the label is assigned by ResNet-50, we see that the performance of LMNN significantly reduces by by 6.6%, while there is only 3.8% performance reduced in our method. We emphasize that the latter case is much more interesting than the former one in practice, as it provides an effective, efficient, and automatic way to harvest a large amount of information from internet with almost no cost. In this case, as Tab. II shows, our BLMNN method achieves the best performance.

D. Large Scale Face Retrieval With Realistic Label Noise

Besides evaluating the performance of our method with random or simulated label noise, we also conduct an lager scale image retrieval experiment with realistic label noise. The dataset used is the big face dataset MS-Celeb-1M [45] which contains about 10M images for 100K celebrities collected by search engine from Internet. In this dataset, there are a number of errors in the labels and the type of label noise

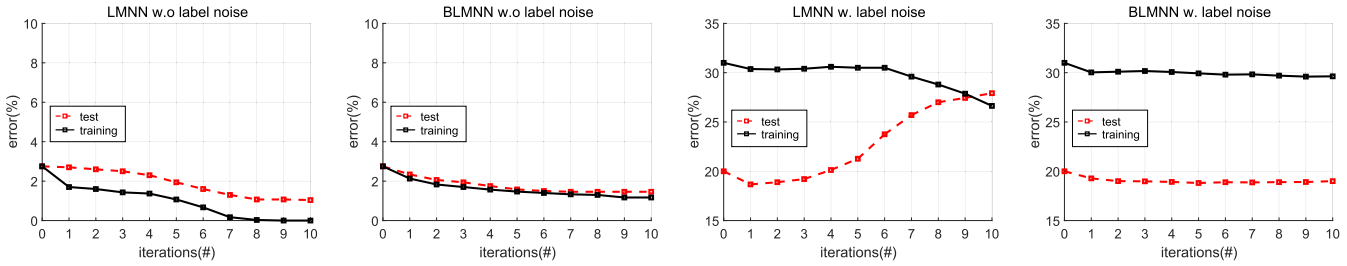

Fig. 3. Learning curves of BLMNN and LMNN on MNIST dataset. (a), (b) are without label noise, while (c), (d) are with 30% label noise. An iteration means a complete processing of all training examples.

TABLE III

FACE RETRIEVAL PERFORMANCE (MAP%) OF VARIOUS METHODS WITH/WITHOUT LABEL NOISE ON THE MS-CELEB DATASET

Algorithms	w.o. label noise	w. label noise	time (mins)
VGGFace baseline [48]	86.54	82.23	–
BML (2007) [24]	86.95	82.86	9.2
NCM (2012) [39]	87.95	82.92	10.2
L1-DML (2014) [20]	87.20	82.73	6.3
DMML (2016) [40]	87.69	82.83	8.7
LSSL (2016) [5]	86.95	81.12	–
GMML (2016) [41]	87.56	82.98	–
LADF (2017) [8]	87.28	82.23	7.0
MLLC (2017) [42]	87.04	82.25	8.3
LMNN (2005) [1]	89.25	79.13	46.0
wLMNN	87.01	82.26	20.3
BLMNN(S)	86.80	82.64	22.5
BLMNN (ours)	**89.27**	**85.06**	18.0

is unknown. In the experiment, we sample a subset of 100K images of 1200 persons, and we randomly split the training and test images with ratio 9:1. The training set contains nearly 20% label noise while the test set is clean. We use the VGGFace model [48] to extract feature representation for each image and then project them into a 100-dimensionality space with PCA. Note that this VGGFace baseline is a state-of-the-art model that achieves 97.4% accuracy on the LFW dataset [49].

Table. III gives the results on the MS-Celeb dataset. One can see that the mean average precision (mAP) of VGGFace baseline is 82.23% under the label noise and 86.64% by removing those label noisy points. When the training set is clean, the best two algorithms among them are LMNN and our BLMNN, both of which achieve 89.2% mAP. The other compared methods can also outperform the VGGFace baseline. However, under 20% label noise, most of these methods are merely slightly better than the VGGFace baseline, and LMNN even decreases to 79.13% (from 89.25%). In this case, our BLMNN still achieves 85.06% mAP which significantly outperforms the other DML algorithms. This illustrates that our method can effectively deal with realistic label noise on large scale dataset.

We have listed the training time of all methods in Table. III. Note that the test/running time are the same for all methods as they all work in subspaces with the same dimensionality. Our computational advantage is established mainly with respect to its non-Bayesian counterpart — LMNN. Actually, our method only needs less than half of the training time (18 mins) of the LMNN (49 mins), which is mainly due to that LMNN mainly focuses on those difficult triplets that are possibly label noise and need too much training efforts. Also notice that the previous Bayesian metric learning — BML [24] is faster than ours, this is because BML only learns a diagonal matrix instead of a full matrix as general DML. Although doing this can reduce the time complexity, it would possibly deteriorate the performance. The results in Table. 1 to 3 all show that our method significantly outperforms the BML method [24].

E. Discussions

To further investigate the behavior of our method, we conduct a serial of experiments on MNIST dataset. Unless specifically pointed out, the experiment settings are the same as Sec. VI-B.

1) Learning Curves: To further validate the robustness of our method, we plot in Fig. 3 the learning curves of both BLMNN and LMNN as the function of the number of iterations. We do experiments on two settings: under no label noise and under 30% label noise. Fig. 3(a) and Fig. 3(b) show when training labels are clean, the training errors of both BLMNN and LMNN will decrease with the iterations going. But when training labels contain some errors, Fig. 3(c) and Fig. 3(d) show that with the iterations going, the training errors of LMNN keep decreases while their test errors tend to rise at the same time, indicating that the point estimation based method is easy to be overfitting under the condition of label noise. Although some empirical tricks such as early stopping can be adopted, the figure clearly shows that this is not an issue for our BLMNN.

2) The Effect of Training Set Size: To evaluate the performance of our method under small sample size, we conduct another experiment by varying the number of training data from 100 to 1000. In each setting, the label noise level is fixed to 30%. Fig. 4 shows the results. When the training set is small, both BLMNN and LMNN are easily affected by label noise. As the training set size increases, the performance of both BLMNN and LMNN increases. However, when the training set size $\geq$ 600, the performance nearly does not increase. Hence, one can see that the problem of label noise can not be solved by merely adding more training data (with the same noise level). This is consistent with Theorem. 2 (eq. (20)).

3) The Effect of Feature Dimension: Due to that we force the transformation matrix A ($\in R^{D\times D}$) to be full rank, the number of parameters in all methods is D^2. Thus the dimension D can indicate the model complexity. To show

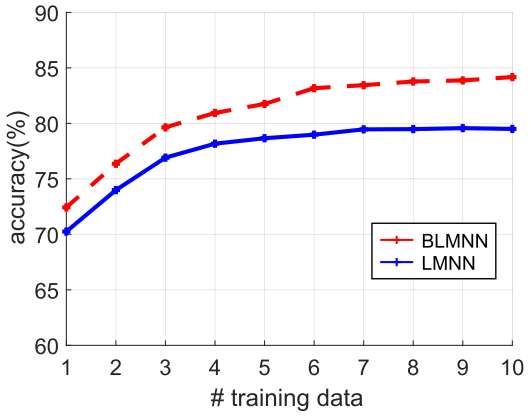

Fig. 4. Performance of BLMNN and LMNN by varying the number of training data on MNIST dataset.

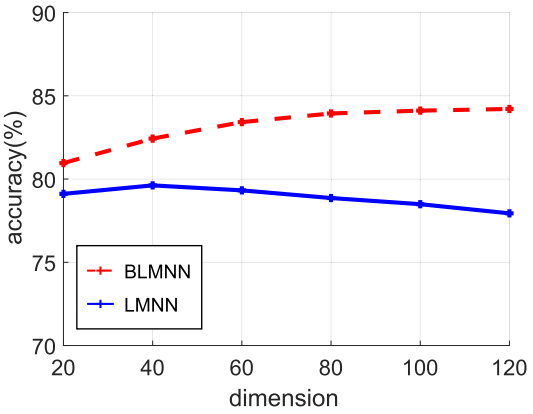

Fig. 5. Performance of BLMNN and LMNN by varying the input dimensionality (with PCA).

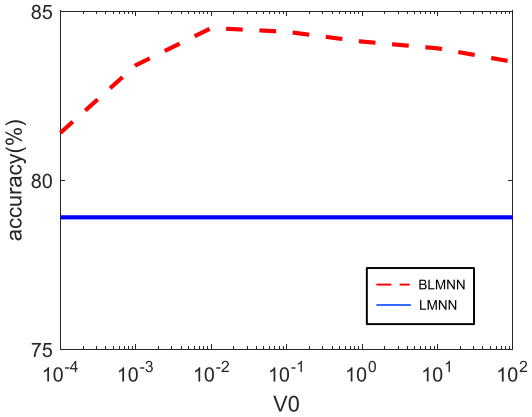

Fig. 6. The effect of prior distribution.

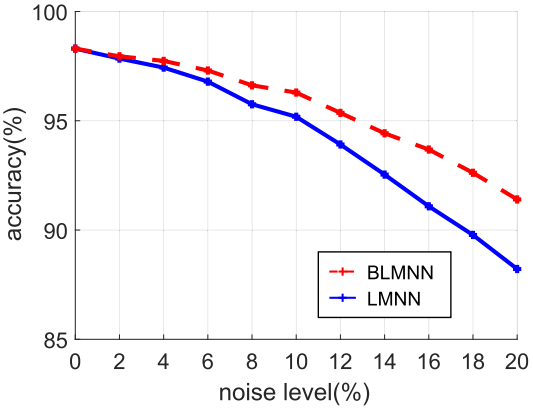

Fig. 7. Performance of BLMNN and LMNN under small noise level on MNIST dataset.

the performance of different hypothesis classes under label noise, we conduct another experiment on MNIST dataset to compare the performance of LMNN and BLMNN by varying D from 20 to 120 under 30% random label noise. Fig. 5 shows the classification results on test set. One can see that as the dimension D increases, the performance of LMNN first increases ($D \leq 40$) and then decreases ($D > 40$), which indicates that LMNN is inclined to overfitting when applied to complex hypothesis class. However, this is not an issue for our BLMNN that even when the dimension increases to 120, the performance is still improved.

4) The Effect of Prior Distribution: To investigate the effect of prior distribution $p(\gamma\,|\mu_0, V_0) = N(\gamma\,|\epsilon\vec{1}, \delta I)$, we conduct an experiment on MNIST dataset. Specifically, we vary the value of δ from 10^{-4} to 10^2 but keeping the mean value μ_0 fixed at the same time. Note that a large value of δ indicates that the prior tends to be more noninformative (i.e., higher uncertain) about the γ value. Fig. 6 shows how the performance changes as a function of the degree of uncertainty in prior. We can see that the prior is beneficial (but not the dominant). The best performer is obtained by choosing δ in the range 10^{-3} to 10^{-1}.

5) Performance Under Small Noise Level: We also conduct an experiment on MNIST dataset by varying the noise level from 2% to 20%. Fig. 7 gives the results. It shows that the proposed BLMNN keeps the performance advantage over the LMNN even from the very low noisy level.

VII. Conclusion

In this paper, we introduce a robust distance metric learning model in the presence of label noise, in which we extend a previous classic DML method — LMNN to its Bayesian version. With the stochastic variational inference based training approach, our method can be easily applied to big noisy dataset. With some assumptions, we show that our method has a tighter generalization error bound in the regularized loss minimization framework. Nowadays collecting a large amount of data directly from internet has become a popular method to address the issue of data shortage, and in this sense, our method potentially provides a valuable solution to the accompany annotation noise problem.

Appendix A

Gaussian Approximation to a Laplace Distribution $\mathcal{L}(x|0,\sigma)$*:*

$$\mathcal{L}(x|0,\sigma) = \frac{1}{2\sigma}\exp\left\{-\frac{|x|}{\sigma}\right\} = \int_0^\infty \frac{1}{\sqrt{2\pi\beta}}\exp\left\{-\frac{x^2}{2\beta}\right\}\cdot\frac{1}{2}\sigma^{-2}\exp\left\{-\frac{\beta}{2\sigma^2}\right\}d\beta \tag{26}$$

and because $2 \times \max(x, 0) = |x| + x$, hence

$$\begin{aligned}
&\exp\{-2 \cdot \max(x, 0)\} \\
&\quad = \exp\{-|x| - x\} \\
&\quad = \int_0^\infty \frac{1}{\sqrt{2\pi\beta}} \exp\left\{-\frac{x^2}{2\beta} - x\right\} \cdot \frac{1}{2} \exp\left\{-\frac{\beta}{2}\right\} d\beta \\
&\quad = \frac{1}{2} \int_0^\infty \frac{1}{\sqrt{2\pi\beta}} \exp\left\{-\frac{1}{2}\frac{(x+\beta)^2}{\beta}\right\} d\beta
\end{aligned} \tag{27}$$

APPENDIX B

Derivation of the Approximation Distributions in Variational Inference:

We use a factorized variational distribution $q(\gamma, \lambda) = q(\gamma) \prod_{ijl} q(\lambda_{ijl})$ to approximate the groundtruth posterior distribution $p(\gamma, \lambda | Y, X, \mu_0, V_0)$. That is:

$$\min \ \mathrm{KL}(q(\gamma, \lambda) || p(\gamma, \lambda | Y, X, \mu_0, V_0)) \tag{28}$$

Through standard variational inference techniques we get:

$$\begin{aligned}
\ln q^*(\gamma) &= E_{-\gamma}[\ln p(\gamma, \lambda, Y, X | \mu_0, V_0)] \\
\ln q^*(\lambda_{ijl}) &= E_{-\lambda_{ijl}}[\ln p(\gamma, \lambda, Y, X | \mu_0, V_0)]
\end{aligned} \tag{29}$$

For simplicity, we ignore the normalizing constant and focus on the following unconstrained joint distribution:

$$\begin{aligned}
&p(\gamma, \lambda, Y, X | \mu_0, V_0) \\
&\propto \prod_{ijl \in S} \frac{1}{\sqrt{2\pi\,\lambda_{ijl}}} \exp\left\{-\frac{1}{2}\frac{(1 + \gamma^T(x_{ij} - x_{il}) + \lambda_{ijl})^2}{\lambda_{ijl}}\right\} \cdot \\
&\quad \times (2\pi)^{-\frac{M}{2}} |V_0|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\gamma - \mu)^T V_0^{-1} (\gamma - \mu)\right\}
\end{aligned} \tag{30}$$

Plugging this into eq. (29), we obtain the respective approximation distributions of interest:

- For γ:

$$\begin{aligned}
\ln q(\gamma) = &-\frac{1}{2} \sum_{ijl \in S} \frac{\|\gamma^T(x_{ij} - x_{il})\|_2^2}{\lambda_{ijl}} \\
&-\frac{1}{2} \sum_{ijl \in S} \frac{2(1 + \lambda_{ijl})\gamma^T(x_{ij} - x_{il})}{\lambda_{ijl}} \\
&-\frac{1}{2}(\gamma - \mu_0)^T V_0^{-1} (\gamma - \mu_0) + \text{const}
\end{aligned} \tag{31}$$

  Then we get the distribution of γ:

$$q^*(\gamma) = \mathcal{N}(\gamma | \bar{\gamma}, V_\gamma) \tag{32}$$

$$V_\gamma = (V_0^{-1} + \sum_{ijl} x_{ijl} \lambda_{ijl}^{-1} x_{ijl}^T)^{-1} \tag{33}$$

$$\bar{\gamma} = V_\gamma [V_0^{-1} \mu_0 - \sum_{ijl} x_{ijl}(1 + \lambda_{ijl}^{-1})] \tag{34}$$

  where $x_{ijl} = x_{ij} - x_{il}$:

- For λ_{ijl}:

$$\begin{aligned}
\ln q(\lambda_{ijl}) = &-\frac{1}{2} \sum_{ijl \in S} \frac{(1 + \gamma^T(x_{ij} - x_{il}))^2 + \lambda_{ijl}^2}{\lambda_{ijl}} \\
&-\frac{1}{2} \sum_{ijl \in S} \log \lambda_{ijl} + \text{const}
\end{aligned} \tag{35}$$

  Then the distribution of λ_{ijl} is

$$q^*(\lambda_{ijl}) = \mathcal{GIG}(\lambda_{ijl} | \frac{1}{2}, 1, (1 + \bar{\gamma}^T x_{ijl})^2) \tag{36}$$

$$\bar{\lambda}_{ijl} = 1 + |1 + \bar{\gamma}^T x_{ijl}| \tag{37}$$

APPENDIX C

Proof of Lemma 1:

In the input space, we let $d_{ij}^2 = \mu_0^T x_{ij}$ (from eq. (5)), then as in Definition. 1, a label noisy triplet (ijl) satisfies $|d_{ij}^2 - d_{il}^2| = |\mu_0^T x_{ijl}| \geq C_d$. And we have,

$$\|x_{ijl}\|_2 \geq \frac{\|x_{ijl}\|_1}{\sqrt{M}} \geq \frac{|\mu_0^T x_{ijl}|}{\sqrt{M}\|\mu_0\|_\infty} \geq \frac{C_d}{\sqrt{M}B} \tag{38}$$

APPENDIX D

Proof of Lemma 2:

As the training set $|S| \to \infty$, the prior term V_0 and μ_0 in eq. (9) can be ignored. Hence eq. (9) reduces to

$$V_\gamma' = (|S| x_{ijl} \lambda_{ijl}^{-1} x_{ijl}^T)^+ \tag{39}$$

$$\bar{\gamma}' = -V_\gamma [|S| x_{ijl}(1 + \lambda_{ijl}{}^{-1})] \tag{40}$$

Set $\bar{\gamma}$ to be $\bar{\gamma}^{t-1}$, then compute λ_{ijl} with eq. (7) and plug it into eq. (40), we have

$$\begin{aligned}
\bar{\gamma}' &= -(x_{ijl} x_{ijl}^T)^+ x_{ijl}(2 + |1 + (\bar{\gamma}^{t-1})^T x_{ijl}|) \\
&= \frac{-x_{ijl}}{\|x_{ijl}\|_2^2}(2 + |1 + (\bar{\gamma}^{t-1})^T x_{ijl}|)
\end{aligned} \tag{41}$$

where the second equation of eq. (41) follows from the property of the generalized inverse of $x_{ijl} x_{ijl}^T$:

$$(x_{ijl} x_{ijl}^T)^+ = x_{ijl}(x_{ijl}^T x_{ijl})^{-1}(x_{ijl}^T x_{ijl})^{-1} x_{ijl}^T \tag{42}$$

Recall that $f(x_{ijl}) = \|\bar{\gamma}^t - \bar{\gamma}^{t-1}\|_2$, then combine eq. (13) we have

$$f(x_{ijl}) = \rho_t \|\bar{\gamma}' - \bar{\gamma}^{t-1}\|_2 \tag{43}$$

where

$$\|\bar{\gamma}'\|_2 = \frac{1}{\|x_{ijl}\|_2}(2 + |1 + (\bar{\gamma}^{t-1})^T x_{ijl}|) \tag{44}$$

$$\leq \frac{1}{\|x_{ijl}\|_2}(3 + |(\bar{\gamma}^{t-1})^T x_{ijl}|) \tag{45}$$

$$\leq \frac{1}{\|x_{ijl}\|_2}(3 + \|\bar{\gamma}^{t-1}\|_\infty \|x_{ijl}\|_1) \tag{46}$$

$$\leq \frac{1}{\|x_{ijl}\|_2}(3 + \sqrt{M}\|\bar{\gamma}^{t-1}\|_\infty \|x_{ijl}\|_2) \tag{47}$$

$$\leq \sqrt{M}B + \frac{3}{\|x_{ijl}\|_2} \tag{48}$$

and

$$\|\bar{\gamma}^{t-1}\|_2 \leq \sqrt{M}\|\bar{\gamma}^{t-1}\|_\infty \leq \sqrt{M}B \tag{49}$$

Then, we obtain the final result,

$$f(x_{ijl}) \leq \|\bar{\gamma}'\|_2 + \|\bar{\gamma}^{t-1}\|_2 \tag{50}$$

$$\leq 2\sqrt{M}B + \frac{3}{\|x_{ijl}\|_2} \tag{51}$$

where we have discarded the step size ρ_t due to $\rho_t \leq 1$.

APPENDIX E

Proof of Theorem 1:

Combine the results in Lemma. 1 and Lemma. 2, we get:

$$f(x_{ijl}) \leq \frac{\sqrt{M}B(2C_d+3)}{C_d} \tag{52}$$

Furthermore, we follow the same assumption of [50] that the influence of one single label noisy triplet z' should decrease as the size of training set S increases, then we get the result in eq. (16), where

$$\beta = \frac{C_\gamma}{|S|} \cdot \frac{\sqrt{M}B(2C_d+3)}{C_d} \tag{53}$$

where C_γ is a constant that related to the number of training steps and the hypothesis class. Note that in eq. (53) we relate β to training set size $|S|$ to facilitate the following theoretic analysis.

APPENDIX F

Proof of Theorem 2:

Due to that the generalization error $L_{\mathcal{D}}(\bar{\gamma}^S)$ can be decomposed as:

$$L_{\mathcal{D}}(\bar{\gamma}^S) = L_{\mathcal{D}}(\bar{\gamma}^{S_C}) + (L_{\mathcal{D}}(\bar{\gamma}^S) - L_{\mathcal{D}}(\bar{\gamma}^{S_C})) \tag{54}$$

We first derive the first term $L_{\mathcal{D}}(\bar{\gamma}^{S_C})$ that training the model on clean data. The objective of the our method is:

$$\max_{q(\gamma)} L_{lb} = E_{q(\gamma)} \log \frac{p(S|\gamma)p(\gamma)}{q(\gamma)} \tag{55}$$

where L_{lb} is a lower bound of $L = \log p(S|\gamma)p(\gamma)$

We assume that without label noise, the learnt posterior distribution $q(\gamma)$ (as the variational training converges) is approximately equal to the groundtruth posterior distribution $p(\gamma|S)$, that is: $\mathrm{KL}(q(\gamma)||p(\gamma|S)) \approx 0$. This is due to that the clean/normal triplets are usually easily to fit, then the MAP parameter $\bar{\gamma}$ of Alg. 1 can be regarded as a minima of eq. (2). Hence, the generalization error of LMNN and our method are approximately equal when learning on a clean data set, that is

$$L_{\mathcal{D}}(\bar{\gamma}^{S_C}) \leq L_{S_C}(\bar{\gamma}^{S_C}) + \frac{2C_LC_d}{\sqrt{|S_C|}} + C_m\sqrt{\frac{2\ln(2/\delta)}{|S_C|}} \tag{56}$$

where each normal triplet satisfies $\|x_{ijl}\|_2 \leq \frac{C_d}{\sqrt{M}B}$ (according to Definition. 1 and Lemma. 1). This result is from [33].

The second term, that the effect of label noise can be bounded as:

$$\begin{aligned}
&|L_{\mathcal{D}}(\bar{\gamma}^{S_C}) - L_{\mathcal{D}}(\bar{\gamma}^{S_C,S_N})| \\
&\quad \leq |L_{\mathcal{D}}(\bar{\gamma}^{S_C}) - L_{\mathcal{D}}(\bar{\gamma}^{S_C,z'_1})| + |L_{\mathcal{D}}(\bar{\gamma}^{S_C,z'_1}) \\
&\qquad - L_{\mathcal{D}}(\bar{\gamma}^{S_C,z'_1,z'_2})| \ldots + |L_{\mathcal{D}}(\bar{\gamma}^{S_C,z'_1,z'_2,\ldots z'_{|S_N|-1}}) \\
&\qquad - L_{\mathcal{D}}(\bar{\gamma}^{S_C,S_N})|
\end{aligned} \tag{57}$$

where $S_N = \{z'_1, z'_2, \ldots, z'_{|S_N|}\}$ and each z'_i is a label noisy triplet. Then with eq. (16) and the C_L-Lipschitz continuity of loss function L, we get,

$$\begin{aligned}
&|L_{\mathcal{D}}(\bar{\gamma}^{S_C}) - L_{\mathcal{D}}(\bar{\gamma}^{S})| \\
&\quad \leq C_\beta\left(\frac{1}{|S_C|} + \frac{1}{|S_C|+1} + \ldots + \frac{1}{|S_C|+|S_N|-1}\right) \\
&\quad < C_\beta \frac{|S_N|}{|S_C|} = C_\beta \frac{\xi}{1-\xi}
\end{aligned} \tag{58}$$

where $C_\beta = C_L\beta|S|$ in which $|S|$ varies from $|S_C|$ to $|S_C| + |S_N| - 1$ (eq. (57) to eq. (58)). Furthermore, we have

$$L_{S_C}(\bar{\gamma}^{S_C}) \leq L_S(\bar{\gamma}^S) \tag{59}$$

This is due to that

$$L_{S_C}(\bar{\gamma}^{S_C}) \leq L_{S_C}(\bar{\gamma}^S) + L_{S_N}(\bar{\gamma}^S) = L_S(\bar{\gamma}^S) \tag{60}$$

where $S = \{S_C, S_N\}$. Then combine eq. (56), eq. (58) and eq. (59), we get eq. (20). Note that if we simply assume $\|x_{ijl}\|_2 \leq \frac{C_R}{\sqrt{M}B}$ for all (ijl) (including those label noisy ones), then eq. (56) becomes:

$$L_{\mathcal{D}}(\bar{\gamma}^S) \leq L_S(\bar{\gamma}^S) + \frac{2C_LC_R}{\sqrt{|S|}} + C_m\sqrt{\frac{2\ln(2/\delta)}{|S|}} \tag{61}$$

which can be regarded as the generalization error bound of LMNN under label noise.

APPENDIX G

Proof of Theorem 3:

As presented in Appendix F, without label noise, finding the MAP parameter $\bar{\gamma}$ via Alg. 1 is almost equivalent to minimizing the regularized loss in eq. (2). Hence, $\mathcal{H}$ is PAC-learnable via Alg. 1 [33] without label noise, and there exists a function $n_H(\epsilon - \epsilon_N, \delta)$ that on $|S_C| \geq n_H(\epsilon - \epsilon_N, \delta)$, $(0 < \epsilon_N < \epsilon)$ clean examples, we have, with probability of at least $1-\delta$, $|L_{\mathcal{D}}(\bar{\gamma}^* - L_{\mathcal{D}}(\bar{\gamma}^{S_C}))| \leq \epsilon - \epsilon_N$. To show the learnability in the presence of label noise, we need to upper bound $|L_{\mathcal{D}}(\bar{\gamma}^*) - L_{\mathcal{D}}(\bar{\gamma}^S)|$ which can be decomposed into,

$$\begin{aligned}
&|L_{\mathcal{D}}(\bar{\gamma}^*) - L_{\mathcal{D}}(\bar{\gamma}^S)| \\
&\quad \leq |L_{\mathcal{D}}(\bar{\gamma}^*) - L_{\mathcal{D}}(\bar{\gamma}^{S_C})| + |L_{\mathcal{D}}(\bar{\gamma}^{S_C}) - L_{\mathcal{D}}(\bar{\gamma}^S)|
\end{aligned} \tag{62}$$

Given eq. (58), we let $\epsilon_N = C_\beta \frac{\xi}{1-\xi}$, then we get the sample complexity

$$n_{HN}(\epsilon, \delta, \xi_H) = n_H(\epsilon - C_\beta \frac{\xi_H}{1-\xi_H}, \delta) \cdot \frac{1}{1-\xi_H} \tag{63}$$

where the additional term $\frac{1}{1-\xi_H} = \frac{|S_C|+|S_N|}{|S_C|}$ is due to the fact that n_H only counts the number of clean data.

As the sample complexity $n_H(\epsilon, \delta)$ of $\mathcal{H}$ in the clean data case is $C_S \cdot \epsilon^{-d_H}$ where C_S and d_H are positive constants that depend on $\mathcal{H}$ and the confidence $1-\delta$. Under noise level of ξ, the sample complexity can increase to

$$\begin{aligned}
n_{HN}(\epsilon, \delta, \xi) &= C_S \cdot \epsilon^{-d_H} \cdot \left(1 - \frac{C_\beta \xi}{\epsilon(1-\xi)}\right)^{-d_H} \cdot \frac{1}{1-\xi} \\
&\geq C_S \cdot \epsilon^{-d_H} \cdot (1 - C_\beta\xi)^{-d_H} \\
&\geq C_S \cdot \epsilon^{-d_H} \cdot \exp(d_H C_\beta \xi) \\
&= n_H(\epsilon, \delta) \cdot \exp(d_H C_\beta \xi)
\end{aligned} \tag{64}$$

REFERENCES

[1] K. Q. Weinberger, J. Blitzer, and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," in *Proc. NIPS*, 2005, pp. 1473–1480.

[2] M. Guillaumin, J. Verbeek, and C. Schmid, "Is that you? Metric learning approaches for face identification," in *Proc. ICCV*, 2009, pp. 498–505.

[3] J. Chen, Z. Zhang, and Y. Wang, "Relevance metric learning for person re-identification by exploiting listwise similarities," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 4741–4755, Dec. 2015.

[4] V. Zantedeschi, R. Emonet, and M. Sebban, "Metric learning as convex combinations of local models with generalization guarantees," in *Proc. CVPR*, 2016, pp. 1478–1486.

[5] Y. Yang, S. Liao, Z. Lei, and S. Z. Li, "Large scale similarity learning using similar pairs for person verification," in *Proc. AAAI*, 2016, pp. 3655–3661.

[6] D. Tao, Y. Guo, M. Song, Y. Li, Z. Yu, and Y. Y. Tang, "Person re-identification by dual-regularized kiss metric learning," *IEEE Trans. Image Process.*, vol. 25, no. 6, pp. 2726–2738, Jun. 2016.

[7] J. Bosveld, A. Mahmood, D. Q. Huynh, and L. Noakes, "Constrained metric learning by permutation inducing isometries," *IEEE Trans. Image Process.*, vol. 25, no. 1, pp. 92–103, Jan. 2016.

[8] C. Sun, D. Wang, and H. Lu, "Person re-identification via distance metric learning with latent variables," *IEEE Trans. Image Process.*, vol. 26, no. 1, pp. 23–34, Jan. 2017.

[9] Z. Ding and Y. Fu, "Robust transfer metric learning for image classification," *IEEE Trans. Image Process.*, vol. 26, no. 2, pp. 660–670, Feb. 2017.

[10] W. Gao, L. Wang, Y.-F. Li, Z.-H. Zhou, "Risk minimization in the presence of label noise," in *Proc. AAAI*, 2016, pp. 1575–1581.

[11] B. Frenay and M. Verleysen, "Classification in the presence of label noise: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 5, pp. 845–869, May 2014.

[12] R. A. Krishna *et al.*, "Embracing error to enable rapid crowdsourcing," in *Proc. CHI Conf. Human Factors Comput. Syst.*, 2016, pp. 3167–3179.

[13] R. Jin, S. Wang, and Y. Zhou, "Regularized distance metric learning: Theory and algorithm," in *Proc. NIPS*, 2009, pp. 862–870.

[14] Z.-J. Zha, T. Mei, M. Wang, Z. Wang, and X.-S. Hua, "Robust distance metric learning with auxiliary knowledge," in *Proc. IJCAI*, 2009, pp. 1327–1332.

[15] M. Liu and B. C. Vemuri, "A robust and efficient doubly regularized metric learning approach," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 646–659.

[16] M. T. Law, N. Thome, and M. Cord, "Fantope regularization in metric learning," in *Proc. CVPR*, 2014, pp. 1051–1058.

[17] Q. Qian, J. Hu, R. Jin, J. Pei, and S. Zhu, "Distance metric learning using dropout: A structured regularization approach," in *Proc. SIGKDD*, 2014, pp. 323–332.

[18] Z. Huo, F. Nie, and H. Huang, "Robust and effective metric learning using capped trace norm: Metric learning via capped trace norm," in *Proc. SIGKDD*, 2016, pp. 1605–1614.

[19] N. D. Lawrence and B. Schölkopf, "Estimating a kernel Fisher discriminant in the presence of label noise," in *Proc. ICML*, 2001, pp. 306–313.

[20] H. Wang, F. Nie, H. Huang, and H. Huang, "Robust distance metric learning via simultaneous L1-norm minimization and maximization," in *Proc. ICML*, 2014, pp. 1836–1844.

[21] D. Wang and X. Tan, "Robust distance metric learning in the presence of label noise," in *Proc. AAAI*, 2014, pp. 1321–1327.

[22] J. Zhu, N. Chen, and E. P. Xing, "Bayesian inference with posterior regularization and applications to infinite latent SVMs," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1799–1847, 2014.

[23] J. Zhu, J. Song, and B. Chen. (2016). "Max-margin nonparametric latent feature models for link prediction." [Online]. Available: https://arxiv.org/abs/1602.07428

[24] L. Yang, R. Jin, and R. Sukthankar, "Bayesian active distance metric learning," in *Proc. UAI*, 2007, pp. 442–449.

[25] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley, "Stochastic variational inference," *J. Mach. Learn. Res.*, vol. 14, no. 1, pp. 1303–1347, 2013.

[26] A. Bellet, A. Habrard, and M. Sebban, "Similarity learning for provably accurate sparse linear classification," in *Proc. ICML*, 2012, pp. 1871–1878.

[27] P. Kar, B. K. Sriperumbudur, P. Jain, and H. C. Karnick, "On the generalization ability of online learning algorithms for pairwise loss functions," in *Proc. ICML*, 2013, pp. 441–449.

[28] Q. Cao, Z.-C. Guo, and Y. Ying, "Generalization bounds for metric and similarity learning," *Mach. Learn.*, vol. 102, no. 1, pp. 115–132, 2016.

[29] J. A. Aslam and S. E. Decatur, "On the sample complexity of noise-tolerant learning," *Inf. Process. Lett.*, vol. 57, no. 4, pp. 189–195, 1996.

[30] N. H. Bshouty, N. Eiron, and E. Kushilevitz, "PAC learning with nasty noise," *Theor. Comput. Sci.*, vol. 288, no. 2, pp. 255–275, 2002.

[31] S. Jabbari, R. C. Holte, and S. Zilles, "PAC-learning with general class noise models," in *Proc. KI*, 2012, pp. 73–84.

[32] N. G. Polson and S. L. Scott, "Data augmentation for support vector machines," *Bayesian Anal.*, vol. 6, no. 1, pp. 1–23, 2011.

[33] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2014.

[34] N. Natarajan, I. Dhillon, P. Ravikumar, and A. Tewari, "Learning with noisy labels," in *Proc. NIPS*, 2013, pp. 1196–1204.

[35] C. Li, J. Zhu, and J. Chen, "Bayesian max-margin multi-task learning with data augmentation," in *Proc. ICML*, 2014, pp. II-415–II-423.

[36] B. Shaby and D. Ruppert, "Tapered covariance: Bayesian estimation and asymptotics," *J. Comput. Graph. Statist.*, vol. 21, no. 2, pp. 433–452, 2012.

[37] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics Intell. Lab. Syst.*, vol. 2, nos. 1–3, pp. 37–52, 1987.

[38] J. Goldberger, G. E. Hinton, S. T. Roweis, and R. Salakhutdinov, "Neighbourhood components analysis," in *Proc. NIPS*, 2004, pp. 513–520.

[39] T. Mensink, J. Verbeek, F. Perronnin, and G. Csurka, "Metric learning for large scale image classification: Generalizing to new classes at near-zero cost," in *Proc. ECCV*, 2012, pp. 488–501.

[40] W. Yao, Z. Weng, and Y. Zhu, "Diversity regularized metric learning for person re-identification," in *Proc. ICIP*, 2016, pp. 4264–4268.

[41] P. H. Zadeh, R. Hosseini, and S. Sra, "Geometric mean metric learning," in *Proc. ICML*, 2016, pp. 2464–2471.

[42] Y. Wang *et al.*, "Learning a discriminative distance metric with label consistency for scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 8, pp. 4427–4440, Aug. 2017.

[43] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[44] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. CVPR*, 2009, pp. 248–255.

[45] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, "MS-celeb-1m: A dataset and benchmark for large-scale face recognition," in *Proc. ECCV*, 2016, pp. 87–102.

[46] D. Wang and X. Tan, "Unsupervised feature learning with C-SVDDNet," *Pattern Recognit.*, vol. 60, pp. 473–485, Dec. 2016.

[47] K. He, X. Zhang, S. Ren, and J. Sun. (2015). "Deep residual learning for image recognition." [Online]. Available: https://arxiv.org/abs/1512.03385

[48] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *Proc. BMVC*, 2015, p. 6.

[49] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," Univ. Massachusetts, Amherst, MA, USA, Tech. Rep. 07-49, 2007.

[50] O. Bousquet and A. Elisseeff, "Stability and generalization," *J. Mach. Learn. Res.*, vol. 2, pp. 499–526, Mar. 2002.

Dong Wang is currently pursuing the Ph.D. degree with the Nanjing University of Aeronautics and Astronautics, Nanjing, China. His research interests include robust learning with label noise, Bayesian learning, and metric learning.

Xiaoyang Tan was a Post-Doctoral Researcher with the LEAR Team, INRIAR Rhone-Alpes, Grenoble, France, from 2006 to 2007. He is currently a Professor with the Department of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China. He has authored or co-authored over 50 conference and journal papers. His research interests include deep learning, reinforcement learning, and Bayesian learning. He and his colleagues were awarded the IEEE Signal Processing Society Best Paper in 2015.