Bayesian Neighborhood Component Analysis

Dong Wang and Xiaoyang Tan

***Abstract*— Learning a distance metric in feature space potentially improves the performance of the *K* nearest neighbor classifier and is useful in many real-world applications. Many metric learning (ML) algorithms are, however, based on the point estimation of a quadratic optimization problem, which is time-consuming, susceptible to overfitting, and lacks a natural mechanism to reason with parameter uncertainty—a property useful especially when the training set is small and/or noisy. To deal with these issues, we present a novel Bayesian ML (BML) method, called *Bayesian neighborhood component analysis (NCA)*, based on the well-known NCA method, in which the metric posterior is characterized by the local label consistency constraints of observations, encoded with a similarity graph instead of independent pairwise constraints. For efficient Bayesian inference, we explore the variational lower bound over the log-likelihood of the original NCA objective. Experiments on several publicly available data sets demonstrate that the proposed method is able to learn robust metric measures from small size data set and/or from challenging training set with labels contaminated by errors. The proposed method is also shown to outperform a previous pairwise constrained BML method.**

***Index Terms*— Bayes modeling, distance metric learning, label noise, neighborhood component analysis.**

I. Introduction

LEARNING a good distance metric in feature space is crucial in many real-world applications. It has been shown to significantly improve the performance of object classification [1], image retrieval [2], image ranking [3], face identification [4], kinship verification [5], clustering [6], or person reidentification [7]. Most of distance metric learning (DML) methods aim to learn a linear transformation, which pulls together samples from the same class while pushing away those from different classes.

There has been considerable research on DML over the past few years [8]–[14]. Among them, neighborhood component analysis (NCA) [15] is a well-known DML method, which is conceptually simple and is developed under a well-formulated probabilistic framework with graph label consistency constraints. There are also several extensions of this method in the literature, such as the large margin nearest neighbor (LMNN) [16], nearest class mean (NCM) [1], label noise robust NCA [17], and so on.

Manuscript received April 7, 2016; revised January 5, 2017, April 13, 2017, and May 31, 2017; accepted June 4, 2017. This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFB0802300, in part by the National Science Foundation of China under Grant 61373060 and Grant 61672280 and in part by the Qing Lan Project. *(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/TNNLS.2017.2712823

Although ML algorithms have achieved great success, in many real-world applications, its performance may be hurt by two problems: 1) collecting a large number of labeled data for DML training is laborious and not easy and 2) even when one has one large data set, more often than not the quality of the collected data cannot be guaranteed. Moreover, most of the DML algorithms are based on point estimation, which is sensitive to the choice of training examples and tends to be overfitting especially when training set is small or noisy. Many robust learning methods have been proposed [18]–[23], but they usually have high computation cost and do not focus on ML. The recently proposed pairwise constrained Bayesian ML (BML) method [24] tries to address these issues by taking the prior distribution of the transformation matrix into account. However, it treats each sample independently and ignores the different importance of each sample, which limits its efficiency in learning.

In this paper, we present a graph constrained BML method to address the above-mentioned issues. The method is based on the NCA method but, for the first time, extends it under the Bayesian framework, hence called *Bayesian NCA (BNCA)*. To be concrete, unlike previous studies on ML methods, our method has the following advantages.

1) It naturally takes account the influence of parameter uncertainty and is less susceptible to overfitting by exploiting the prior knowledge.
2) It provides robust estimation even when there are errors in data annotation, with the help of graph label consistency constraints and the adopted local variational training method.
3) It significantly reduces the computational cost while preserving the effectiveness of Bayesian estimation, due to a newly developed variational lower bound of the log-likelihood objective.

We verify the effectiveness of the proposed method on several real-world applications, including image classification, digital recognition, and face recognition. Our results demonstrate that the BNCA method is able to learn robust metric measures from small size data sets or from data sets with noisy labels. It is also shown to outperform a previous pairwise constrained BML method [24] and several other state-of-the-art DML methods.

The remaining parts of this paper are organized as follows. In Section II, preliminaries are provided; then, we detail our proposed method in Section III, make some analysis of it in Section IV, and verify its performance in Section V. We conclude this paper in Section VI.

II. Preliminary

Assuming that we have a data set D of N data points, denoted as $D = \{x_i,\ y_i\}, i = 1, 2, 3, \ldots, N$, where y_i is

the label of the ith data point x_i. In DML, we aim to learn a Mahalanobis matrix—A using some form of supervision information. Mahalanobis distance metric measures the squared distance between two data points x_i and x_j as follows:

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

where $A \geq 0$ is a positive semidefinite matrix and $x_i, x_j \in R^d$ is a pair of samples (i, j). For simplicity, we denote $d_A^2(x_i, x_j)$ as $d_{A^{ij}}^2$. With these notations, in what follows, we give a brief overview on two state-of-the-art works closely related to ours in learning a Mahalanobis metric, i.e., NCA [15] and pairwise constrained BML [24].

A. Neighborhood Component Analysis

The NCA algorithm [15] begins by constructing a complete graph with each data point as its node. Let the weight of each edge between any two nodes denoted as p_{ij}. It is interpreted as the probability that data point x_i selects x_j as its neighbor and can be calculated as follows:

$$p_{ij} = \frac{\exp\left(-d_{A^{ij}}^2\right)}{\sum_{t \in N_i} \exp\left(-d_{A^{it}}^2\right)} \tag{2}$$

where N_i denotes the set of neighbors of x_i. It can be checked that $p_{ij} \geq 0$ and $\sum_{j \in N_i} p_{ij} = 1$, and hence, p_{ij} is a valid probability measure.

The object of NCA is then to learn a linear transformation A, which maximizes the log likelihood that after transformation, each data point selects the points with the same labels as itself as neighbors, i.e.,

$$\max \; L(A) = \sum_i \log \left(\sum_{j \in N_i} 1\{y_i = y_j\} \cdot p_{ij} \right). \tag{3}$$

B. Pairwise Constrained Bayesian Metric Learning

Yang *et al.* [24] proposed a BML method that estimates the posterior distribution for the distance metric from labeled pairwise constraints. It defines the probability for two data points x_i and x_j to form an equivalence or inequivalence constraint under a given distance metric A

$$P(y_{ij}|x_i, x_j, A, \mu) = \frac{1}{1 + \exp(y_{ij}(d_{A^{ij}}^2 - \mu))} \tag{4}$$

$$\text{where} \quad y_{ij} = \begin{cases} +1 & (x_i, x_j) \in \mathcal{S} \\ -1 & (x_i, x_j) \in \mathcal{D}. \end{cases} \tag{5}$$

As mentioned earlier, $\mathcal{S}$ and $\mathcal{D}$, respectively, denote the sets of equivalence or inequivalence constraints. Given this, the posterior distribution of metric A and the threshold μ can be estimated by maximizing the following objective:

$$L(A, \mu) = \prod_{(i,j)} P(y_{ij}|x_i, x_j, A, \mu) p(A) p(\mu). \tag{6}$$

This method effectively overcomes some of the limitations of traditional ML methods. However, it does not take the structure of data into consideration and does not scale well. Particularly, since its objective just requires that the distance between similar pairs of points should be lower than that between dissimilar ones, all pairs (i, j) $(o(N^2))$ need to be calculated for training. This not only increases the computational cost, but also ignores the importance weight of each sample regards to model training, which significantly decreases the learning efficiency, because, ideally, we should focus more on those data whose labels are not consistent with most of its neighbors, instead of treating them indifferently. This problem is partially addressed later by Yang *et al.* [24] with an active learning method for data pair selection, but the computational cost remains high.

III. Bayesian Neighborhood Component Analysis

A. Proposed Method

We start our derivation by considering the three components of a general Bayesian model, i.e., prior, likelihood, and posterior. Since the original NCA is a discriminant model, we write its likelihood as $P(Y|X, A)$ in our BNCA, where A is the linear transformation matrix to be learned. We follow the same assumption as that of NCA, i.e., the sample labels are conditionally independent given the labels of their nearest neighbors. Hence, the conditional model can be written as

$$P(Y|X, A) = \frac{1}{Z(A)} \prod_i P(y_i|x_i, Y_{N_i}, X_{N_i}, A) \tag{7}$$

where $Z(A)$ is a normalizing constant known as the partition function. To be consistent with NCA, we define

$$P(y_i = k|x_i, Y_{N_i}, X_{N_i}, A) = \frac{\sum_{j \in N_i} 1\{y_j = k\} \cdot \exp\left(-d_{A^{ij}}^2\right)}{\sum_{t \in N_i} \exp\left(-d_{A^{it}}^2\right)}. \tag{8}$$

Comparing (8) with (4), we see that one of the major differences between our model and the BML lies in that the local neighborhood structure N_i is naturally embedded into the model in our method.

To compute the posterior of the distance metric A, a prior for it should be specified, and a convenient choice for this could be the Wishart prior. Unfortunately, it is well known that combining the Wishart prior with a non-Gaussian likelihood is difficult to compute. In addition, the integration of A is intractable as well.

To bypass the above-mentioned issues, we first approximate the distance metric A as a linear combination of the top eigenvectors of the observed data, and then estimate the posterior distribution of the combination weights using variational method.

1) Eigenapproximation: Let $X = (x_1, x_2, \ldots, x_N)$ denote all the examples, and $v_l, (l = 1, 2, \ldots, d)$ be the top d eigenvectors of XX^T. Inspired by [24], we approximate A using the first d eigenvectors, i.e., $A = \sum_{l=1}^{d} \gamma_l v_l v_l^T$, where $\gamma_l, (l = 1, 2, \ldots, d)$ are the combination coefficients. With this, the likelihood $P(y_i = k|x_i, Y_{N_i}, X_{N_i}, A)$ in (8) reduces to its equivalent form $P(y_i = k|x_i, Y_{N_i}, X_{N_i}, \gamma)$

$$P(y_i = k|x_i, Y_{N_i}, X_{N_i}, \gamma) = \frac{\sum_{j \in N_i} 1\{y_j = k\} \cdot \exp\left(-d_{\gamma^{ij}}^2\right)}{\sum_{t \in N_i} \exp\left(-d_{\gamma^{it}}^2\right))} \tag{9}$$

where we define

$$\begin{aligned} w_{ij}^{l} &= \left(v_l^T (x_i - x_j)\right)^2 \\ w_{ij} &= \left[w_{ij}^1, w_{ij}^2, \ldots, w_{ij}^d\right]^T \\ \gamma &= [\gamma_1, \gamma_2, \ldots, \gamma_d]^T \end{aligned} \tag{10}$$

then $d^2_{\gamma^{ij}} = d^2_{A^{ij}} = \gamma^T w_{ij}$.

Our task then boils down to compute the posterior distribution of γ. For simplicity, we assume that the prior distribution of γ to be Gaussian

$$p(\gamma) = N(\gamma | m_0, V_0) \tag{11}$$

where m_0 and V_0 are, respectively, mean and covariance.

2) Variational Approximation: At the second step, we employ the variational method to estimate the posterior distribution of γ. The main idea is to introduce variational distributions for γ to construct the lower bound and then maximize the lower bound to obtain the approximate estimation for the posterior distribution. We begin with the unnormalized logarithm likelihood $\log\{Z(A)P(Y|X,\gamma)\}$. Note that maximizing this objective directly regarding to A leads to the standard NCA algorithm, but our goal here is for local variational approximation; hence, the partition function $Z(A)$ is simply treated as a constant. Particularly

$$\begin{aligned} L &= \log\{Z(A)P(Y|X,\gamma)\} \\ &= \sum_i \sum_k 1\{y_i = k\} \log\{p(y_i = k | x_i, Y_{N_i}, X_{N_i}, \gamma)\} \\ &= \sum_i \sum_k 1\{y_i = k\} \log\left\{ \frac{\sum_{j \in N_i} 1\{y_j = k\} \cdot \exp\left(-d^2_{\gamma^{ij}}\right)}{\sum_{t \in N_i} \exp\left(-d^2_{\gamma^{it}}\right)} \right\}. \end{aligned} \tag{12}$$

Since $\log(a+b) > \log(a) + \log(b)$ if $0 < a, b < 1$, we have

$$\begin{aligned} L &> \sum_i \sum_{j \in N_i} y_{ij} \log\left\{ \frac{\exp\left(-d^2_{\gamma^{ij}}\right)}{\sum_{t \in N_i} \exp\left(-d^2_{\gamma^{it}}\right)} \right\} \\ &> \sum_i \sum_{j \in N_i} y_{ij} \log\left\{ \frac{1}{1 + \sum_{t \in N_i} \exp\left(d^2_{\gamma^{ij}} - d^2_{\gamma^{it}}\right)} \right\}. \end{aligned} \tag{13}$$

Let $x_{N_{i_1}}, x_{N_{i_2}}, \ldots, x_{N_{i_K}}$ be, respectively, the K nearest neighbors (KNNs) of x_i. For convenience, we introduce the following notations:

$$\begin{aligned} \eta_{ij}^t &= d^2_{\gamma^{ij}} - d^2_{\gamma^{it}} = (w_{ij} - w_{it})^T \gamma \\ W_i^j &= [w_{ij} - w_{iN_{i_1}}, w_{ij} - w_{iN_{i_2}}, \ldots, w_{ij} - w_{iN_{i_K}}] \\ \eta_{ij} &= \left[\eta_{ij}^{N_{i_1}}, \eta_{ij}^{N_{i_2}}, \ldots, \eta_{ij}^{N_{i_K}}\right]^T = \left(W_i^j\right)^T \gamma. \end{aligned} \tag{14}$$

Recall the definition of log-sum-exp function

$$\mathrm{lse}(\eta_{ij}) \triangleq \log\left(1 + \sum_{t \in N_i} \exp\left(\eta_{ij}^t\right)\right). \tag{15}$$

Then, (13) can be rewritten as

$$L > -\sum_i \sum_{j \in N_i} y_{ij} \mathrm{lse}(\eta_{ij}). \tag{16}$$

Algorithm 1 BNCA

Input:

Input: Training set $\{(x_i, y_i) | \ i = 1, 2, \ldots, N\}$, prior distribution $\mathcal{N}(\gamma | m_0, V_0)$;

Output:

posterior distribution $\mathcal{N}(\gamma | m_T, V_T)$

—— Training Stage

1: Define W_i^j, H according to (14) and (18) respectively.
2: Compute V_T with eq. (22).
3: Repeat
4: compute ψ_{ij} for all (i, j) with eq. (24)
5: compute b_{ij} for all (i, j) with eq. (19)
6: compute m_T with Eq. (23).
7: Until converged.
8: Return $\mathcal{N}(\gamma | m_T, V_T)$.

Using Bohning's quadratic bound (see [25, p. 758, Sec. 21.8.2] for details), we have

$$\begin{aligned} L &> \sum_i \sum_{j \in N_i} y_{ij} \left\{ -\frac{1}{2} \eta_{ij}^T H \eta_{ij} + b_{ij}^T \eta_{ij} - c_{ij} \right\} \\ &= \sum_i \sum_{j \in N_i} y_{ij} \left\{ -\frac{1}{2} \gamma^T W_i^j H \left(W_i^j\right)^T \gamma + b_{ij}^T \left(W_i^j\right)^T \gamma - c_{ij} \right\} \end{aligned} \tag{17}$$

where c_{ij} is a constant and the remaining notations are defined as

$$H = \frac{1}{2}\left[I_K - \frac{1}{K+1} 1_K 1_K^T \right] \tag{18}$$

$$b_{ij} = H \psi_{ij} - g(\psi_{ij}) \tag{19}$$

$$g(\psi_{ij}) = \exp(\psi_{ij} - \mathrm{lse}(\psi_{ij})). \tag{20}$$

Note that ψ_{ij} is the variational parameter.

Now, we proceed to compute the posterior distribution of γ, which we model as a Gaussian, denoted as $\mathcal{N}(\gamma | m_T, V_T)$. We write the unconstrained posterior distribution

$$p(\gamma | X, Y) \propto p(Y | X, \gamma) p(\gamma) \tag{21}$$

and plug in the approximated likelihood (17) and the prior distribution $\mathcal{N}(\gamma | m_0, V_0)$ to get

$$V_T = \left[V_0^{-1} + \sum_i \sum_{j \in N_i} y_{ij} W_i^j H \left(W_i^j\right)^T \right]^{-1} \tag{22}$$

$$m_T = V_T \left(V_0^{-1} m_0 + \sum_i \sum_{j \in N_i} y_{ij} W_i^j b_{ij} \right). \tag{23}$$

Finally, the variational parameter ψ_{ij} is updated as

$$\psi_{ij} = \left(W_i^j\right)^T m_T. \tag{24}$$

We summarize the proposed method in Algorithm 1.

B. Distance Estimation

For inference, we are interested in the expectation of the point-to-point distance $d^2_{\gamma^{ij}}$ for a new couple of data (i, j)

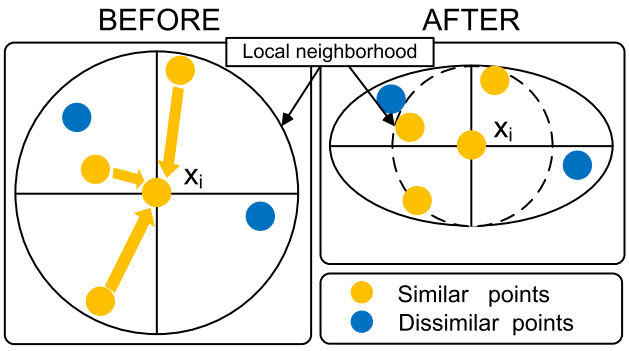


Fig. 1. BNCA appropriately scale the axis to allow x_i's neighbors in the same class to be closer.

according to the posterior distribution of γ, which is a Gaussian distribution as shown earlier. Particularly, we have

$$d^2_{\gamma ij} \sim \mathcal{N}\left(d^2_{\gamma ij} | m_{ij}, \sigma^2_{ij}\right) \tag{25}$$

where

$$\begin{aligned} m_{ij} &= (w_{ij})^T m_T \\ \sigma^2_{ij} &= (w_{ij})^T V_T w_{ij}. \end{aligned} \tag{26}$$

It is worthwhile to mention that this mechanism of outputting model uncertainty in distance metric calculation is potentially beneficial to many real-world applications but unfortunately is largely ignored in the field. For example, in the application of image retrieval rather than ranking the results purely based on the estimated similarity, we could now construct a more robust ranking scheme by taking the value of the related similarity uncertainty (i.e., σ^2_{ij}) into account. We would not pursue this issue any further as it is out of the range of this paper, but it will be the focus of our future work.

Instead, one could simply use the mean value—m_{ij} to estimate each $d^2_{\gamma ij}$. To see the difference between this with the traditional NCA method, we decompose its expectation as follows:

$$\begin{aligned} E(d^2_{\gamma ij}) &= \sum_l w^l_{ij} m^l_T \\ &= \sum_l (x_i - x_j)^T v_l m^l_T v^T_l (x_i - x_j) \end{aligned} \tag{27}$$

where w^l_{ij} and m^l_T are the lth element of w_{ij} and m_T, respectively.

Now defining the new coordinate axes as $[v'_1, v'_2, \ldots, v'_d]$ with $v'_l = (m^l_T)^{(1/2)} \cdot v_l$, we see that the inference equation (26) essentially calculates the distance in a feature space spanned by the top d the eigenvectors of XX^T but scaled by $(m_T)^{\frac{1}{2}}$, according to the distribution of the corresponding eigenvalues (see Fig. 1).

C. Prediction Under Parameter Uncertainty

Under the difficult condition of small size training samples or samples with label noise, a single estimate of parameter A tends to be unreliable, and the traditional DML methods that are based on it may cause overconfidence in the future predictions. In other words, they just make predictions but cannot tell whether these predictions make sense. By contrast, for Bayesian methods, this is really not a problem, because no errors would be introduced due to the inaccurate estimation of A. Particularly, the prediction for a never-seen sample x_i can be obtained from $p(y_i|x_i, Y_{N_i}, X)$. Recall that the variational posterior of metric parameter γ is a Gaussian distribution, i.e., $q(\gamma) = \mathcal{N}(\gamma | m^T, V^T)$ [see (22) and (23)], we have

$$p(y_i|x_i, Y_{N_i}, X_{N_i}) = \int_\gamma p(y_i|x_i, Y_{N_i}, X_{N_i}, \gamma) q(\gamma) d\gamma. \tag{28}$$

The difficulty here is that this integration of γ is untractable, because $p(y_i|x_i, Y_{N_i}, X, \gamma)$ is a multinomial distribution while $q(\gamma)$ is a Gaussian one. Instead, we adopt an Markov chain Monte Carlo (MCMC) method [26] to approximate this expectation

$$p(y_i|x_i, Y_{N_i}, X_{N_i}) \approx \frac{1}{T} \sum_{l=1}^{T} p(y_i|x_i, Y_{N_i}, X_{N_i}, \gamma_l) \tag{29}$$

where $\gamma_l (l = 1, 2, \ldots, T)$ are sampled i.i.d. from $q(\gamma)$.

IV. Analysis of the Proposed Method

A. Adaptive Sample Selection in Learning

In the process of ML, it is beneficial to exploit the local property of samples in the input space to improve the learning efficiency. Take the NCA algorithm as an example. Its gradient is calculated as follows:

$$\frac{\partial L}{\partial A} = 2A \sum_i \left(\sum_{j \in N_i} p_{ij} x_{ij} x^T_{ij} - \frac{\sum_{j \in N_i} y_{ij} p_{ij} x_{ij} x^T_{ij}}{\sum_{j \in N_i} y_{ij} p_{ij}} \right). \tag{30}$$

That is, for any point x_i, if and only if all its KNNs have the same labels as that of x_i, then $\sum_{j \in N_i} y_{ij} p_{ij} = 1$, which means that the gradient equals to $\vec{0}$. Hence, the NCA algorithm would pay more attention on those points whose labels are inconsistent with its KNNs. In other words, not all pairs (x_i, x_j) are active in constraining the search space of the transformation matrix in the same way.

Similar observations can be made in other NCA extensions, such as the large margin nearest neighbor (LMNN) [16] method

$$\min \; L(A) = \sum_i \sum_{j \in N_i} y_{ij} \left(d^2_{\gamma ij} + \mu \sum_{l \in N_i} (1 - y_{il}) \xi_{ijl}(A) \right) \tag{31}$$

where the first term can be seen as a regularizer while the second one penalizes those data points that violate the large margin condition.

But the situation is different for pairwise constrained models [24] in the sense that they usually lack an automatic sample selection mechanism in the ML objective by itself [see (6)]. In our opinion, it is important to give different importance weights to different points during training, because doing this properly potentially allows us to significantly reduce the computational costs and to lessen the likelihood of overfitting. Actually, to avoid computing distance for all possible data pairs, Yang *et al.* [24] designed an effective active learning method to select the most uncertainty pairs in training process, but the algorithm still needs to compute and store all the pairs' uncertainty scores.

Let us come to our results of BNCA shown in (22) and (23). For simplicity, here, we only care about the diagonal elements V_T^{ll} $(l = 1, 2, \ldots, d)$ of V_T. Let us define $W_i^{j^l}$ as the lth row of W_i^j $(W_i^j = [W_i^{j^1}, W_i^{j^2}, \ldots, W_i^{j^d}]^T)$

$$W_i^{j^l} = \left[w_{ij}^l - w_{iN_{i1}}^l, w_{ij}^l - w_{iN_{i2}}^l, \ldots, w_{ij}^l - w_{iN_{iK}}^l\right]. \quad (32)$$

From (22), we get

$$V_T^{ll} = \left(\left(V_0^{ll}\right)^{-1} + \sum_i \sum_{j\in N_i} \sum_{t\in N_i} y_{ij} W_i^{j^l} H_{ll} \left(W_i^{j^l}\right)^T\right)^{-1}. \quad (33)$$

Assuming that $K \gg 1$, H can be approximated by $(1/2)I$, such that

$$V_T^{ll} = \left(\left(V_0^{ll}\right)^{-1} + \frac{1}{2}\sum_i \sum_{j\in N_i} y_{ij} \sum_{t\in N_i} \left(w_{ij}^l - w_{it}^l\right)^2\right)^{-1}. \quad (34)$$

If we simply throw away the nondiag elements of V_T from (23), we see that V_T^{ll} is in proportion to m_T^l. In other words, in BNCA, we scale the axis v_l by reducing the variance of γ_l, such that all x_i's neighbors in the same class will be closer in that direction, as shown in Fig. 1.

To see how our proposed BNCA handles different data points adaptively, we consider the following two extreme circumstances: 1) all x_i's nearest neighbors have the same labels as that of x_i and have the same distance to x_i and 2) none of x_i's nearest neighbors belongs to the same class of x_i. In both cases, the term $\sum_{j\in N_i} y_{ij} \sum_{t\in N_i} (w_{ij}^l - w_{it}^l)^2$ [see (34)] equals to 0, such that the variance will not be changed by x_i. In the first circumstance, those x_i can be thought of as perfect points that do not need to be adjusted, while the second case illustrates how our method handles the data in a robust way when some of them lie on the decision boundary or when their labels are too noisy to be learned from.

B. Robustness Against Label Noise

To reveal the influence of label noise on the training of a DML model, we start the analysis with the NCA method. First, let us denote the two major components of its gradient (30) as C_E and C_I, respectively

$$C_E = \sum_i \sum_{j\in N_i} p_{ij} x_{ij} x_{ij}^T \quad (35)$$

$$C_I = \sum_i \frac{\sum_{j\in N_i} y_{ij} p_{ij} x_{ij} x_{ij}^T}{\sum_{j\in N_i} y_{ij} p_{ij}} \quad (36)$$

$$= \sum_i \sum_j p_{ij} x_{ij} x_{ij}^T \sum_k \frac{1(y_i = k) \cdot 1(y_j = k)}{\sum_j 1(y_j = k) \cdot p_{ij}}. \quad (37)$$

We see that

$$\frac{\partial L}{\partial A} = 2A(C_E - C_I). \quad (38)$$

Intuitively, the C_E term denotes the total scatter matrix of the data points lying on the manifold induced by A and C_I is the corresponding intraclass scatter matrix (38) reveals that, up to a constant matrix, in each step, the NCA algorithm tries to seek a better linear transformation, such that after projection, the total covariance becomes larger while the intraclass covariance becomes smaller. However, when the class labels are inaccurate or noisy, the estimation of C_I tends to be inaccurate (the C_E value will be not influenced by this).

The same situation occurs in LMNN. As can be seen from (31), label noise would possibly result in a lot of incorrect training triples (ijl), which pull together samples from different class while pushing away those from the same classes. This issue becomes more and more troublesome with the increase in the noise level, and actually, all the DML techniques trained in a supervised way would suffer from this if not properly taken care of. This is witnessed by our experiments given later, showing that under some high noise level, many traditional start-of-the-art DML methods, such as NCA and LMNN, will even be inferior to the unsupervised baseline, i.e., principal component analysis (PCA).

As for the proposed BNCA model, there are two sources of regularization: one is the incorporation of the prior distribution, and the other is through eigenapproximation (spectral) and local variational inference. Both are useful against label noise, but in our opinion, the latter one plays a more important role: recall that for a nonconvex objective function, an observation with label noise could potentially change the locations and number of its local minima. Adding penalty onto the objective [as in penalized ML estimator or equivalently, maximum *a posteriori* estimator] using prior distribution helps but is not enough, as the influence of the prior becomes weaker with the increasing amount of data, while the local variational inference replaces each term of the joint distribution with a Gaussian, leading to a Gaussian-like lower bound to the global objective. Such mechanism effectively smooths the influence of observations with label noise, making the optimization process much easier. This advantage is clearly independent of the size of training data.

C. Computational Cost

To analyze the computational cost of the proposed method, first note that usually the most time-consuming step in Laplace approximation or a conjugate gradient method is related to the calculation of Hessian matrix. In each iteration, it needs $o(N^3)$ computational cost (where N is the number of training data, e.g., if N is 10^3, the computational cost could be as large as 10^9). While in our case, thanks to the Bohning's approximation, the Hessian matrix becomes a constant matrix [see (18)], calculated only once. Furthermore, our BNCA method avoids the time-consuming gradient iterations completely—the lower bound of (17) actually gives us an analytic solution [see (22) and (23)].

V. Experiments

To verify the effectiveness of the proposed method, in this section, we first compare the robustness performance of our method with several related DML methods on the data sets either with small sample size or with label noise; then, we turn to investigate in depth the behavior of the proposed method.

A. Experimental Settings

We compare the performance of the proposed method with several other closely related DML methods, including

TABLE I

COMPARATIVE PERFORMANCE (%) ON UCI DATA SETS WITH VARYING SIZES OF TRAINING SET. (THE ASTERISKS INDICATE A STATISTICALLY SIGNIFICANT DIFFERENCE BETWEEN THE SECOND BEST PERFORMER AND THE PROPOSED METHOD AT A SIGNIFICANCE LEVEL OF 0.05)

per class (#)	L1	PCA	NCA	NCM	LMNN	BML	BNCA
Balance 10	69.07± 0.20	69.42± 0.20	67.85± 0.35	66.28± 0.20	70.38± 0.33	71.38± 0.33*	**75.42± 0.31**
Balance 20	74.34± 0.12	74.53± 0.12	74.38± 0.29	72.62± 0.13	77.38± 0.28*	76.74± 0.26	**80.76± 0.26**
Balance 30	77.17± 0.10	77.23± 0.10	79.38± 0.18	76.16± 0.10	81.18± 0.16*	79.74± 0.14	**83.26± 0.14**
Ionosphere 10	68.58± 0.17	68.91± 0.17	70.97± 0.28	68.91± 0.17	73.54± 0.30	73.68± 0.28*	**76.75± 0.28**
Ionosphere 20	72.86± 0.15	73.12± 0.15	75.75± 0.21	73.19± 0.15	78.64± 0.23*	76.52± 0.20	**81.22± 0.21**
Ionosphere 30	75.98± 0.11	76.14± 0.11	80.96± 0.17	77.82± 0.10	83.88± 0.16*	80.86± 0.16	**85.96± 0.16**
Spambase 10	72.94± 0.23	73.42± 0.23	70.38± 0.37	70.08± 0.24	72.85± 0.35	75.38± 0.35*	**79.42± 0.34**
Spambase 20	77.17± 0.20	77.53± 0.20	76.74± 0.24	75.56± 0.20	79.38± 0.25*	78.52± 0.24	**83.76± 0.24**
Spambase 30	79.41± 0.14	79.53± 0.14	81.74± 0.23	79.74± 0.14	84.38± 0.23*	80.97± 0.22	**86.76± 0.22**

NCA [15], LMNN [16], ML for NCM [1], and pairwise constrained BML [24]. Both the NCA and the LMNN are ML methods with graph constraints, and the NCA is the method our method based on, hence chosen to be the baseline algorithm. Like ours, the BML method proposed by Yang *et al.* [24] is a Bayesian method as well, but with pairwise constraints for learning. We also adopted two unsupervised methods as baselines: PCA [27] and L1 distance, since unsupervised learning algorithms are essentially irrelevant to the issue of label noise.

For all the methods except the NCM (which has its own classifier), we used the KNN method equipped with the corresponding learned metric for classification. In addition, the performance of all the compared methods is based on the original implementation kindly provided by the corresponding authors, and the related hyperparameters are fine-tuned through cross validation. 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.

Implementation Details: In BNCA, there are a few parameters need to be initialized, mainly including the parameters of the prior distribution $\mathcal{N}(\gamma\,|m_0, V_0)$ and local variational approximation parameters, including b_{ij} (19) and ψ_{ij} (24). In general, searching a right prior distribution is difficult, although there exist several methods for that. For example, McKay's evidence maximization [28] deals with this issue by learning it directly from data (based on the principle of evidence maximization). Ideally, we could use McKay's framework to learn the prior and then proceeds with the variational inference method to approximate the posterior of distance metric. Unfortunately, adopting an empirical Bayes method (e.g., the evidence maximization) is challenging under our BNCA model, due to the difficulty in making analytical approximations of the marginal maximum likelihood estimator—our likelihood function is non-Gaussian, and hence, it needs further approximation to obtain the explicit form of the energy function of the parameters. Another reason is that, establishing a particular specification of the prior distribution is not so necessary in our case as we only need something kind of weakly informative prior distribution to regularize the posterior distribution of the distance metric. Hence, in experiments, we simply use the grid search method to look for a roughly good prior distribution. In particular, we set m_0 to $\epsilon\vec{1}$, where $\vec{1}$ is all 1's vector and ϵ is a small scalar (e.g., 0.1). From Section III-B, we see that this choice of m_0 is equivalent to initialize BNCA with PCA, which is commonly used in ML for initialization and will not be affected by label noise. Besides, we set V_0 to $\sigma^2\,I$, where σ^2 is a very small value (e.g., 0.001). This helps to preserve the stability of V_T (22), one important property related to overfitting. Then, we compute b_{ij} and ψ_{ij} according to (19) and (24), respectively.

B. Learning From Small Size Training Set

First, we investigate the performance of our method with small sample size on three UCI data sets ("Balance," "Ionosphere," and "Spambase"). In each data set, we randomly sample three subsets as training set with the size of $10\times\ C$, $20\times C$, and $30\times C$ (C is the number of categories), respectively, and use an extra subset containing 100 data points as test set.

Table I gives the classification performance. One can see that when the training set is small, point estimation-based methods tend to be unreliable. With only ten training samples, the standard NCA performs even worse than the unsupervised baseline approaches (PCA and L1) on two of the three data sets tested. By contrast, the proposed BNCA performs the best among the compared methods, partly due to the advantage of the Bayesian framework, which potentially provides reliable estimation even when the size of the training data is small.

Table I also shows that with increasing number of training points, the performance of all the methods considered here improves a lot. As expected, when we sample 30 data from each class, the performance gap between the Bayesian approaches and the point estimation-based methods (such as LMNN) becomes small.

C. Learning Under Random Label Noise

To test the performance of our method under label noise, we tested our method on several real-world applications, including image classification (on the Caltech-10 data set [29]), digital recognition (on the MNIST [30]), and face recognition (on the FRGC-2 [31]). The data sets adopted are popular benchmark on each of the task, respectively.

1) Caltech-10 is a subset sampled from Caltech-256 image data set [32] with ten most popular categories. The training set contains 300 images (30 from each

TABLE II

COMPARATIVE PERFORMANCE (%) ON DIFFERENT DATA SETS 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 (%)	L1	PCA	NCA	NCM	LMNN	BML	BNCA
Caltech 0%	80.04± 0.12	80.12± 0.12	81.49± 0.20	76.65± 0.20	82.84± 0.21*	80.68± 0.20	**83.89± 0.20**
Caltech 10%	78.63± 0.14	79.02± 0.14	79.03± 0.22	74.49± 0.22	79.38± 0.23*	79.11± 0.22	**81.79± 0.22**
Caltech 20%	75.07± 0.14	76.15± 0.14	74.59± 0.26	68.97± 0.26	74.27± 0.28	76.53± 0.26*	**79.41± 0.24**
Caltech 30%	67.40± 0.16	68.96± 0.16	66.05± 0.34	61.84± 0.34	65.09± 0.36	68.31± 0.34*	**73.79± 0.31**
MNIST 0%	89.24± 0.12	89.12± 0.12	92.05± 0.20	92.19± 0.20	92.84± 0.21	90.68± 0.20	**93.21± 0.20**
MNIST 10%	87.13± 0.14	87.49± 0.14	90.35± 0.22	88.91± 0.23	91.08± 0.23	89.11± 0.22	**91.79± 0.22**
MNIST 20%	84.39± 0.14	85.15± 0.14	84.59± 0.26	83.21± 0.27	84.27± 0.28	86.53± 0.26*	**88.41± 0.24**
MNIST 30%	79.82± 0.16	80.96± 0.16	78.84± 0.34	76.37± 0.35	78.09± 0.36	80.31± 0.34*	**84.38± 0.31**
FRGC 0%	94.35± 0.17	94.39± 0.17	95.52± 0.21	93.81± 0.21	**98.50± 0.19**	94.42± 0.19	**98.50± 0.18**
FRGC 10%	89.13± 0.18	89.54± 0.18	88.85± 0.25	85.93± 0.25	90.12± 0.25*	89.35± 0.25	**93.61± 0.24**
FRGC 20%	82.91± 0.21	83.72± 0.21	80.81± 0.31	77.45± 0.31	81.26± 0.33	84.11± 0.29*	**86.34± 0.30**
FRGC 30%	74.79± 0.25	76.25± 0.25	73.88± 0.38	69.62± 0.35	74.64± 0.36	76.92± 0.32*	**78.89± 0.35**

TABLE III

COMPARISON OF TRAINING TIME (IN SECONDS) ON DIFFERENT DATA SETS. (THE ASTERISKS INDICATE A STATISTICALLY SIGNIFICANT DIFFERENCE BETWEEN THE SECOND BEST PERFORMER AND THE PROPOSED METHOD AT A SIGNIFICANCE LEVEL OF 0.05)

Dataset	NCA	NCM	LMNN	BML	BNCA
Caltech-10	17.89± 0.11	16.64± 0.12	22.85± 0.12	16.02± 0.11*	**11.72± 0.11**
MNIST	16.60± 0.09	15.95± 0.09	17.95± 0.10	15.51± 0.08*	**10.43± 0.08**
FRGC-2	20.06± 0.12	19.33± 0.12	22.56± 0.14	18.71± 0.12*	**13.50± 0.12**

class) and the test set is another randomly sampled 300 images.

2) The data set of MNIST we used contains 600 digit images sampled from the full data set (60 from each class; training/test: 300/300).
3) The data set of FRGC we used contains 400 face images from 20 subjects (20 images per subject, training/test: 200/200).

On all these data sets, we inject random label noise on three levels (10%, 20%, and 30%) and the test sets are kept clean.

Fig. 2 shows some of the noisy data of Caltech-10. One can see that in each category, there exist some portion of images that are not belonging to this category, possibly due to the errors introduced in the labeling procedure, and very few work investigate the consequence of this.

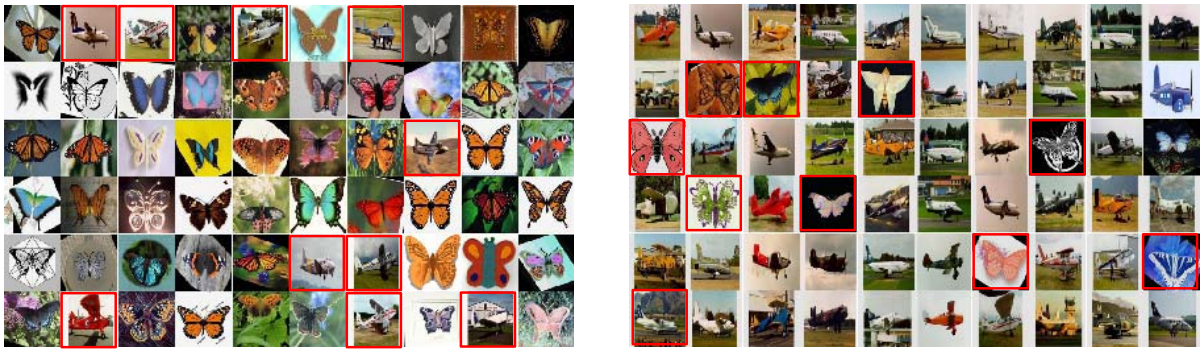
Table II shows how the ML algorithms perform under random label noise. Label noise could mislead ML algorithms in a way that it pulls data from different class together while keeps those from the same class away. When there is no label noise, almost all ML methods help to make an improvement in accuracy. However, it can be seen that the performance of all the methods declines with the increasing of noise level. Particularly, as the noise level increases to 30%, some of the ML methods do not work (such as NCA, NCM, and LMNN) in the sense that they even perform worse than the unsupervised baseline approaches (PCA and L1). Our BNCA works significantly better than traditional ML methods even under this challenging case—even when the noise level reaches 30%, the p-value is smaller than 0.001 when comparing our method with the second best performer in terms of accuracy.

Table III compares the corresponding running time of the methods in Table II under the situation of no label noise, with our (unoptimized MATLAB) implementation. Table II shows that on the average, the proposed BNCA runs more 61.2% faster than the NCA algorithm and more than 41.0% faster than the BML method. This is consistent with our analysis described in Section IV-C. That is, in each iteration of variational inference, the introduction of fixed curvature Bohning bound effectively avoids computing the Hessian matrix, resulting in significant reduction of running time.

D. Predictive Performance Under Difficult Conditions

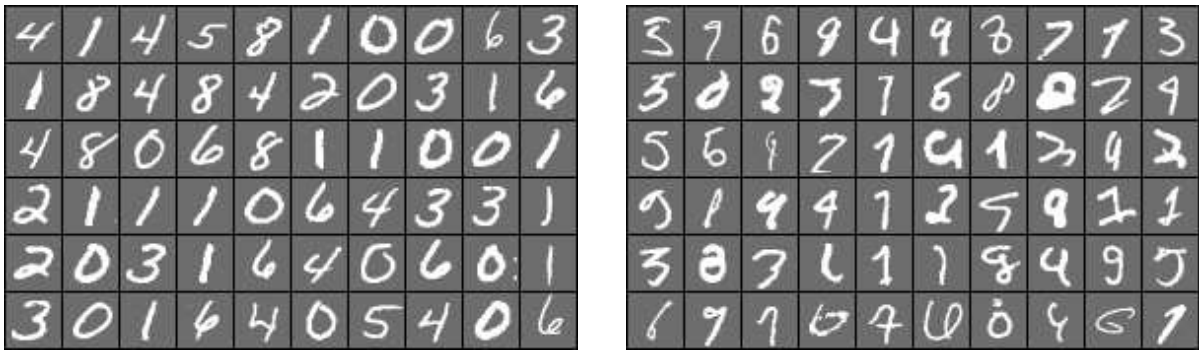
In Sections V-B and V-C, we have shown the benefits of the proposed BNCA method that learns either from a small number of training examples or from examples with label noise. In this section, we investigate empirically the robustness performance of the BNCA method under difficult conditions by comparing it with the baseline NCA method. The motivation for this is that since the difficult samples are commonly those lying either in the uncertain region, or those lying far away from the normal distribution, making prediction under these conditions would impose a great challenge for a traditional DML method based on point estimation, due to its lack of accounting for parameter uncertainty.

We conducted this series of experiments using MNIST [30]. First, 300 normal data points are sampled (by normal we mean that those digital images are not difficult for a human to recognize), and are used to train two models, i.e., an NCA and a BNCA model. For test, we collect two different test sets. One is normal while the other is most difficult in the sense that all digital images in this set are hard to recognize even by human. Since it is both time-consuming and error prone to select those difficult samples manually, we adopt one state-of-the-art model on the MNIST data set, i.e., the C-SVDDNet [33], as the expert to choose samples, and those samples close to the decision boundary of C-SVDDNet would be regarded as difficult samples, otherwise, as normal samples. In this way, we collect 300 random normal samples and



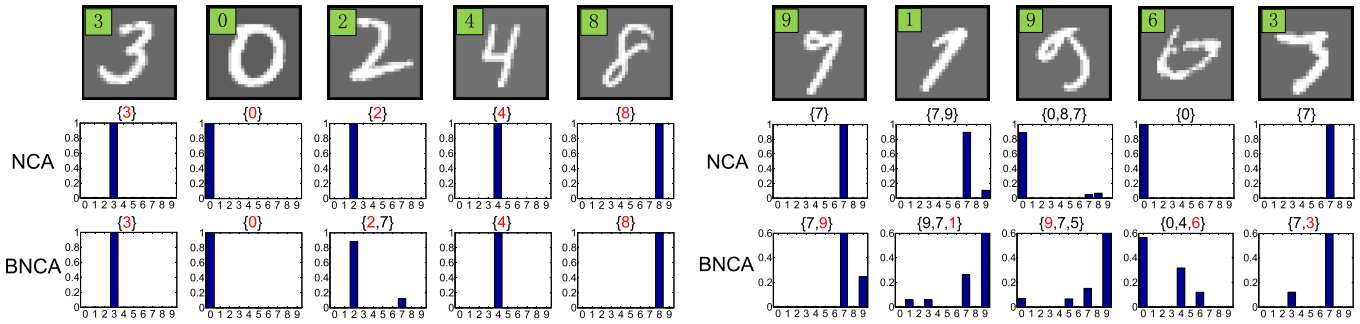
(a) (b)

Fig. 2. Illustration of training images injected with random label noise from the category of (a) plane and (b) butterfly in the Caltech-10 data set, where images with inaccurate labels are marked with a red square.



(a) (b)

Fig. 3. Illustration of the data from the MNIST data sets. (a) Normal data. (b) Difficult data.



(a) (b)

Fig. 4. Visualization of predictive probability: $P(y_i = k|x_i, Y_{N_i}, X_{N_i}, A)$ and $P(y_i = k|x_i, Y_{N_i}, X_{N_i})$. (a) Result on normal test data that the distribution of both NCA and BNCA has a single peak probability mass. (b) Result on difficult test data that NCA still has a single peak while BNCA assigns probability to several possible candidates. The digit in green square is the ground-truth label and the digits in braces are the ranking list of predictions.

300 most difficult samples, respectively, as test sets. There is no overlapping between these two test sets. Some of the samples are shown in Fig. 3. To evaluate the performance of NCA and BNCA in the two cases, we compute the predictive probability $P(y_i|x_i, Y_{N_i}, X_{N_i}, A)$ [using (8)] and $P(y_i|x_i, Y_{N_i}, X_{N_i})$ [using (29)] on the test data.

Fig. 4(a) visualizes the probability mass assigned to the normal test samples by the two models. We can see that both NCA and BNCA have a single peak probability mass, indicating that both of them are quite certain about their predictions. However, on the difficult set, their behaviors are largely different. Fig. 4(b) gives the results on this harder test set, and the ranking list of predictions according to their assigned probability.

It is obvious that the NCA is overconfident in its prediction. For example, the leftmost column of Fig. 4(b) shows that NCA incorrectly classifies the image of "9" as "7" with a high predictive probability of over 0.99 (those predictions with posterior mass less than 0.01 are canceled), indicating that this type of approximation to the posterior with a point mass is inadequate. On the other hand, the predictions of the BNCA are more moderate. One can see that although the true labels

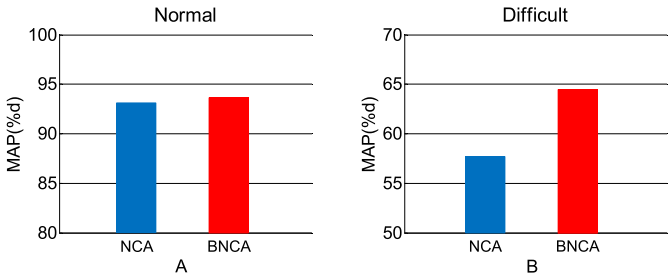


Fig. 5. Comparison of mean average precision (%) of NCA and BNCA on (a) normal test set and (b) difficult test set.

TABLE IV
COMPARATIVE PERFORMANCE (%) OF VARIOUS METHODS ON THE IMAGENET DATA SET (15.0% LABEL NOISE)

Algorithms	w.o label noise	w. label noise	time(seconds)
NCA	83.2	79.3	224
NCM	83.0	78.6	160
BML	82.6	79.7	155
LMNN	**84.6**	80.8	187
BNCA	83.0	**81.5**	**89**

may not be ranked the highest, they are correctly among the first few high ranking candidates. Under the previous example, although there is over 60% probability is assigned to the digital of "7" by our method, a significantly higher amount of predictive mass ($\geq$ 25%) than that of NCA is correctly assigned to the number of "9." This reveals that under difficult conditions, BNCA provides a much better approximation to the posterior than the point estimation method of NCA, by considering the uncertainty of parameters.

More precisely, we compare the performance of NCA and BNCA on the two test sets. For this, a new measurement, i.e., modified mean average precision (MAP), is introduced as our performance metric, which is defined as

$$\text{MAP} = \sum_{i=1}^{N}\sum_{k=1}^{K} p_i(k)\cdot 1\{y_i = k\}\cdot 1\{p(y_i = k|x_i) > \tau\} \quad (39)$$

where $p_i(k)$ is the precision at cutoff k in i's ranking list and τ is a truncating threshold. The threshold is usually set to be a very small number (e.g., 0.01), since if the corresponding response $p(y_i = k|x_i)$ of the model to the input x_i is too small, there is no point to count it in performance measuring.

Fig. 5 gives the results. One can see that while on the normal test data, the MAP accuracy of BNCA is slightly better than that of NCA (about 1.0% higher), the BNCA significantly outperforms NCA by more than 5.0% on the difficult set. This reveals that taking the prediction uncertainty into account indeed helps to improve the mean average precision. Also note that since the pairwise constrained BML method [24] only estimates whether a date pair belongs to the same class or not, while not being able to give the predictive distribution $p(y_i|x_i)$, it is not included for comparison here.

E. Experiments on Large Scale Data Set

To evaluate the performance of our method on large scale natural images, we conduct an experiment on ImageNet [34] data set. This data set contains over 1.2 million color images of totally 1000 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. Table IV gives the results. We can see that if the ground truth of label information is used for model training, our method perform comparable to NCA and LMNN. However, with 15% label noise, we see that the performance of NCA significantly reduces by 3.9%, while there is only 1.5% performance reduction in our BNCA. Furthermore, the training time is given in the third column of Table IV, and shows that our BNCA method is much faster than the compared methods. For example, it only takes less than half of the time of NCA for our model to train. This is consistent with our analysis that in each iteration of variational inference, the introduction of fixed curvature Bohning bound effectively avoids computing the Hessian matrix, resulting in significant reduction of time.

F. Discussions

1) Robustness Against Overfitting: To further investigate the behavior of the proposed BNCA method, we plot in Fig. 6 the learning curves of both BNCA and NCA as the function of the number of iterations. Three data sets (Caltech-10, MNIST, and FRGC) are used for this, with the same experimental setting as before, and on each data set, there are 30.0% random label noise injected. For NCA training, we used the conjugate gradient method [35], which seeks the steepest gradient direction with proper step size in each training step. Fig. 6 shows that with the iterations going, on all of the three data sets, the training errors of NCA keep decreasing but their test errors tend to rise at the same time, indicating that the method is easy to be overfitting under the condition of label noise. Although some empirical tricks, such as early stopping, can be adopted, Fig. 6 clearly shows that this is not an issue for our Bayesian extension to the NCA. Actually, Fig. 6 reveals that it only takes a few iterations before the learning converges.

In general, there are two types of noise, i.e., the data noise and the label noise. Although in this paper we focus on in the latter type of noise, i.e., noise caused by label error, it is interesting to further investigate the robustness of BNCA against data noise. Particularly, we, respectively, add speckle noise and Gaussian noise onto the training images of the Caltech-10 data set. For Gaussian noise, we vary the standard variance from 0σ to 0.1σ, where σ is the standard variance of data, while for speckle noise, the percent of noise features varies from 0% to 10%. We also evaluate the performance by varying the number of training images per class. Fig. 7 gives the results. One can see that in all the three cases of BNCA consistently outperform NCA, which validates the robustness of BNCA as analysis in Section IV-B. Particularly, Fig. 7(c) shows that with only six training images per class, our BNCA method improves the performance of NCA from 65.0% to about 74.0%, revealing that our method is less sensitive to the small sample problem than the NCA method.

2) Comparison of Various Training Methods: In this section, we investigate the effectiveness of our training

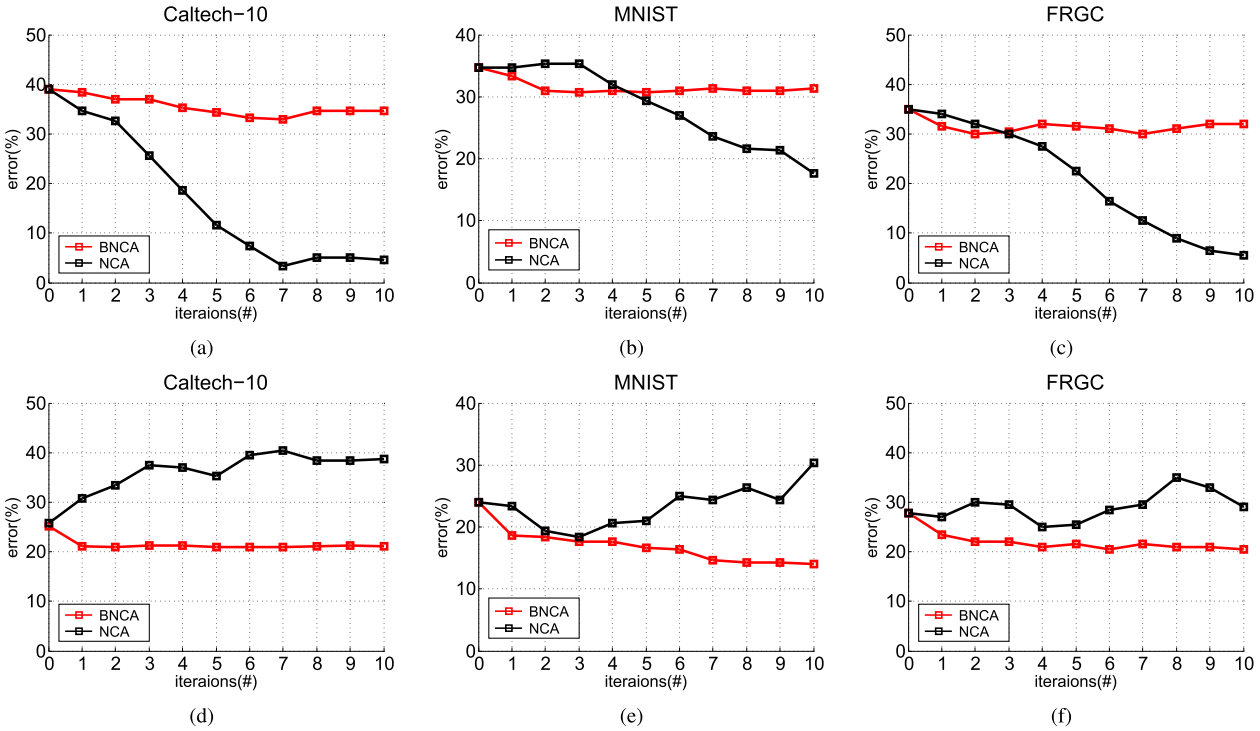


Fig. 6. Learning curves of NCA and BNCA. The classifier is KNN and the noise level is at 30%. (a)–(c) With the training set. (d)–(f) With the test set.

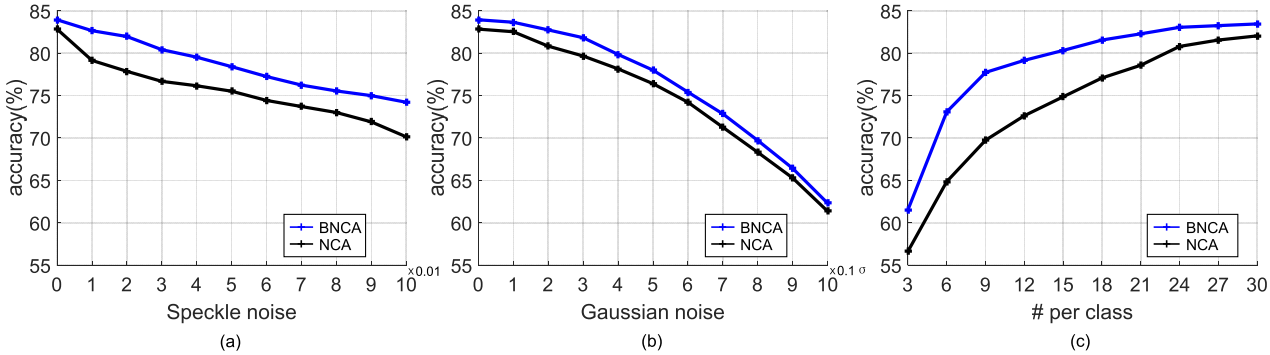


Fig. 7. Performance of BNCA and NCA under (a) speckle noise, (b) Gaussian noise, and (c) small sample size. For Gaussian noise, we vary the standard variance from 0σ to 0.1σ, where σ is the standard variance of data, while for speckle noise, the percent of noise features varies from 0% to 10%.

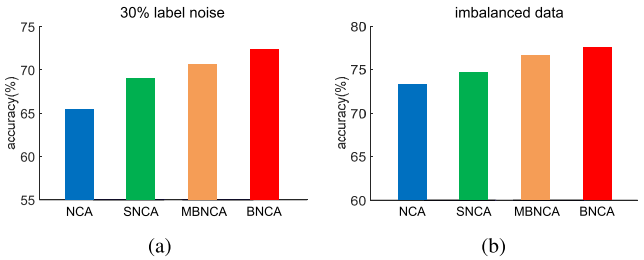


Fig. 8. Performance comparison among different variants of NCA methods (NCA, SNCA, MBNCA, and BNCA) on the Caltech-10 data set (a) with 30% label noise and (b) under imbalanced training set.

method by comparing it with two methods that solve numerically the nonapproximate functional instead of with closed form, i.e., spectral NCA (SNCA) and MCMC BNCA (MBNCA). The SNCA is a variant of NCA that uses the eigenapproximation to learn the model, i.e., replacing the distance metric parameter A in (1) to (3) with $A = \sum \gamma_i v_i v_i^T$, and performing the optimization using gradient ascent

$$\frac{\partial L}{\partial \gamma} = \sum_i \left(\sum_{j \in N_i} p_{ij} w_{ij} - \frac{\sum_{j \in N_i} y_{ij} p_{ij} w_{ij}}{\sum_{j \in N_i} y_{ij} p_{ij}} \right) \tag{40}$$

where w_{ij} is defined in (10).

The MBNCA is a sampling-based [36] approach to learn $\bar{\gamma}$

$$\begin{aligned} \bar{\gamma} &= \int_\gamma \gamma \cdot p(\gamma | S) d\gamma \\ &= \frac{1}{C} \int_\gamma \gamma \cdot p(S|\gamma) p(\gamma) d\gamma \\ &\approx \frac{1}{C} \sum_{t=1}^{T} \gamma_t \cdot p(S|\gamma_t) \end{aligned} \tag{41}$$

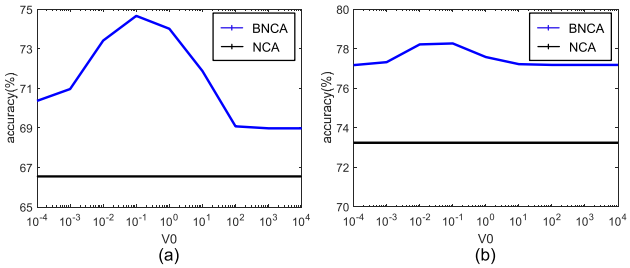


Fig. 9. Influence of prior with 30% label noise on the Caltech-10 data set. (a) Training size = 300. (b) Training size = 600.

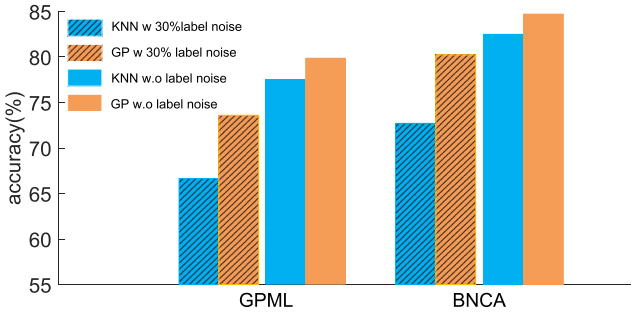


Fig. 10. Comparison of GPML and BNCA on the Caltech-10 data set.

where γ_t is sampled from prior distribution $p(\gamma)$, and the normalizing const C is estimated as follows:

$$C = \int_{\gamma} p(S|\gamma)p(\gamma)d\gamma \approx \sum_{t=1}^{T} p(S|\gamma_t). \tag{42}$$

To test these methods, we conducted a series of experiments on the Caltech-10 data set with two settings: 1) with 30% label noise and 2) under imbalanced training data. For the latter case, we randomly sampled different numbers (e.g., 3, 6,…30) of images from each of the ten class as training set; hence, each time in training the size of each class is different.

Fig. 8 gives the results. One can see that under both experimental settings, compared with the baseline algorithm (NCA), both approximation methods based on numerical techniques (e.g., SNCA and MBNCA) and our closed form solver (BNCA) achieve better performance, and our BNCA method performs the best. For example, on the imbalanced data set, the accuracy of NCA is 73.5% and the SNCA slightly improves this to 74.8%, while our BNCA significantly outperforms both methods, achieving an accuracy of 77.5%. Fig. 8 also shows that the BNCA method consistently performs better than the sampling-based MBNCA method, revealing the effectiveness of our closed form solution.

3) Effect of Regularization: To investigate the effect of relative importance of two major components of the proposed method, i.e., prior and local variational inference with spectral decomposition, we conducted a series of experiments on the Caltech-10 data set with 30% random label noise by changing the degree of prior knowledge used in the model. Specifically, we vary the value of σ^2 ($V_0 = \sigma^2 I$) from 10^{-4} to 10^4 but keeping the mean value m_0 fixed at the same time. Note that a large value of σ^2 indicates that the prior tends to be more noninformative (i.e., higher uncertain) about the γ value. Fig. 9 shows how the performance changes as a function of the degree of uncertainty in prior. We have three observations from Fig. 9: 1) the prior is beneficial when the value of σ^2 in a relatively large range between 10^{-3} and 10^1; 2) even when the prior is very flat (e.g., σ^2 is larger than 10^1), our BNCA method still works better than the baseline NCA method, which indicates that the robustness capability of our approach; and 3) as expected, with increasing amount of data, the influence of the prior distribution on posterior quantities becomes weaker.

4) Comparison of BNCA and Gaussian Process: Besides BML, Gaussian process (GP) is also a widely used approach to estimate the uncertainty of prediction. There are some connections between them. In [37], a GP for ML (GPML) method is proposed. The objective is to maximize the marginal distribution $p(Y|A)$, which is shown to be

$$\log p(Y|A) = -\frac{1}{2}Y^T K^{-1} Y - \frac{1}{2}\log|K| + \text{const} \tag{43}$$

where $Y = [y_1, y_2, \ldots y_N]^T$ ($y_i \in \{-1, 1\}$) is the class labels of training data. The covariance (kernel) matrix K is chosen to be $K_{ij} = 1/(x_i - x_j)^T A(x_i - x_j)$. Let $A = MM^T$; then, $-\log|K|$ can be regarded as an L2-norm regularizer on M. Note that $Y^T K^{-1} Y = \sum_i \sum_j y_{ij} K_{ij}^{-1}$, where $y_{ij} = 1\{y_i = y_j\}$. Hence, GP can also be regarded as a pairwise constrained ML method (similar argument is also valid for logistic regression, as in [1]).

However, despite such connection in the context of DML, there are two main differences between GP and BML: 1) in GPML, the parameters are learned via point estimation (evidence maximization), which is not robust against label noise and 2) GP is a nonparametric method and has higher computational cost than ours, due to the need of computing the inverse of the covariance matrix.

Due to the above-mentioned reasons, our method is superior to GP regarding both robustness and efficiency. Note that one can think of the DML as a kind of feature extraction and, hence, can embed it into GP through the covariance matrix K for classification. We conduct an experiment to compare the performance of GPML and BNCA on Caltech-10 data set using two classifiers, i.e., KNN and GP, respectively. For KNN, both GPML and BNCA are used as the similarity measure, while for GP, they are just two different ways to calculate the K matrix. Fig. 10 gives the results. It shows that our BNCA consistently outperforms GPML no matter with or without label noise. Fig. 10 also shows that as a nonlinear classifier, GP leads to higher performance than KNN.

VI. Conclusion

We present a new BML method—BNCA that effectively improves the performance of KNN classifier under the condition of small sample size and/or when data labels are noisy. The method is based on the classical NCA method with point estimation, and for the first time extends it under the Bayesian framework. The major advantages of BNCA over NCA in DML are threefolds: 1) it is easy to train without worrying about overfitting; 2) it performs more robust compared with

NCA under difficult conditions; and 3) it naturally handles label noise by reducing the influence of data points with possible labeling errors. In addition, to improve the efficiency of Bayesian learning, we introduce a new variational lower bound of the log-likelihood of the objective. Extensive experiments conducted on several challenging real-world applications show that the performance of the proposed BNCA method significantly improves upon the baseline NCA method and it outperforms several other state-of-the-art DML methods as well. We are currently investigating more applications of the proposed BNCA method, such as image retrieval with model uncertainty.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions.

REFERENCES

[1] 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.

[2] S. C. H. Hoi, W. Liu, M. R. Lyu, and W.-Y. Ma, "Learning distance metrics with contextual constraints for image retrieval," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2. Jun. 2006, pp. 2072–2078.

[3] C. Li, Q. Liu, J. Liu, and H. Lu, "Ordinal distance metric learning for image ranking," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 7, pp. 1551–1559, Jul. 2015.

[4] M. Guillaumin, J. Verbeek, and C. Schmid, "Multiple instance metric learning from automatically labeled bags of faces," in *Proc. ECCV*, 2010, pp. 634–647.

[5] J. Lu, X. Zhou, Y.-P. Tan, Y. Shang, and J. Zhou, "Neighborhood repulsed metric learning for kinship verification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 2, pp. 331–345, Feb. 2014.

[6] J. Ye, Z. Zhao, and H. Liu, "Adaptive distance metric learning for clustering," in *Proc. CVPR*, 2007, pp. 1–7.

[7] M. Dikmen, E. Akbas, T. S. Huang, and N. Ahuja, "Pedestrian recognition with a learned metric," in *Proc. ACCV*, 2011, pp. 501–512.

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

[9] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid, "Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation," in *Proc. ICCV*, 2009, pp. 309–316.

[10] M. Koestinger, M. Hirzer, P. Wohlhart, P. M. Roth, and H. Bischof, "Large scale metric learning from equivalence constraints," in *Proc. CVPR*, 2012, pp. 2288–2295.

[11] W. Bian and D. Tao, "Constrained empirical risk minimization framework for distance metric learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 8, pp. 1194–1205, Aug. 2012.

[12] S. Fouad, P. Tino, S. Raychaudhury, and P. Schneider, "Incorporating privileged information through metric learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 7, pp. 1086–1098, Jul. 2013.

[13] C. Shen, J. Kim, F. Liu, L. Wang, and A. van den Hengel, "Efficient dual approach to distance metric learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 2, pp. 394–406, Feb. 2014.

[14] J. Li, X. Lin, X. Rui, Y. Rui, and D. Tao, "A distributed approach toward discriminative distance metric learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 9, pp. 2111–2122, Sep. 2015.

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

[16] 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.

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

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

[19] J. Van Hulse, T. M. Khoshgoftaar, and A. Napolitano, "A novel noise filtering algorithm for imbalanced data," in *Proc. ICMLA*, 2010, pp. 9–14.

[20] T. Leung, Y. Song, and J. Zhang, "Handling label noise in video classification via multiple instance learning," in *Proc. ICCV*, 2011, pp. 2056–2063.

[21] S. Fefilatyev *et al.*, "Label-noise reduction with support vector machines," in *Proc. ICPR*, 2012, pp. 3504–3508.

[22] D. Wang and X. Tan, "Label-denoising auto-encoder for classification with inaccurate supervision information," in *Proc. ICPR*, 2014, pp. 3648–3653.

[23] 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.

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

[25] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA, USA: MIT Press, 2012.

[26] C. Andrieu, N. De Freitas, A. Doucet, and M. I. Jordan, "An introduction to MCMC for machine learning," *Mach. Learn.*, vol. 50, nos. 1–2, pp. 5–43, 2003.

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

[28] D. J. C. Mackay, "A practical Bayesian framework for backprop networks," *Neural Comput.*, vol. 4, no. 3, pp. 448–472, 1992.

[29] 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.

[30] 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.

[31] P. J. Phillips *et al.*, "Overview of the face recognition grand challenge," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1. Jun. 2005, pp. 947–954.

[32] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," California Inst. Technol., Tech. Rep. 7694, 2007.

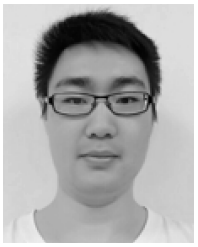
[33] D. Wang and X. Tan, "Unsupervised feature learning with c-svddnet," *Pattern Recognit.*, vol. 60, pp. 473–485, Dec. 2016.

[34] 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.

[35] J. R. Shewchuk, "An introduction to the conjugate gradient method without the agonizing pain," Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. CMU-CS-94-125, 1994.

[36] M. Isard and A. Blake, "Condensation—Conditional density propagation for visual tracking," *Int. J. Comput. Vis.*, vol. 29, no. 1, pp. 5–28, 1998.

[37] N. A. Zaidi and D. M. Squire, "Data dependent distance metric for efficient Gaussian processes classification," Tech. Rep. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.719.8106



Dong Wang is currently pursuing the Ph.D. degree with the Nanjing University of Aeronautics and Astronautics, Nanjing, China.

His current 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 current research interests include deep learning, reinforcement learning, and Bayesian learning.

Dr. Tan received the 2015 IEEE Signal Processing Society Best Paper Award.