Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

A new constrained maximum margin approach to discriminative learning of Bayesian classifiers*

Ke GUO[1], Xia-bi LIU[1], Lun-hao GUO[1], Zong-jie LI[1], Zeng-min GENG[‡2]

[1]*Beijing Laboratory of Intelligent Information Technology, School of Computer Science,*
Beijing Institute of Technology, Beijing 100081, China

[2]*Computer Information Center, Beijing Institute of Fashion Technology, Beijing 100029, China*

E-mail: guoke@bit.edu.cn; liuxiabi@bit.edu.cn; guolunhao@bit.edu.cn; leezongjie@163.com; jsjgzm@bift.edu.cn

Received Jan. 4, 2017; Revision accepted Aug. 21, 2017; Crosschecked May 10, 2018

Abstract: We propose a novel discriminative learning approach for Bayesian pattern classification, called 'constrained maximum margin (CMM)'. We define the margin between two classes as the difference between the minimum decision value for positive samples and the maximum decision value for negative samples. The learning problem is to maximize the margin under the constraint that each training pattern is classified correctly. This nonlinear programming problem is solved using the sequential unconstrained minimization technique. We applied the proposed CMM approach to learn Bayesian classifiers based on Gaussian mixture models, and conducted the experiments on 10 UCI datasets. The performance of our approach was compared with those of the expectation-maximization algorithm, the support vector machine, and other state-of-the-art approaches. The experimental results demonstrated the effectiveness of our approach.

Key words: Discriminative learning; Statistical modeling; Bayesian pattern classifiers; Gaussian mixture models; UCI datasets
https://doi.org/10.1631/FITEE.1700007 **CLC number:** TP391

1 Introduction

It is crucial for Bayesian classifiers to learn representative class information from samples of different classes. Approaches to this problem can be divided into two categories: generative learning and discriminative learning. They are differentiated by their criteria for evaluation of learning results. In generative learning algorithms, such as classical maximum likelihood (ML) based algorithms, the first concern is how well the class membership probabilities of the training data are estimated. The Bayesian classification theory tells us that the optimal discrimination between classes can be achieved indirectly in this way if the class distribution model of the training set is perfectly consistent with the real distribution of data. However, the training data is often insufficient or contains noise; therefore, the class distribution estimated by generative learning algorithms often deviates from real ones, leading to unsatisfactory classifiers. To solve this problem, discriminative learning algorithms are introduced to consider the discrimination between classes directly in the training phase. They focus on the difference between classes instead of the distribution of the data within each class.

The main discriminative learning criteria for the optimization of Bayesian classifiers include conditional maximum likelihood (CML) (Nádas, 1983), maximum mutual information (MMI) (Povey and Woodland, 2002), minimum classification error (MCE) (Juang and Katagiri, 1992), and maximum margin (MM) (Pernkopf and Wohlmayr, 2010; Pernkopf et al., 2012). The common ground shared by

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (Nos. 60973059 and 81171407) and the Program for New Century Excellent Talents in University, China (No. NCET-10-0044)

ORCID: Ke GUO, http://orcid.org/0000-0002-9278-4046

these criteria is that the difference between classes is expressed based on the joint probability distribution of the class and training data. Under some minor conditions, the MMI criterion is in fact identical to the CML criterion (Jiang, 2010). Both of them try to maximize the difference between the joint probability distribution of the class in the training set and the sum of all the joint probabilities. Alternatively, the MCE and MM criteria consider the difference between the joint probability of each training instance and its true class label and the joint probabilities of this data and the competitive class labels. In the MM based method, the difference is used to measure the margin between classes. The minimum margin over the training data is maximized as far as possible. In the MCE based method, the difference is considered to be the misclassification measure. The attempt is made to minimize the smoothed empirical classification error rate over the training set. The MCE criterion is modified by introducing sub-string errors to obtain the minimum phone error (MPE) and minimum word error (MWE) criteria for large vocabulary continuous speech recognition (Povey and Woodland, 2002). The classifier optimization problems based on the criteria above can be solved using the gradient descent algorithm (Juang and Katagiri, 1992), conjugate gradient algorithm (Pernkopf et al., 2012), extended Baum-Welch (EBW) algorithm (Woodland and Povey, 2002; Pernkopf and Wohlmayr, 2010), evolutionary strategy (Dong and Zhou, 2014), and heuristic search algorithm (Karabatak, 2015), among others.

In this paper, we propose a novel approach to the discriminative learning of Bayesian pattern classifiers. In the proposed approach, we define the optimal binary classifier as the one that maximizes the separation between two classes under the constraints of correct classification of training data. This approach is called 'constrained maximum margin (CMM)'. As mentioned above, the maximum separation between two classes for each sample is pursued by the previous discriminative learning approaches for Bayesian classification. Conversely, our learning criterion focuses on the separation between all the samples of a class and those of another class. Furthermore, in our approach, the constraints of correct classification of training data are considered, which are somewhat ignored in previous studies. Based on the defined learning criteria, we employ the sequential unconstrained minimization technique (Fiacco and Mc-Cormick, 1990) to find the optimal binary classifiers. Finally, multi-class classification was performed using the 'Max Wins' voting strategy on the classification results from all the binary classifiers. We applied the proposed CMM approach to learn Bayesian classifiers based on Gaussian mixture models (GMMs). The corresponding classification experiments were conducted on 10 datasets from the well-known UCI Machine Learning Repository (University of California, 2013).

The performance of our approach was compared with that of classical generative and discriminative counterparts: expectation-maximization (EM) algorithm (Dempster et al., 1977) and support vector machines (SVMs) (Vapnik, 2013). We also compared the performance of our approach with the best previous results to the best of our knowledge (Gorman and Sejnowski, 1988; Bredensteiner and Bennett, 1999; Kwok, 1999; Jiang and Zhou, 2004; Webb et al., 2005; Dvořák and Savický, 2007; Jiang et al., 2009, 2012). Additionally, we ran the non-parametric Wilcoxon signed-rank test to compare our approach with EM, SVM, and some other state-of-the-art Bayesian classifiers (Friedman et al., 1997; Webb et al., 2005; Jiang et al., 2009, 2012). The experimental results showed that the proposed CMM approach to discriminative learning of Bayesian classifiers is effective.

2 The proposed approach

2.1 Learning objective

Given a feature vector $\boldsymbol{x}$ and a finite set of classes $\{\omega_1, \omega_2, \ldots, \omega_n\}$, let $P(\omega_i)$, $p(\boldsymbol{x}|\omega_i)$, and $P(\omega_i|\boldsymbol{x})$ be the prior probability, class-conditional probability density function, and posterior probability, respectively. The Bayesian classification rule for minimization of the probability of the classification error is to classify $\boldsymbol{x}$ into class ω^* with the maximum posterior probability:

$$\omega^* = \arg\max_{\omega_i} P\left(\omega_i \mid \boldsymbol{x}\right). \tag{1}$$

Based on the Bayes formula

$$P(\omega_i \mid \boldsymbol{x}) = \frac{p(\boldsymbol{x} \mid \omega_i) P(\omega_i)}{p(\boldsymbol{x})}, \tag{2}$$

we have

$$\frac{P(\omega_i\,|\,\boldsymbol{x})}{P(\omega_j\,|\,\boldsymbol{x})}=\frac{p(\boldsymbol{x}\,|\,\omega_i)P(\omega_i)}{p(\boldsymbol{x}\,|\,\omega_j)P(\omega_j)}. \tag{3}$$

Thus, the completely equivalent and actually used decision rule is given by

$$\omega^*=\arg\max_{\omega_i} p(\boldsymbol{x}\,|\,\omega_i)P(\omega_i). \tag{4}$$

Consider a binary classification problem. Let $\hat{\boldsymbol{x}}$ be the feature vector of an arbitrary positive sample of a class, e.g., class ω_1, and $\bar{\boldsymbol{x}}$ be the feature vector of an arbitrary negative sample of the class. $\hat{\boldsymbol{x}}$ is classified correctly if and only if

$$\frac{p(\hat{\boldsymbol{x}}\,|\,\omega_1)P(\omega_1)}{p(\hat{\boldsymbol{x}}\,|\,\omega_2)P(\omega_2)}>1. \tag{5}$$

Let $P_x=\frac{P(\omega_1)}{P(\omega_2)}$. Then constraint (5) is simplified as

$$P_x\frac{p(\hat{\boldsymbol{x}}\,|\,\omega_1)}{p(\hat{\boldsymbol{x}}\,|\,\omega_2)}>1, \tag{6}$$

or equivalently,

$$\log P_x\frac{p(\hat{\boldsymbol{x}}\,|\,\omega_1)}{p(\hat{\boldsymbol{x}}\,|\,\omega_2)}>0. \tag{7}$$

Similarly, we have

$$\log P_x\frac{p(\bar{\boldsymbol{x}}\,|\,\omega_1)}{p(\bar{\boldsymbol{x}}\,|\,\omega_2)}<0 \tag{8}$$

for $\bar{\boldsymbol{x}}$.

According to constraints (7) and (8), we define the margin between two classes as

$$\begin{aligned}\gamma&=\min\log P_x\frac{p(\hat{\boldsymbol{x}}\,|\,\omega_1)}{p(\hat{\boldsymbol{x}}\,|\,\omega_2)}-\max\log P_x\frac{p(\bar{\boldsymbol{x}}\,|\,\omega_1)}{p(\bar{\boldsymbol{x}}\,|\,\omega_2)}\\&=\min\log\frac{p(\hat{\boldsymbol{x}}\,|\,\omega_1)}{p(\hat{\boldsymbol{x}}\,|\,\omega_2)}-\max\log\frac{p(\bar{\boldsymbol{x}}\,|\,\omega_1)}{p(\bar{\boldsymbol{x}}\,|\,\omega_2)}.\end{aligned} \tag{9}$$

We replace the minimum and maximum operators by the differentiable softmin (Kim and Pfister, 2011) and softmax (Pernkopf et al., 2012) functions, respectively. This means that

$$\begin{aligned}\min\log\frac{p(\hat{\boldsymbol{x}}\,|\,\omega_1)}{p(\hat{\boldsymbol{x}}\,|\,\omega_2)}&\approx-\frac{1}{h}\log\sum_{\hat{x}}\exp\left(-h\log\frac{p(\hat{\boldsymbol{x}}\,|\,\omega_1)}{p(\hat{\boldsymbol{x}}\,|\,\omega_2)}\right)\\&=-\frac{1}{h}\log\sum_{\hat{x}}\frac{p^h(\hat{\boldsymbol{x}}\,|\,\omega_2)}{p^h(\hat{\boldsymbol{x}}\,|\,\omega_1)},\end{aligned} \tag{10}$$

$$\begin{aligned}\max\log\frac{p(\bar{\boldsymbol{x}}\,|\,\omega_1)}{p(\bar{\boldsymbol{x}}\,|\,\omega_2)}&\approx\frac{1}{h}\log\sum_{\bar{x}}\exp\left(h\log\frac{p(\bar{\boldsymbol{x}}\,|\,\omega_1)}{p(\bar{\boldsymbol{x}}\,|\,\omega_2)}\right)\\&=\frac{1}{h}\log\sum_{\bar{x}}\frac{p^h(\bar{\boldsymbol{x}}\,|\,\omega_1)}{p^h(\bar{\boldsymbol{x}}\,|\,\omega_2)},\end{aligned} \tag{11}$$

where h≥1. The approximation replacement by the soft function can be considered as using the limiting case of a general expression. Then the approximate margin is

$$\gamma'=-\frac{1}{h}\log\sum_{\hat{x}}\frac{p^h(\hat{\boldsymbol{x}}\,|\,\omega_2)}{p^h(\hat{\boldsymbol{x}}\,|\,\omega_1)}-\frac{1}{h}\log\sum_{\bar{x}}\frac{p^h(\bar{\boldsymbol{x}}\,|\,\omega_1)}{p^h(\bar{\boldsymbol{x}}\,|\,\omega_2)}. \tag{12}$$

The margin definition is crucial due to the participation of all samples from both the positive and negative classes. Moreover, the sum for each class allows the margin to be defined on the class level, which is different from those defined on the sample level. The margin takes all the samples of a class as a whole and the separation would further be maximized on the class level. The learning objective is to maximize this approximate margin, i.e., max(γ'), which can be transformed into a minimization problem, min($-\gamma'$). Furthermore, constraints (7) and (8) should be satisfied for separable cases.

Let $\boldsymbol{x}_i$ be the i^{th} sample from the feature set, which can be either a positive or a negative sample. By introducing

$$y_i=\begin{cases}1, & \text{if } \boldsymbol{x}_i \text{ is a positive sample,}\\-1, & \text{if } \boldsymbol{x}_i \text{ is a negative sample,}\end{cases} \tag{13}$$

the two constraints are unified as

$$y_i\left\lfloor\log P_x+\log p(\boldsymbol{x}_i\,|\,\omega_1)-\log p(\boldsymbol{x}_i\,|\,\omega_2)\right\rfloor>0. \tag{14}$$

During the experiments, we found that learning can be improved if a bound larger than 0 is exerted on the right-hand side of constraint (14). The explanation for this improvement is that this strategy leads to a larger separation between the two classes. Additionally, P_x should satisfy P_x>0, which is transformed to an unconstrained domain by

$$P_x = \exp(\tilde{P}_x). \tag{15}$$

According to the two considerations above, we modify constraint (14) as follows:

$$y_i \left\lfloor \tilde{P}_x + \log p(\boldsymbol{x}_i \mid \omega_1) - \log p(\boldsymbol{x}_i \mid \omega_2) \right\rfloor > \xi, \tag{16}$$

where ξ is a variable larger than 0.

In the following descriptions, we let $g_i(\Lambda)= y_i\left\lfloor \tilde{P}_x + \log p(\boldsymbol{x}_i| \omega_1) - \log p(\boldsymbol{x}_i| \omega_2) \right\rfloor$ and $f(\Lambda)=-\gamma'= \frac{1}{h}\log\sum_{\hat{\boldsymbol{x}}} \frac{p^h(\hat{\boldsymbol{x}} \mid \omega_2)}{p^h(\hat{\boldsymbol{x}} \mid \omega_1)} + \frac{1}{h}\log\sum_{\bar{\boldsymbol{x}}} \frac{p^h(\bar{\boldsymbol{x}} \mid \omega_1)}{p^h(\bar{\boldsymbol{x}} \mid \omega_2)}$, where Λ denotes unknown parameters, including P_x and those in two class-conditional probability density functions. Therefore, the final learning problem to be solved is

$$\begin{aligned} &\min f(\Lambda) \\ &\text{s.t. } g_i(\Lambda) > \xi,\ i = 1,2,\ldots,N, \end{aligned} \tag{17}$$

where N denotes the number of all the samples.

2.2 Sequential unconstrained minimization

The exterior point minimization method, a kind of sequential unconstrained minimization technique (SUMT) (Fiacco and McCormick, 1990), is applied to solve the nonlinear programming problem (17). In accordance with the exterior point minimization method, we define function

$$F(\Lambda) = f(\Lambda) + M\left[\sum_{i\in H}\left(g_i(\Lambda)\right)^2\right], \tag{18}$$

where M is a penalty factor larger than 0 and H is the set of subscripts that correspond with those samples that violate the constraints. This can be formulated as

$$H = \{ i \mid g_i(\Lambda) \le \xi \}, \tag{19}$$

where ξ has the same definition as described in constraint (16).

Any unconstrained minimization method can be used to minimize $F(\Lambda)$ for the specified value of M. The procedure is then repeated with an increased value of M. We detail this kind of exterior point minimization algorithm in Algorithm 1.

Algorithm 1 Exterior point minimization algorithm for classifier optimization

```
Input: training dataset, initial parameter $\Lambda$, and initial penalty
    factor $M$ and its magnification coefficient $\beta$.
Output: the optimal parameters.
  // Optimization:
  1 Use an unconstrained minimization method to solve the
        unconstrained minimization problem expressed in
        function (18);
  2 If $\sum_{i\in H}(g_i(\Lambda))^2 < \varepsilon$ ($\varepsilon$ is an infinitesimal) or the number of
        iterations exceeds the preset maximum value, we stop
        the computation and take the output of current
        parameters as optimal; else, let $M=\beta M$, and then go to
        step 1.
```

For the unconstrained minimization method required in step 1 of Algorithm 1, we used the gradient descent method. The following iterative equation was used to update the parameters in step 1:

$$\Lambda_{t+1} = \Lambda_t - \lambda_t \nabla F(\Lambda_t), \tag{20}$$

where Λ_t and λ_t are the parameter set and the step size in the t^{th} iteration of the gradient descent algorithm, respectively, and $\nabla F(\Lambda_t)$ is the partial derivative of $F(\Lambda)$ with respect to all the parameters in Λ_t. Let Ψ denote an arbitrary parameter in Λ, which can be a parameter in $p(\boldsymbol{x}|\omega_1)$ (denoted by Ψ_1), a parameter in $p(\boldsymbol{x}|\omega_2)$ (denoted by Ψ_2), or $\tilde{P}_x$. Then we have

$$\frac{\partial F(\Lambda)}{\partial \Psi} = \frac{\partial f(\Lambda)}{\partial \Psi} + M\sum_{i\in H} 2g_i(\Lambda)\frac{\partial g_i(\Lambda)}{\partial \Psi}, \tag{21}$$

where

$$\begin{aligned} \frac{\partial f(\Lambda)}{\partial \Psi_1} = &\left[\sum_{\bar{\boldsymbol{x}}} \frac{p^h(\bar{\boldsymbol{x}} \mid \omega_1)}{p^h(\bar{\boldsymbol{x}} \mid \omega_2)}\right]^{-1} \sum_{\bar{\boldsymbol{x}}} \frac{p^{h-1}(\bar{\boldsymbol{x}} \mid \omega_1)}{p^h(\bar{\boldsymbol{x}} \mid \omega_2)} \frac{\partial p(\bar{\boldsymbol{x}} \mid \omega_1)}{\partial \Psi_1} \\ &- \left[\sum_{\hat{\boldsymbol{x}}} \frac{p^h(\hat{\boldsymbol{x}} \mid \omega_2)}{p^h(\hat{\boldsymbol{x}} \mid \omega_1)}\right]^{-1} \sum_{\hat{\boldsymbol{x}}} \frac{p^h(\hat{\boldsymbol{x}} \mid \omega_2)}{p^{h+1}(\hat{\boldsymbol{x}} \mid \omega_1)} \frac{\partial p(\hat{\boldsymbol{x}} \mid \omega_1)}{\partial \Psi_1}, \end{aligned} \tag{22}$$

$$\frac{\partial f(\Lambda)}{\partial \Psi_2}=\left[\sum_{\hat{x}}\frac{p^h(\hat{\boldsymbol{x}}\,|\,\omega_2)}{p^h(\hat{\boldsymbol{x}}\,|\,\omega_1)}\right]^{-1}\sum_{\hat{x}}\frac{p^{h-1}(\hat{\boldsymbol{x}}\,|\,\omega_2)}{p^h(\hat{\boldsymbol{x}}\,|\,\omega_1)}\frac{\partial p(\hat{\boldsymbol{x}}\,|\,\omega_2)}{\partial \Psi_2}$$
$$-\left[\sum_{\bar{x}}\frac{p^h(\bar{\boldsymbol{x}}\,|\,\omega_1)}{p^h(\bar{\boldsymbol{x}}\,|\,\omega_2)}\right]^{-1}\sum_{\bar{x}}\frac{p^{h}(\bar{\boldsymbol{x}}\,|\,\omega_1)}{p^{h+1}(\bar{\boldsymbol{x}}\,|\,\omega_2)}\frac{\partial p(\bar{\boldsymbol{x}}\,|\,\omega_2)}{\partial \Psi_2}, \tag{23}$$

$$\frac{\partial g_i(\Lambda)}{\partial \Psi_1}=y_i\frac{1}{p(\boldsymbol{x}_i\,|\,\omega_1)}\frac{\partial p(\boldsymbol{x}_i\,|\,\omega_1)}{\partial \Psi_1}, \tag{24}$$

$$\frac{\partial g_i(\Lambda)}{\partial \Psi_2}=-y_i\frac{1}{p(\boldsymbol{x}_i\,|\,\omega_2)}\frac{\partial p(\boldsymbol{x}_i\,|\,\omega_2)}{\partial \Psi_2}, \tag{25}$$

$$\frac{\partial g_i(\Lambda)}{\partial \tilde{P}_x}=y_i. \tag{26}$$

In Eqs. (22)–(25), $\frac{\partial p(\boldsymbol{x}\,|\,\omega_i)}{\partial \Psi}$ depends on $p(\boldsymbol{x}|\omega_i)$ and Ψ; thus, they have to be decided in the applications. Based on Eqs. (21)–(26), the gradient descent algorithm used in step 1 of Algorithm 1 is given in Algorithm 2.

Algorithm 2 Gradient descent algorithm used in step 1 of Algorithm 1

```
Input: training dataset; initial parameters.
Output: updated parameters.
  // Optimization:
  repeat
    Compute the partial derivative of $F(\Lambda)$ with respect to
      each parameter using Eq. (21);
    Compute step size $\lambda_t$ using the improved 0.618 method
      (Forsythe et al., 1977);
    Update the parameters using Eq. (20);
  until convergence or the preset maximum number of
    iterations is reached. Let $\varepsilon'$ be an infinitesimal. The
    convergence condition is
```

$$\|\boldsymbol{g}_t\|=\left(\sum\left(\frac{\partial F(\Lambda)}{\partial \Psi}\right)^2\right)^{1/2}\le\varepsilon'.$$

2.3 Multi-class classification

Based on the binary classifier learning method described above, the multi-class classification problem can be solved by one of two strategies: one-against-one (1v1) and one-against-others (1vO). In the 1v1 strategy, we establish a binary classifier for each pair of classes. For an input pattern, the recognition result is determined by 'Max Wins' voting, in which all binary classifiers vote for each class and the winning class is the one with the maximum votes. In the 1vO strategy, we train a classifier for each class to discriminate between the samples of this class and those of all the other classes. An arbitrary input pattern is then classified based on its correspondence to the maximum output of all classifiers. We compared these two strategies in experiments. The average results demonstrated the superiority of the 1v1 strategy over 1vO.

3 Configuring the proposed approach for Gaussian mixture modeling based Bayesian classifiers

Before the proposed approach to Bayesian classification is ready to use, two parts need to be adapted to the situation at hand. The first one is the modeling of the class-conditional probability density function, and the second is the method for determining the initial parameters of the classifiers.

3.1 Gaussian mixture modeling

In this study, the class-conditional probability density function is approximated using GMM, which is a general model for estimating an unknown probability density function. Under regular conditions, it may approximate any continuous functions having a finite number of discontinuities (Vlassis and Likas, 1999). Let K be the number of Gaussian components in GMM, $\boldsymbol{w}_k$, $\boldsymbol{\mu}_k$, and $\boldsymbol{\Sigma}_k$ be the weight, mean, and covariance matrices of the k^{th} Gaussian component, respectively, $\sum_{k=1}^{K}\boldsymbol{w}_k=1$. Then we have

$$p(\boldsymbol{x}\,|\,\omega_i)=\sum_{k=1}^{K}\boldsymbol{w}_k N(\boldsymbol{x}\,|\,\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k), \tag{27}$$

where

$$N(\boldsymbol{x}\,|\,\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)$$
$$=(2\pi)^{-\frac{d}{2}}\left|\boldsymbol{\Sigma}_k\right|^{-\frac{1}{2}}\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^{\mathrm{T}}\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k)\right). \tag{28}$$

The use of a GMM with full covariance matrices leads to a large number of parameters, presenting the risk of over-fitting. The covariance matrices are therefore often constrained to be spherical or diagonal

(Moerland, 1999). In this study, the diagonal covariance matrices were employed and behaved well throughout the experiments, i.e., $\boldsymbol{\Sigma}_k = \left\{\sigma_{kj}\right\}_{j=1}^{m} = \boldsymbol{\sigma}_k$, where m is the dimension of feature vectors, and σ_{kj} the variance value of the j^{th} attribute in the k^{th} component. Thus, i denotes the class label. Then the parameter set in our binary classifier based on GMM is

$$\Lambda = \{\tilde{P}_x, w_k^i, \boldsymbol{\mu}_k^i, \boldsymbol{\sigma}_k^i\},\ i=0,1,\ k=1,2,...,K. \tag{29}$$

Some parameters in Eq. (29) must satisfy certain constraints, which are transformed to an unconstrained domain for easier implementation in the learning stage. The constraints and transformation of parameters are listed in Table 1. A tiny variance value in the covariance matrices of GMM can lead to the computational instability of the class-conditional probability density function; thus, we imposed a positive minimum limit on the variance value, denoted as τ in Table 1. Consequently, the transformed parameter set is

$$\tilde{\Lambda} = \{\tilde{P}_x, \tilde{w}_k^i, \tilde{\boldsymbol{\mu}}_k^i, \tilde{\boldsymbol{\sigma}}_k^i\},\ i=0,1,\ k=1,2,...,K, \tag{30}$$

where $\tilde{\boldsymbol{\sigma}}_k^i = \{\tilde{\sigma}_{kj}^i\}_{j=1}^m$. $\tilde{\Lambda}$ is to be updated in the learning stage, and then transformed inversely into Λ after the learning is completed.

Table 1 Constraints and transformation of parameters in the learning stage

Original parameter and constraint	Transformation of parameters
$\sigma_{kj}^i > \tau$, $i=0,1,\ k=1,2,...,K.$	$\sigma_{kj}^i = \exp(\tilde{\sigma}_{kj}^i) + \tau$, $i=0,1,\ k=1,2,...,K.$
$\sum_{k=1}^{K} w_k = 1.$	$w_k = \dfrac{\exp(\tilde{w}_k)}{\sum_{k=1}^{K}\exp(\tilde{w}_k)},\ k=1,2,...,K.$

3.2 Initialization strategy

An initial parameter set is needed to trigger the optimization process described in Section 2.2. We employed the commonly used clustering algorithm of k-means to complete this task. For each class, its samples were clustered into k groups by using the k-means algorithm. Each group corresponded with a component in GMM. The mean and variance of the component are those of the samples in the group. We calculated the ratio of the number of the samples in the group to the total number of all samples, and took this ratio as the weight. Thus, the training time complexity of the proposed algorithm is $O(cmTKN^2\log N)$, where T is the number of training iterations, N is the number of training examples, m is the number of attributes, K is the number of GMM components, and c is the number of classes. The time complexity of recognition is $O(cmK)$.

4 Experiments and discussion

4.1 Experimental setup

4.1.1 Datasets

We evaluated the proposed CMM approach on the benchmark datasets for multivariate classification from the UCI Machine Learning Repository (University of California, 2013). Since the inputs into GMM should be real-value feature vectors, we used 10 datasets from the repository for multivariate classification, containing only real attributes, to test our approach. For each dataset, the experiments were conducted by following either the instructions in the dataset description or experimental schemes detailed in previous work if the instruction was not available. In summary, we have two types of experimental scheme: (1) 10-fold cross-validation and (2) fixed training and test sets. Table 2 shows the 10 datasets and corresponding experimental schemes, where '#samples', '#training', and '#test' denote the numbers of all samples, training samples, and test samples, respectively.

4.1.2 Algorithm parameters tuning

The parameters in our CMM learning algorithm include h, ξ (in problem (17)), β, ε, initial M (Algorithm 1), ε' (Algorithm 2), τ (Table 1), and K for GMM modeling.

The larger h, the more precise softmax or softmin computation. The value of 10 is enough for these experiments. As the separation between two classes should become larger and larger along with the iterations in Algorithm 1, we initially set a small value (2.6 in all the following experiments). After each iteration of Algorithm 1 is completed, if the constraints in problem (17) have been satisfied, we scale up ξ by a factor of 10.

Table 2 The UCI datasets used for multivariate classification, containing only real attributes

No.	Dataset	Experimental scheme
1	Breast cancer Wisconsin (diagnostic)	10-fold cross-validation; #samples=569
2	Connectionist bench (sonar, mines vs. rocks)	Fixed; #training=104; #test=104
3	Glass identification	10-fold cross-validation; #samples=214
4	Image segmentation	Fixed; #training=210; #test=2100
5	Iris	10-fold cross-validation; #samples=150
6	MAGIC gamma telescope	10-fold cross-validation; #samples=19 020
7	Statlog (image segmentation)	10-fold cross-validation; #samples=2310
8	Waveform database generator (version 1)	10-fold cross-validation; #samples=5000
9	Waveform database generator (version 2)	10-fold cross-validation; #samples=5000
10	Wine	10-fold cross-validation; #samples=178

The values of β, ε, ε', and the initial value of M were empirically set to be 10, 0.0001, 0.001, and 1 in all the following experiments, respectively.

We conducted the experiments on each dataset using different values of τ. The value increases from 0.000 025 to 0.250 000, multiplied by 10 each time. The final value of τ corresponding to the best result for each dataset is given in Table 3.

For K (the number of components in GMM), we tested the numbers from three to five, and determined the best one according to the experimental results. The final value of K corresponding to the best result for each dataset is given in Table 3.

Table 3 The best τ and K for each dataset

Dataset	τ	K	Dataset	τ	K
1	0.000 025	3	6	0.000 025	4
2	0.002 500	4	7	0.002 500	5
3	0.025 000	5	8	0.000 025	5
4	0.002 500	4	9	0.025 000	4
5	0.025 000	5	10	0.002 500	3

4.2 Experimental results

4.2.1 Comparison between two strategies of multi-class classification

We tested two strategies of multi-class classification, 1v1 and 1vO, for each dataset. The corresponding recognition rates are shown in Table 4. According to Table 4, 1v1 performs better than 1vO. The performance of our CMM approach is the better one in the two strategies.

Table 4 Recognition rates from two strategies of multi-class classification

Dataset	Recognition rate (%)			
	Training set		Test set	
	1v1	1vO	1v1	1vO
1	95.90	95.90	98.46	98.46
2	98.08	98.08	84.62	84.62
3	82.71	83.49	47.06	52.60
4	96.19	88.57	90.48	85.33
5	98.30	94.30	97.33	94.00
6	78.65	78.65	78.73	78.73
7	90.84	80.85	90.13	80.39
8	87.23	87.60	85.26	86.52
9	87.73	87.99	85.52	86.44
10	99.75	99.25	98.33	97.77
Average	91.54	89.47	85.59	84.49

1v1: one against one; 1vO: one against others

4.2.2 Comparison with expectation maximization and support vector machine

We compared the performance of our CMM approach with those of EM and SVM, respectively. They are classical representatives of generative learning and discriminative learning, respectively. The EM algorithm was implemented by the Open Source Computer Vision Library (OpenCV Team, 2015). The number of components in GMM for EM learning was set to be the same as that used in CMM learning. We used the generative learning algorithm EM to estimate the parameters in GMM, and labeled samples by their likelihood ratio of GMM. The SVM method was implemented by the sequential minimal optimization (SMO) algorithm in WEKA (Hall et al., 2009), a machining learning library in Java. Four kinds of kernel functions for SVM are provided in WEKA, including the poly kernel, normalized poly kernel, Pearson VII universal kernel/Pearson VII function based universal kernel (PUK), and radial basis function (RBF) kernel. We tested all four kernel functions and attempted to tune the corresponding parameters for each dataset. The resultant best recognition rates are reported in the following.

The performance comparisons of CMM, EM, and SVM are shown in Tables 5–7, which correspond to the recognition rates on training sets, the recognition rates on test sets, and the generalization ability, respectively. In Tables 5 and 6, IR_EM and IR_SVM demonstrate the increase in the recognition rates with CMM, compared with those of EM and SVM, respectively. In Table 7, IRG_EM and IRG_SVM also show CMM's increase of the generalization ability rates, compared with EM and SVM, respectively.

The data in Tables 5 and 6 shows that our CMM approach brings better average recognition rates than EM and SVM. On the training sets, CMM performs better on six datasets compared with EM, and seven datasets compared with SVM. The average increase in recognition rates on the 10 training datasets delivered by CMM is 0.30% and 5.54%, compared with EM and SVM, respectively. On the test sets, CMM performs better on eight datasets compared with EM, and seven datasets compared with SVM. The average increase in the recognition rates on the 10 test datasets delivered by CMM are 5.70% and 13.56%, compared with EM and SVM, respectively.

Table 5 Recognition rates on training sets of datasets

Method dataset	Recognition rate (%)				
	EM	SVM	CMM	IR_EM	IR_SVM
1	95.02	97.93	95.90	0.93	−2.07
2	93.27	86.54	98.08	5.16	13.33
3	81.05	58.51	83.49	3.01	42.69
4	93.33	90.95	96.19	3.06	5.76
5	98.45	96.52	98.30	−0.15	1.84
6	84.09	82.08	78.65	−6.47	−4.18
7	92.61	93.27	90.84	−1.91	−2.61
8	87.43	87.47	87.60	0.19	0.15
9	88.77	87.84	87.99	−0.88	0.17
10	99.69	99.44	99.75	0.06	0.31
Average	91.37	88.06	91.68	0.30	5.54

EM: expectation-maximization; SVM: support vector machine; CMM: constrained maximum margin. IR_EM and IR_SVM demonstrate the increase in the recognition rates with CMM, compared with those of EM and SVM, respectively

Table 7 shows the generalization ability of the three methods. As shown here, our CMM approach delivers better generalization performance on eight datasets, compared with EM, and half of the datasets, compared with SVM. The average increase in rates of generalization ability on the 10 datasets delivered by CMM are 5.29% and 6.06% above the EM and SVM, respectively.

Table 6 Recognition rates on test sets of datasets

Method dataset	Recognition rate (%)				
	EM	SVM	CMM	IR_EM	IR_SVM
1	95.08	98.20	98.46	3.55	0.26
2	77.88	80.77	84.62	8.65	4.77
3	36.82	24.52	52.60	42.86	114.52
4	88.24	87.43	90.48	2.54	3.49
5	96.00	96.67	97.33	1.39	0.68
6	80.80	70.21	78.73	−2.56	12.14
7	91.21	92.47	90.13	−1.18	−2.53
8	86.36	87.04	86.52	0.19	−0.60
9	86.08	86.48	86.44	0.42	−0.05
10	97.22	95.55	98.33	1.14	2.91
Average	83.57	81.93	86.27	5.70	13.56

EM: expectation-maximization; SVM: support vector machine; CMM: constrained maximum margin. IR_EM and IR_SVM demonstrate the increase in the recognition rates with CMM, compared with those of EM and SVM, respectively

Table 7 Generalization ability on each dataset

Method dataset	Generalization ability (%)				
	EM	SVM	CMM	IRG_EM	IRG_SVM
1	100.06	100.28	102.67	2.60	2.39
2	83.50	93.33	86.28	3.33	−7.56
3	45.43	41.91	63.00	38.68	50.34
4	94.55	96.13	94.06	−0.51	−2.15
5	97.51	100.16	99.01	1.54	−1.14
6	96.09	85.54	100.10	4.18	17.03
7	98.49	99.14	99.22	0.74	0.08
8	98.78	99.51	98.77	−0.01	−0.74
9	96.97	98.45	98.24	1.31	−0.22
10	97.52	96.09	98.58	1.08	2.59
Average	90.89	91.05	93.99	5.29	6.06

EM: expectation-maximization; SVM: support vector machine; CMM: constrained maximum margin. IRG_EM and IRG_SVM demonstrate the increase in the generalization ability rates with CMM, compared with those of EM and SVM, respectively

4.2.3 Comparison with the best previous results

We collected the results from the previous work for each dataset as far as we could, and compared our results with the best of them. To make the comparisons as fair as possible, the experimental schemes for testing our CMM approach were the same as those for the corresponding best previous results. Our CMM approach delivers better recognition rates on six test sets. The details are given in Table 8, where 'Reference' indicates the citation of the relevant papers listed in the reference section, and 'IR' denotes the increase in rates associated with our CMM approach, compared with these best previous results. Note that

Table 8 Comparisons between recognition rates on test sets from the CMM approach and those of the best results from previous work

Dataset	Previous method	Reference	Recognition rate (%)		IR
			Ours	Best previous	
1	HNB	Jiang et al. (2009)	98.46	95.78	2.80
2	Neural-network	Gorman and Sejnowski (1988)	84.62	90.40	−6.39
3	*k*-SVM	Bredensteiner and Bennett (1999)	52.60	72.43	−27.38
4	SVM	Kwok (1999)	90.48	90.20	0.31
5	3NN	Jiang and Zhou (2004)	97.33	95.67	1.74
6	C5.0	Dvořák and Savický (2007)	78.73	86.77	−9.27
7	WAODE	Jiang et al. (2012)	90.13	95.03	−5.16
8	HNB	Jiang et al. (2009)	86.52	85.46	1.24
9	AODE	Webb et al. (2005)	86.44	84.87	1.85
10	WAODE	Jiang et al. (2012)	98.33	96.63	1.76

IR denotes the increase in rates associated with the constrained maximum margin (CMM) approach, compared with the best results from previous work. HNB: hidden naïve Bayes; SVM: support vector machine; NN: nearest neighbor; AODE: aggregating one-dependence estimators; WAODE: weighted AODE

although we had tried our best to collect the best previous results, it is difficult to confirm that we have not omitted any good prior work.

4.2.4 Statistical comparison of different classifiers

To achieve a more comprehensive analysis of the results, we compared our proposed method with some other state-of-the-art Bayesian classifiers on the 10 UCI datasets, including tree augmented naïve Bayes (TAN) (Friedman et al., 1997), averaged one-dependence estimation estimators (AODE) (Webb et al., 2005), hidden naïve Bayes (HNB) (Jiang et al., 2009), and weighted AODE (WAODE) (Jiang et al., 2012). We employed the existing implementations of TAN, AODE, HNB, and WAODE in the WEKA platform. To process the numeric input feature vectors, we used the testing technique proposed by Jiang et al. (2009). We chose the unsupervised discretized filter to preprocess the numeric attributes, which in the WEKA platform is the unsupervised 10-bin discretization. Combining the previous results of both EM and SVM, the recognition rates of the seven algorithms are displayed in Table 9. These results demonstrate that on the 10 UCI datasets with numeric attributes, our proposed algorithm has a higher averaged recognition rate, which outperforms other counterparts and tends to have a robust performance over different datasets.

The non-parametric Wilcoxon test was further performed for the comparison of each pair of algorithms. By taking advantage of the KEEL data-mining tool (Alcalá-Fdez et al., 2009, 2011), the Wilcoxon signed-rank test ranks the differences in performances of two classifiers of each dataset, ignoring the signs, and compares the ranks for the positive and the negative differences (Demšar, 2006). The Wilcoxon test ranking results are presented in Table 10, where the sum of ranks for the datasets on which the algorithm in the row outperforms the algorithm in the corresponding column is shown in each value below the diagonal (the sum of ranks for positive differences, denoted by R^+), and each value above the diagonal reflects the sum of ranks for the datasets on which the algorithm in the column is worse than the algorithm in the corresponding row (the sum of ranks for the negative differences, denoted by R^-). The decision rule of the Wilcoxon signed-rank test is to reject the null-hypothesis when the smaller of R^+ and R^- is equal to or less than the critical value (for a confidence level of α=0.05 and N=10 datasets, the critical value is 8). Although no element in Table 10 was found to be less than or equal to the corresponding critical value, the difference in performance of these algorithms is shown in Table 11. The proposed CMM improves the performance of EM and SVM, and TAN is outperformed by both HNB and WAODE. The average recognition rate of CMM (86.36%) is also much higher than those of EM (83.57%) and SVM (81.93%). It is the highest among the compared algorithms.

Table 9 Recognition rates on test sets: comparisons between CMM and TAN, AODE, HNB, WAODE, EM, and SVM

Dataset	Recognition rate (%)						
	CMM	TAN	AODE	HNB	WAODE	EM	SVM
1	98.46	95.43	95.25	95.78	95.08	95.08	98.20
2	84.62	75.39	79.04	80.89	78.04	77.88	80.77
3	52.60	58.64	61.13	59.33	59.58	36.82	24.52
4	90.48	81.03	80.54	80.79	80.52	88.24	87.43
5	97.33	94.07	94.47	93.93	95.87	96.00	96.67
6	78.73	82.92	81.13	81.49	81.07	80.80	70.21
7	90.13	93.91	92.94	94.72	95.03	91.21	92.47
8	86.52	78.38	85.22	85.46	85.00	86.36	87.04
9	86.44	79.10	84.87	84.31	84.00	86.08	86.48
10	98.33	93.26	96.07	94.94	96.63	97.22	95.55
Average	86.36	83.21	85.07	85.16	85.08	83.57	81.93

CMM: constrained maximum margin; TAN: tree augmented naïve Bayes; AODE: aggregating one-dependence estimators; HNB: hidden naïve Bayes; WAODE: weighted AODE; EM: expectation maximization; SVM: support vector machine

Table 10 Ranks computed by the Wilcoxon test

Algorithm	Sum of ranks for the datasets						
	CMM	TAN	AODE	HNB	WAODE	EM	SVM
CMM	–	42.0	37.0	34.0	35.0	46.0	46.0
TAN	13.0	–	13.0	9.0	9.0	19.0	20.0
AODE	18.0	42.0	–	27.0	32.0	26.0	22.0
HNB	21.0	46.0	28.0	–	33.5	28.0	25.0
WAODE	20.0	46.0	23.0	21.5	–	21.0	26.0
EM	9.0	36.0	29.0	27.0	24.0	–	29.0
SVM	9.0	35.0	33.0	30.0	29.0	26.0	–

Values above the diagonal indicate that the algorithm in the row outperforms the one in the column; values below the diagonal indicate the opposite. CMM: constrained maximum margin; TAN: tree augmented naïve Bayes; AODE: aggregating one-dependence estimators; HNB: hidden naïve Bayes; WAODE: weighted AODE; EM: expectation maximization; SVM: support vector machine

Table 11 Summary of the Wilcoxon test

Algorithm	CMM	TAN	AODE	HNB	WAODE	EM	SVM
CMM	–					•	•
TAN		–		○	○		
AODE			–				
HNB				–			
WAODE					–		
EM						–	
SVM							–

• indicates that the method in the row improves the one in the column; ○ indicates that the method in the column improves the one of the row. Above the diagonal level of significance, α=0.90; below the diagonal level of significance, α=0.95. CMM: constrained maximum margin; TAN: tree augmented naïve Bayes; AODE: aggregating one-dependence estimators; HNB: hidden naïve Bayes; WAODE: weighted AODE; EM: expectation maximization; SVM: support vector machine

5 Conclusions

In this paper, we have proposed a novel discriminative learning approach to Bayesian pattern classification, called 'constrained maximum margin (CMM)'. We focused on the most representative representation of pattern information and high recognition rates with a good generalization ability, compared with most of the previous research, which has concentrated mainly on the separation between samples from different classes. We capitalized on margin maximization of different patterns, and corrected classification constraints for discriminative learning. The differences between CMM and the previous discriminative learning approaches to Bayesian classification are as follows:

1. The margin between two classes is defined at the class level, instead of the sample level. Since we took all the samples of a class as a whole, the separation between all the samples of a class and those of another class would be maximized, while the separation between two classes is measured on each single sample in similar previous work. More statistical information will be involved during the learning for different patterns when the margin is defined at the class level.

2. Except for the objective of MM between two classes, the constraints of correct classification of training data were considered further in our approach. The penalty factor and its magnification coefficient of the constraint item will boost the learning results by increasing the effect of misclassification samples during the optimization. Furthermore, this constraint optimization problem was solved by using the sequential unconstrained minimization technique.

To evaluate the proposed CMM approach, we applied it to learning Bayesian classifiers based on GMMs and conducted experiments on 10 datasets, which were established for classification and contained only real attributes from the well-known UCI Machine Learning Repository. In the experiments, the performance of our CMM approach was compared with those of EM, SVM, and some state-of-the-art Bayesian classifiers. Additionally, we performed the non-parametric Wilcoxon test on different classifiers. According to the average recognition rates and generalization ability measured on the datasets, the CMM approach performed better than EM and SVM. Note that CMM delivered results better than the best results from similar previous work on six of 10 datasets. The statistical comparison indicates that our CMM not only outperforms its counterparts in terms of the average recognition rate, but also improves the classification performance of EM and SVM. These experimental results show that our CMM approach is effective and promising.

References

Alcalá-Fdez J, Sanchez L, Garcia S, et al., 2009. KEEL: a software tool to assess evolutionary algorithms for data mining problems. *Soft Comput*, 13(3):307-318. https://doi.org/10.1007/s00500-008-0323-y

Alcalá-Fdez J, Fernández A, Luengo J, et al., 2011. KEEL data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. *J Multi-Valued Log Soft Comput*, 17(2-3):255-287.

Bredensteiner EJ, Bennett KP, 1999. Multicategory classification by support vector machines. In: Pang JS (Ed.), Computational Optimization. Springer US, New York, p.53-79. https://doi.org/10.1007/978-1-4615-5197-3_5

Dempster AP, Laird NM, Rubin DB, 1977. Maximum likelihood from incomplete data via the EM algorithm. *J R Stat Soc B*, 39(1):1-38.

Demšar J, 2006. Statistical comparisons of classifiers over multiple data sets. *J Mach Learn Res*, 7(Jan):1-30.

Dong W, Zhou M, 2014. Gaussian classifier-based evolutionary strategy for multimodal optimization. *IEEE Trans Neur Netw Learn Syst*, 25(6):1200-1216. https://doi.org/10.1109/TNNLS.2014.2298402

Dvořák J, Savický P, 2007. Softening splits in decision trees using simulated annealing. Int Conf on Adaptive and Natural Computing Algorithms, p.721-729. https://doi.org/10.1007/978-3-540-71618-1_80

Fiacco AV, McCormick GP, 1990. Nonlinear Programming: Sequential Unconstrained Minimization Techniques. SIAM, Philadelphia. https://doi.org/10.1137/1.9781611971316

Forsythe GE, Malcolm MA, Moler CB, 1977. Computer Methods for Mathematical Computations (1st Ed.). Prentice Hall, New Jersey.

Friedman N, Geiger D, Goldszmidt M, 1997. Bayesian network classifiers. *Mach Learn*, 29(2-3):131-163. https://doi.org/10.1023/A:1007465528199

Gorman RP, Sejnowski TJ, 1988. Analysis of hidden units in a layered network trained to classify sonar targets. *Neur Netw*, 1(1):75-89. https://doi.org/10.1016/0893-6080(88)90023-8

Hall M, Frank E, Holmes G, et al., 2009. The WEKA data mining software: an update. *ACM SIGKDD Explor Newsl*, 11(1):10-18. https://doi.org/10.1145/1656274.1656278

Jiang H, 2010. Discriminative training of HMMs for automatic speech recognition: a survey. *Comput Speech Lang*, 24(4):589-608. https://doi.org/10.1016/j.csl.2009.08.002

Jiang L, Zhang H, Cai Z, 2009. A novel Bayes model: hidden naïve Bayes. *IEEE Trans Knowl Data Eng*, 21(10): 1361-1371. https://doi.org/10.1109/TKDE.2008.234

Jiang L, Zhang H, Cai Z, et al., 2012. Weighted average of one-dependence estimators. *J Exp Theor Artif Intell*, 24(2):219-230. https://doi.org/10.1080/0952813X.2011.639092

Jiang Y, Zhou ZH, 2004. Editing training data for *k*NN classifiers with neural network ensemble. Advances in Neural Networks—Int Symp on Neural Networks, p.356-361. https://doi.org/10.1007/978-3-540-28647-9_60

Juang BH, Katagiri S, 1992. Discriminative learning for minimum error classification (pattern recognition). *IEEE Trans Signal Process*, 40(12):3043-3054. https://doi.org/10.1109/78.175747

Karabatak M, 2015. A new classifier for breast cancer detection based on naïve Bayesian. *Measurement*, 72:32-36.

https://doi.org/10.1016/j.measurement.2015.04.028

Kim BH, Pfister HD, 2011. An iterative joint linear-programming decoding of LDPC codes and finite-state channels. IEEE Conf on Communications, p.1-6. https://doi.org/10.1109/icc.2011.5962814

Kwok JTY, 1999. Moderating the outputs of support vector machine classifiers. *IEEE Trans Neur Netw*, 10(5): 1018-1031. https://doi.org/10.1109/72.788642

Moerland P, 1999. A comparison of mixture models for density estimation. 9th Int Conf on Artificial Neural Networks, p.25-30. https://doi.org/10.1049/cp:19991079

Nádas A, 1983. A decision theoretic formulation of a training problem in speech recognition and a comparison of training by unconditional versus conditional maximum likelihood. *IEEE Trans Audio Speech Signal Process*, 31(4):814-817. https://doi.org/10.1109/TASSP.1983.1164173

OpenCV Team, 2015. Open Source Computer Vision Library. http://opencv.org [Accessed on July 15, 2016].

Pernkopf F, Wohlmayr M, 2010. Large margin learning of Bayesian classifiers based on Gaussian mixture models. Joint European Conf on Machine Learning and Knowledge Discovery in Databases, p.50-66. https://doi.org/10.1007/978-3-642-15939-8_4

Pernkopf F, Wohlmayr M, Tschiatschek S, 2012. Maximum margin Bayesian network classifiers. *IEEE Trans Patt Anal Mach Intell*, 34(3):521-532. https://doi.org/10.1109/TPAMI.2011.149

Povey D, Woodland PC, 2002. Minimum phone error and I-smoothing for improved discriminative training. IEEE Int Conf on Acoustics, p.105-108. https://doi.org/10.1109/ICASSP.2002.5743665

University of California, 2013. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml [Accessed on Aug. 10, 2016].

Vapnik V, 2013. The Nature of Statistical Learning Theory (2nd Ed.). Springer-Verlag, New York. https://doi.org/10.1007/978-1-4757-3264-1

Vlassis N, Likas A, 1999. A kurtosis-based dynamic approach to Gaussian mixture modeling. *IEEE Trans Syst Man Cybern A*, 29(4):393-399. https://doi.org/10.1109/3468.769758

Webb GI, Boughton JR, Wang Z, 2005. Not so naïve Bayes: aggregating one-dependence estimators. *Mach Learn*, 58(1):5-24. https://doi.org/10.1007/s10994-005-4258-6

Woodland PC, Povey D, 2002. Large scale discriminative training of hidden Markov models for speech recognition. *Comput Speech Lang*, 16(1):25-47. https://doi.org/10.1006/csla.2001.0182