



GMM discriminant analysis with noisy label for each class

Jian-wei Liu¹ · Zheng-ping Ren¹ · Run-kun Lu¹ · Xiong-lin Luo¹

Received: 8 June 2019 / Accepted: 13 May 2020 / Published online: 1 June 2020
© Springer-Verlag London Ltd., part of Springer Nature 2020

Abstract

Real-world datasets often contain noisy labels, and learning from such datasets using standard classification approaches may not produce the desired performance. In this paper, we propose a Gaussian Mixture Discriminant Analysis (GMDA) with noisy label for each class. We introduce flipping probability and class probability and use EM algorithms to solve the discriminant problem with label noise. We also provide the detail proofs of convergence. Experimental results on synthetic and real-world datasets show that the proposed approach notably outperforms other four state-of-the-art methods.

Keywords Gaussian mixture models · Label noise · Discriminant analysis · Maximum likelihood estimate

1 Introduction

Noisy label problem has been investigated for a long time in the machine learning literature, and label-noise robust algorithms have numerous applications in medical image processing, spam filtering [1–3], Alzheimer’s disease prediction [1], gene expression classification [4], text processing [5–11], and image recognition [12–17]. Noisy labels are introduced by expert error and other unknown and unexpected factors. Mislabeled instances may lead to various potential negative consequences: bias the learning process, debase the prediction accuracy, and increase algorithm complexity of inferred models [3, 4] and the number of necessary labeling training samples, which is often produced by an expensive and time-consuming hand-annotation process or inefficient automatic annotation [1, 2], and increase difficulties in feature selection [18, 19]. The methods to deal with label noise can be classified into three [1]: (1) The label noise is ignored, and approaches that are robust to the presence of label noise, such as ensemble AdaBoost [20] and decision trees [5], are searched; (2) mislabeled instances are detected and removed, and then, cleaned training samples [21], [22] are used to

learn; and (3) models considering label noise are designed, and label noise-tolerant methods are determined. Label noise-tolerant methods enable researchers to take advantage of noise knowledge and use more sample information than noise-cleansing methods. The disadvantages are the increment in algorithm complexity and the increase in the number of parameters to estimate.

Bootkrajang presented a robust normal discriminant analysis (rNDA) algorithm [23]. The algorithm solves the maximum likelihood estimate problem by employing the expectation maximization (EM) algorithm [24–26]. The rNDA model assumes that the examples in each class obey single Gaussian distribution; it is scarcely to verify. Thus, its performance on datasets that are not strictly Gaussian in each class seems insufficient.

Numerous studies on GMM have appeared in many fields, such as outlier mining [27], image processing [28, 29], clustering [30], and community detection [6]. [27] devised an approach to adapt to a continuously evolving outlier distribution. [31] proposed initializing mean vectors by choosing points with higher concentrations of neighbors and using a truncated normal distribution for the preliminary estimation of dispersion matrices. DivideMix models the per-sample loss distribution with a mixture model to dynamically divide the training data into a labeled set with clean samples and an unlabeled set with noisy samples [28]. [29] addressed noisy labels issue and proposed selective negative learning and positive learning approach trained using a complementary label. [30] constructed a kernel Fisher discriminant (KFD) from training examples

✉ Jian-wei Liu
liujw@cup.edu.cn

¹ Department of Automation, College of Information Science and Engineering, China University of Petroleum, Beijing Campus (CUP), 260 mailbox China University of Petroleum, Changping District, Beijing 102249, China

with noisy labels. [6] presented a procedure for community detection using GMMs that incorporates certain truncation and shrinkage effects that arise in the nonvanishing noise regime.

To solve this problem, we propose a new scheme to carry out the discriminant analysis with Gaussian mixture models (GMM) which has the ability to handle the non-Gaussian distributions. We employ a linear combination of Gaussian distributions to approximate the probabilistic distributions in each class and use the EM algorithm to solve the maximum likelihood estimate [25, 26, 32]. In the last several decades, researchers in the fields of statistics and computer vision have been interested in GMM.

The discriminant analysis discussed in this paper uses GMM to approximate data distributions and is applied to classification in the case of label noise. Maximum likelihood estimate method is used to determine the parameters. Moreover, this study derives the updating formulas of the parameters of the proposed Gaussian Mixture Discriminant Analysis (GMDA). The performance of GMDA is then compared with that of AdaBoost, rNDA, rLR, and rmLR [6] on two synthetic and six real-world datasets. Results show that our method can effectively and correctly estimate the parameters of both distribution and noise.

Our main contributions are as follows:

1. We propose a general discriminant analysis framework for attacking the noisy label problem. Different from previous approach, in this framework, the probabilistic distribution of each class on real data is captured by GMM instead of the single Gaussian distribution, and this single Gaussian assumption for each class is apparently too harsh to be verified and can scarcely reflect the actual scenarios.
2. We show that when flipping probability and class probability are introduced, the parameters of GMDA model and posterior probability to predict an unlabeled instance can be computed by using EM algorithm.
3. We provide the detail proofs of convergence for general situation, e.g., Gaussian classes with noisy labels and class-conditional Gaussian mixtures with noisy labels.
4. We have conducted extensive experiments on two synthetic datasets and six real-life datasets, which have different properties and scales, to demonstrate the effectiveness and efficiency of our proposed formulation.

The rest of this paper is organized as follows: The proposed GMMs discriminant analysis with noisy label for each class is formally introduced in Sect. 2, and convergence analysis for general situation, Gaussian classes with noisy labels, and class-conditional Gaussian mixtures with noisy labels are presented in Sect. 3, and related work is

presented in Sect. 4. Experimental results using synthetic and real-world datasets are discussed in Sect. 5. Finally, the conclusion and future work are summarized in Sect. 6.

1.1 Discriminant analysis based on Gaussian mixture models

1.1.1 Description of the problem with the noise labels

Considering a statistical decision problem (pattern recognition, classification, and discrimination), we assume that some real data vectors

$$\mathbf{x} = (x_1, \dots, x_d) \in \mathcal{X}, \quad (\mathcal{X} = \mathcal{R}^d)$$

have to be classified with respect to a finite set of classes $\Omega = \{\omega_1, \omega_2, \dots, \omega_K\}$. The data vectors $\mathbf{x} \in \mathcal{X}$ are supposed to occur randomly according to some unknown class-conditional pdfs $p(\mathbf{x}|\omega)$ and the respective priori class probabilities $p(\omega)$, $\omega \in \Omega$.

In case of supervised learning, we are given a training set \mathcal{S}_ω for each class $\omega \in \Omega$:

$$\mathcal{S}_\omega = \{\mathbf{x} \in \mathcal{X}\}, \quad \omega \in \Omega; \quad \mathcal{S} = \bigcup_{\omega \in \Omega} \mathcal{S}_\omega, \quad |\mathcal{S}| = \sum_{\omega \in \Omega} |\mathcal{S}_\omega|,$$

where $|\mathcal{S}|$ and $|\mathcal{S}_\omega|$ denote the number of elements in set \mathcal{S}_ω . The decision problem can be solved by means of Bayes decision function by computing the maximum likelihood estimates of the conditional densities $p(\mathbf{x}|\omega)$. The related log-likelihood criterion L_ω is given by

$$L_\omega = \frac{1}{|\mathcal{S}_\omega|} \sum_{\mathbf{x} \in \mathcal{S}_\omega} \log p(\mathbf{x}|\omega) p(\omega), \quad (\omega \in \Omega),$$

with normalization coefficient $1/|\mathcal{S}_\omega|$ included for convenience.

In case of noisy labels, we assume that the true label $\omega \in \Omega$ of a given observation $\mathbf{x} \in \mathcal{S}$ may be randomly interchanged (flipped, corrupted, substituted), i.e., for each data vector $\mathbf{x} \in \mathcal{S}$, we are given a unique observed label $\tilde{\omega} \in \Omega$, which may differ from the true label ω . Obviously, the probabilities $p(\tilde{\omega})$ of the observed labels $\tilde{\omega}$ may differ from the true probabilities $p(\omega)$ of the true labels ω . Assuming randomly substituted labels, we denote that $p(\omega|\tilde{\omega})$ is the probability of the true label ω given the observed label $\tilde{\omega}$. In this sense, the conditional probability density $\tilde{p}(\mathbf{x}|\tilde{\omega})$ of $\mathbf{x} \in \mathcal{X}$ given an observed label $\tilde{\omega}$ is a mixture

$$\tilde{p}(\mathbf{x}|\tilde{\omega}) = \sum_{\omega \in \Omega} p(\mathbf{x}|\omega) p(\omega|\tilde{\omega}), \quad \mathbf{x} \in \mathcal{X}, \quad \tilde{\omega} \in \Omega,$$

and consequently, with the probability $p(\omega|\tilde{\omega})$, any of the classes $\omega \in \Omega$ can be the true source of the observation $\mathbf{x} \in \mathcal{X}$. Note that the probabilities $p(\omega|\tilde{\omega})$, $\omega \in \Omega$, $p(\tilde{\omega})$,

$\tilde{\omega} \in \Omega$ are generally unknown and have to be estimated from data.

1.1.2 Gaussian mixture model

Lawrence and Schölkopf [6] proposed a probabilistic approach to label noise, and Bootkrajang [8, 22] extended the same model to multi-class case, assuming a Gaussian density for each class. We propose discriminant analysis based on GMM where GMM is used to approximate the probabilistic distribution of each class on real data.

Supposed that the data log-likelihood is:

$$L = \log \prod_{\mathbf{x} \in \mathcal{S}} p(\mathbf{x}|\tilde{\omega}, \theta_{\tilde{\omega}})p(\tilde{\omega})$$

$$= \sum_{\mathbf{x} \in \mathcal{S}} \log p(\mathbf{x}|\tilde{\omega}, \theta_{\tilde{\omega}})p(\tilde{\omega})$$

under our assumption conditions, the given model may be no longer valid. Thus, a hidden variable $\omega \in \Omega$ is introduced to address the problem. The hidden variable ω is considered as the real class label:

$$L = \sum_{\mathbf{x} \in \mathcal{S}} \log \sum_{\omega \in \Omega} p(\mathbf{x}, \omega|\tilde{\omega}, \theta_{\tilde{\omega}})p(\tilde{\omega})$$

$$= \log \prod_{\mathbf{x} \in \mathcal{S}} \sum_{\omega \in \Omega} p(\mathbf{x}, \omega|\tilde{\omega}, \theta_{\tilde{\omega}})p(\tilde{\omega}). \tag{1}$$

The observed labels are generated from true labels and random noise. The joint probability ω and $\tilde{\omega}$ can be expressed as $p(\omega, \tilde{\omega}_n) = p(\tilde{\omega}_n|\omega)p(\omega)$. Input \mathbf{x} is conditionally independent from observed label $\tilde{\omega}_n$ after knowing true label ω , because the label noise is random. Thus, we change Eq. (1) into

$$L(\Theta) = \sum_{\mathbf{x} \in \mathcal{S}} \log \sum_{\omega \in \Omega} p(\mathbf{x}|\tilde{\omega}, \omega, \theta_{\tilde{\omega}_n})p(\tilde{\omega}_n, \omega)$$

$$= \sum_{\mathbf{x} \in \mathcal{S}} \log \sum_{\omega \in \Omega} p(\mathbf{x}|\omega, \theta_{\omega})p(\tilde{\omega}|\omega)p(\omega)$$

$$= \sum_{\tilde{\omega} \in \Omega} I(\tilde{\omega} = \omega) \sum_{\mathbf{x} \in \mathcal{S}} \log \sum_{\omega \in \Omega} p(\mathbf{x}|\omega, \theta_{\omega})p(\tilde{\omega}|\omega)p(\omega) \tag{2}$$

where $I(\cdot)$ is an indicator function. Flipping probability is defined as $\gamma_{\tilde{\omega}, \omega} \stackrel{def}{=} p(\tilde{\omega}|\omega)$, which is similar to Ref. [23], indicating the probability of true label ω flipped to the observed label $\tilde{\omega}$. Class probability is defined as $\pi_{\omega} \stackrel{def}{=} p(\omega)$, and the constraint conditions are $\sum_{\tilde{\omega} \in \Omega} \gamma_{\tilde{\omega}, \omega} = 1$ and $\sum_{\omega \in \Omega} \pi_{\omega} = 1$. The set of flipping probability and class probability are denoted as $\Gamma = \{\gamma_{\tilde{\omega}, \omega}\}_{\tilde{\omega} \in \Omega}$ and $\Pi = \{\pi_{\omega}\}_{\omega \in \Omega}$, respectively. The Gaussian mixture density function is

$$p(\mathbf{x}|\omega, \theta_{\omega}) = \sum_{m \in \mathcal{M}} w_m g(\mathbf{x}|\omega, \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)$$

where $\theta_{\omega} = \{w_m, \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m\}_{m \in \mathcal{M}}$ is the parameter set of the class k . The elements are weight, mean vector, and covariance matrix of the component m , and M is the number of components. The logarithm of a sum in (2) can be rewritten as:

$$\log \sum_{\omega \in \Omega} p(\mathbf{x}|\omega; \theta_{\omega})p(\tilde{\omega}|\omega)p(\omega)$$

$$= \log \sum_{\omega \in \Omega} \gamma_{\tilde{\omega}, \omega} \pi_{\omega} \sum_{m \in \mathcal{M}} w_m g(\mathbf{x}|\omega, \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m) \tag{3}$$

$$= \log \sum_{\omega \in \Omega} q(\omega) \frac{\gamma_{\tilde{\omega}, \omega} \pi_{\omega} \sum_{m \in \mathcal{M}} w_m g(\mathbf{x}|\omega, \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)}{q(\omega)}$$

where $q(\cdot)$ is an arbitrary distribution of ω , and the constraint condition is $\sum_{\omega \in \Omega} q(\omega) = 1$.

According to Jensen’s inequality [33], if Z is a random variable, and $g(\cdot)$ is a concave function, then

$$g(E(Z)) \geq E(g(Z))$$

Thus, the lower bound of (3) is derived. Since $g(\cdot)$ is a concave function, and again by Jensen’s inequality we have:

$$f\left(E_{\omega \sim q(\omega)}\left(\frac{\gamma_{\tilde{\omega}, \omega} \pi_{\omega} p(\mathbf{x}|\omega, \theta_{\omega})}{q(\omega)}\right)\right)$$

$$\geq E_{\omega \sim q(\omega)}\left(f\left(\frac{\gamma_{\tilde{\omega}, \omega} \pi_{\omega} p(\mathbf{x}|\omega, \theta_{\omega})}{q(\omega)}\right)\right),$$

i.e.,

$$\log \sum_{\omega \in \Omega} q(\omega) \frac{\gamma_{\tilde{\omega}, \omega} \pi_{\omega} p(\mathbf{x}|\omega, \theta_{\omega})}{q(\omega)} \geq \sum_{\omega \in \Omega} q(\omega) \log \frac{\gamma_{\tilde{\omega}, \omega} \pi_{\omega} p(\mathbf{x}|\omega, \theta_{\omega})}{q(\omega)}$$

A Lagrange multiplier λ_{G_1} is introduced to find a local optimum of the objective function subject to $\sum_{\omega \in \Omega} q(\omega) = 1$, and the corresponding Lagrange function is:

$$G_1 = \sum_{\omega \in \Omega} q(\omega) [\log \gamma_{\tilde{\omega}, \omega} \pi_{\omega} p(\mathbf{x}|\omega, \theta_{\omega}) - \log q(\omega)]$$

$$+ \lambda_{G_1} \left[1 - \sum_{\omega \in \Omega} q(\omega) \right]$$

By setting the derivative w.r.t $q(\omega)$ equal to zero, we obtain the formula as follows:

$$\log p(\mathbf{x}|\omega, \theta_{\omega}) \gamma_{\tilde{\omega}, \omega} \pi_{\omega} - \log q(\omega) - 1 - \lambda_{G_1} = 0$$

$$\Rightarrow \log q(\omega) = \log p(\mathbf{x}|\omega, \theta_{\omega}) \gamma_{\tilde{\omega}, \omega} \pi_{\omega} - 1 - \lambda_{G_1}. \tag{4}$$

$$\Rightarrow q(\omega) = p(\mathbf{x}|\omega, \theta_{\omega}) \gamma_{\tilde{\omega}, \omega} \pi_{\omega} \cdot e^{-(1+\lambda_{G_1})}$$

Further, we integrate $q(\omega)$ in the field of Ω , and the optimal solution w.r.t λ_{G_1} is:

$$\sum_{\omega \in \Omega} q(\omega) = e^{-(1+\lambda_{G_1})} \sum_{\omega \in \Omega} p(\mathbf{x}|\omega, \theta_\omega) \gamma_{\tilde{\omega}, \omega} \pi_\omega = 1$$

$$\Rightarrow \lambda_{G_1} = \log \sum_{\omega \in \Omega} p(\mathbf{x}|\omega, \theta_\omega) \gamma_{\tilde{\omega}, \omega} \pi_\omega - 1.$$

λ_{G_1} is plugged back into (4), and $q(\omega)$ is solved as follows:

$$q(\omega) = \frac{p(\mathbf{x}|\omega, \theta_\omega) \gamma_{\tilde{\omega}, \omega} \pi_\omega}{\sum_{\omega \in \Omega} p(\mathbf{x}|\omega, \theta_\omega) \gamma_{\tilde{\omega}, \omega} \pi_\omega} \tag{5}$$

$$= p(\omega|x, \tilde{\omega})$$

Jensen’s inequality is employed, and the lower bound of $\log p(\mathbf{x}|\omega, \theta_\omega) = \log \sum_{m \in \mathcal{M}} w_{\omega, m} g(\mathbf{x}|\omega, \boldsymbol{\mu}_{\omega, m}, \boldsymbol{\Sigma}_{\omega, m})$ is derived as

$$\log p(\mathbf{x}|\omega, \theta_\omega) = \log \sum_{m \in \mathcal{M}} w_{\omega, m} g(\mathbf{x}|\omega, \boldsymbol{\mu}_{\omega, m}, \boldsymbol{\Sigma}_{\omega, m})$$

$$= \log \sum_{m \in \mathcal{M}} h_{\omega, m} \frac{w_{\omega, m} g(\mathbf{x}|\omega, \boldsymbol{\mu}_{\omega, m}, \boldsymbol{\Sigma}_{\omega, m})}{h_{\omega, m}}$$

$$\geq \sum_{m \in \mathcal{M}} h_{\omega, m} \log \frac{w_{\omega, m} g(\mathbf{x}|\omega, \boldsymbol{\mu}_{\omega, m}, \boldsymbol{\Sigma}_{\omega, m})}{h_{\omega, m}}$$

$$= \sum_{m \in \mathcal{M}} h_{\omega, m} [\log w_{\omega, m} g(\mathbf{x}|\omega, \boldsymbol{\mu}_{\omega, m}, \boldsymbol{\Sigma}_{\omega, m}) - \log h_{\omega, m}]$$

where $\sum_{m \in \mathcal{M}} h_{\omega, m} = 1$. Lagrange multiplier λ_{G_2} is introduced again to solve for $h_{\omega, m}$, and the corresponding Lagrange function is as follows:

$$G_2 = \sum_{m \in \mathcal{M}} h_{\omega, m} [\log w_{\omega, m} g(\mathbf{x}|\omega, \boldsymbol{\mu}_{\omega, m}, \boldsymbol{\Sigma}_{\omega, m}) - \log h_{\omega, m}]$$

$$+ \lambda_{G_2} \left(1 - \sum_{m \in \mathcal{M}} h_{\omega, m} \right)$$

By setting the derivative w.r.t $h_{\omega, m}$ equal to zero, we obtain the formula as follows:

$$\log w_{\omega, m} g(\mathbf{x}|\omega, \boldsymbol{\mu}_{\omega, m}, \boldsymbol{\Sigma}_{\omega, m}) - \log h_{\omega, m} - 1 - \lambda_{G_2} = 0$$

$$\Rightarrow \log h_{\omega, m} = \log w_{\omega, m} g(\mathbf{x}|\omega, \boldsymbol{\mu}_{\omega, m}, \boldsymbol{\Sigma}_{\omega, m}) - 1 - \lambda_{G_2}$$

$$\Rightarrow h_{\omega, m} = w_{\omega, m} g(\mathbf{x}|\omega, \boldsymbol{\mu}_{\omega, m}, \boldsymbol{\Sigma}_{\omega, m}) \cdot e^{-(1+\lambda_{G_2})}$$

Furthermore, we integrate $h_{\omega, m}$ and the optimal solution w.r.t λ_{G_2} is:

$$\sum_{m \in \mathcal{M}} h_{\omega, m} = \sum_{m \in \mathcal{M}} w_{\omega, m} g(\mathbf{x}|\omega, \boldsymbol{\mu}_{\omega, m}, \boldsymbol{\Sigma}_{\omega, m}) \cdot e^{-(1+\lambda_{G_2})}$$

$$\Rightarrow \lambda_{G_2} = \log \sum_{m \in \mathcal{M}} w_{\omega, m} g(\mathbf{x}|\omega, \boldsymbol{\mu}_{\omega, m}, \boldsymbol{\Sigma}_{\omega, m}) - 1.$$

We plug λ_{G_2} back and solve for $h_{\omega, m}$ as follows:

$$h_{\omega, m} = \frac{w_{\omega, m} g(\mathbf{x}|\omega, \boldsymbol{\mu}_{\omega, m}, \boldsymbol{\Sigma}_{\omega, m})}{\sum_{m \in \mathcal{M}} w_{\omega, m} g(\mathbf{x}|\omega, \boldsymbol{\mu}_{\omega, m}, \boldsymbol{\Sigma}_{\omega, m})} \tag{6}$$

$$= \Pr(m|\mathbf{x}, \omega)$$

As a result, Eqs. (5) and (6) are plugged back to derive the objective function (7):

$$Q = \sum_{\tilde{\omega} \in \Omega} I(\tilde{\omega} = \omega) \sum_{\mathbf{x} \in \mathcal{S}} \sum_{\omega \in \Omega} \left\{ p(\omega|\mathbf{x}, \tilde{\omega}) \times \sum_{m \in \mathcal{M}} \Pr(m|\mathbf{x}, \omega) \log \frac{w_{\omega, m} g(\mathbf{x}|\omega, \boldsymbol{\mu}_{\omega, m}, \boldsymbol{\Sigma}_{\omega, m})}{\Pr(m|\mathbf{x}, \omega)} \right\}$$

$$+ \sum_{\mathbf{x} \in \mathcal{S}} \sum_{\omega \in \Omega} p(\omega|\mathbf{x}, \tilde{\omega}) \log \gamma_{\tilde{\omega}, \omega} + \sum_{\mathbf{x} \in \mathcal{S}} \sum_{\omega \in \Omega} p(\omega|\mathbf{x}, \tilde{\omega}) \log \pi_\omega \tag{7}$$

The parameters to be estimated are $\Theta = \{\theta_\omega\}_{\omega \in \Omega}$, $\Gamma = \{\gamma_{\tilde{\omega}, \omega}\}_{\tilde{\omega} \in \Omega}$, and $\Pi = \{\pi_\omega\}_{\omega \in \Omega}$, where $\theta_\omega = \{w_{\omega, m}, \boldsymbol{\mu}_{\omega, m}, \boldsymbol{\Sigma}_{\omega, m}\}_{m \in \mathcal{M}}$. The EM method is employed to solve for the parameters.

E step $p(\omega|\mathbf{x}, \tilde{\omega})$ and $\Pr(m|\mathbf{x}, \omega)$ are calculated according to Eqs. (5) and (6), which are listed as follows:

$$p(\omega|\mathbf{x}, \tilde{\omega}) = \frac{p(\mathbf{x}|\omega, \theta_\omega) \gamma_{\tilde{\omega}, \omega} \pi_\omega}{\sum_{\omega \in \Omega} p(\mathbf{x}|\omega, \theta_\omega) \gamma_{\tilde{\omega}, \omega} \pi_\omega} \tag{8}$$

$$\Pr(m|\mathbf{x}_n, \omega_n = k) = \frac{w_{\omega, m} g(\mathbf{x}|\omega, \boldsymbol{\mu}_{\omega, m}, \boldsymbol{\Sigma}_{\omega, m})}{\sum_{m \in \mathcal{M}} w_{\omega, m} g(\mathbf{x}|\omega, \boldsymbol{\mu}_{\omega, m}, \boldsymbol{\Sigma}_{\omega, m})} \tag{9}$$

M step: (7) is optimized to solve for the local optimum of $\Theta = \{\theta_\omega\}_{\omega \in \Omega}$, $\Gamma = \{\gamma_{\tilde{\omega}, \omega}\}_{\tilde{\omega} \in \Omega}$, and $\Pi = \{\pi_\omega\}_{\omega \in \Omega}$.

1.1.3 Updating

Updating rule of $\boldsymbol{\mu}_{\omega, m}$: By setting the derivative of Eq. (7) w.r.t $\boldsymbol{\mu}_{\omega, m}$ equal to zero, we derive the updating rule as follows:

$$\frac{\partial Q}{\partial \boldsymbol{\mu}_{\omega, m}} = \sum_{\tilde{\omega} \in \Omega} I(\tilde{\omega} = \omega) \sum_{\mathbf{x} \in \mathcal{S}} \left\{ p(\omega|\mathbf{x}, \tilde{\omega}) \cdot \Pr(m|\mathbf{x}, \omega) \cdot \boldsymbol{\Sigma}_{\omega, m}^{-1} \cdot (\mathbf{x} - \boldsymbol{\mu}_{\omega, m}) = 0 \right\}$$

$$\Rightarrow \boldsymbol{\mu}_{\omega, m} = \frac{\sum_{\tilde{\omega} \in \Omega} I(\tilde{\omega} = \omega) \sum_{\mathbf{x} \in \mathcal{S}} p(\omega|\mathbf{x}, \tilde{\omega}) \Pr(m|\mathbf{x}, \omega) \mathbf{x}}{\sum_{\tilde{\omega} \in \Omega} I(\tilde{\omega} = \omega) \sum_{\mathbf{x} \in \mathcal{S}} p(\omega|\mathbf{x}, \tilde{\omega}) \Pr(m|\mathbf{x}, \omega)} \tag{10}$$

Updating rule of $\boldsymbol{\Sigma}_{\omega, m}$: By setting the derivative of Eq. (7) w.r.t $\boldsymbol{\Sigma}_{\omega, m}$ equal to zero, we derive the updating rule as follows:

$$\begin{aligned} \frac{\partial Q}{\partial \Sigma_{\omega,m}} &= - \sum_{\tilde{\omega} \in \Omega} I(\tilde{\omega} = \omega) \sum_{\mathbf{x} \in \mathcal{S}} \{p(\omega|\mathbf{x}, \tilde{\omega}) \Pr(m|\mathbf{x}, \omega) \\ &\cdot [\Sigma_{\omega,m}^{-1} - \Sigma_{\omega,m}^{-1}(\mathbf{x} - \boldsymbol{\mu}_{\omega,m})(\mathbf{x} - \boldsymbol{\mu}_{\omega,m})^T \Sigma_{\omega,m}^{-1}]\} = 0 \Rightarrow \Sigma_{\omega,m} \\ &= \frac{\sum_{\tilde{\omega} \in \Omega} I(\tilde{\omega} = \omega) \sum_{\mathbf{x} \in \mathcal{S}} p(\omega|\mathbf{x}, \tilde{\omega}) \Pr(m|\mathbf{x}, \omega) (\mathbf{x} - \boldsymbol{\mu}_{\omega,m})(\mathbf{x} - \boldsymbol{\mu}_{\omega,m})^T}{\sum_{\tilde{\omega} \in \Omega} I(\tilde{\omega} = \omega) \sum_{\mathbf{x} \in \mathcal{S}} p(\omega|\mathbf{x}, \tilde{\omega}) \Pr(m|\mathbf{x}, \omega)} \end{aligned} \tag{11}$$

Updating rule of $w_{\omega,m}$: A Lagrange multiplier $\lambda_{w_{\omega,m}}$ is introduced to guarantee the constraint condition $\sum_{m \in \mathcal{M}} w_{\omega,m} = 1$, and the corresponding Lagrange function is designed as follows:

$$Q_{\lambda_{w_{\omega,m}}} = Q + \lambda_{w_{\omega,m}} \left(1 - \sum_{m \in \mathcal{M}} w_{\omega,m} \right),$$

and by setting the derivative of $Q_{\lambda_{w_{\omega,m}}}$ w.r.t $w_{\omega,m}$ equal to zero, we derive the formula as follows:

$$\begin{aligned} \frac{\partial Q_{\lambda_{w_{\omega,m}}}}{\partial w_{\omega,m}} &= 0 \\ &= \sum_{\tilde{\omega} \in \Omega} I(\tilde{\omega} = \omega) \sum_{\mathbf{x} \in \mathcal{S}} p(\omega|\mathbf{x}, \tilde{\omega}) \Pr(m|\mathbf{x}, \omega) \frac{1}{w_{\omega,m}} - \lambda_{w_{\omega,m}} \\ &\Rightarrow \lambda_{w_{\omega,m}} w_{\omega,m} = \sum_{\tilde{\omega} \in \Omega} I(\tilde{\omega} = \omega) \sum_{\mathbf{x} \in \mathcal{S}} p(\omega|\mathbf{x}, \tilde{\omega}) \Pr(m|\mathbf{x}, \omega) \\ \text{intergrate } w_{\omega,m} &\Rightarrow \lambda_{w_{\omega,m}} \sum_{m \in \mathcal{M}} w_{\omega,m} = \lambda_{w_{\omega,m}} \\ &= \sum_{\tilde{\omega} \in \Omega} I(\tilde{\omega} = \omega) \sum_{\mathbf{x} \in \mathcal{S}} p(\omega|\mathbf{x}, \tilde{\omega}) \sum_{m \in \mathcal{M}} \Pr(m|\mathbf{x}, \omega), \end{aligned}$$

and we plug $\lambda_{w_{\omega,m}}$ back and solve for $w_{\omega,m}$ as follows:

$$w_{\omega,m} = \frac{\sum_{\tilde{\omega} \in \Omega} I(\tilde{\omega} = \omega) \sum_{\mathbf{x} \in \mathcal{S}} p(\omega|\mathbf{x}, \tilde{\omega}) \Pr(m|\mathbf{x}, \omega)}{\sum_{\tilde{\omega} \in \Omega} I(\tilde{\omega} = \omega) \sum_{\mathbf{x} \in \mathcal{S}} p(\omega|\mathbf{x}, \tilde{\omega}) \sum_{m \in \mathcal{M}} \Pr(m|\mathbf{x}, \omega)} \tag{12}$$

Updating the rule of $\gamma_{\tilde{\omega},\omega}$: A Lagrange multiplier $\lambda_{\gamma_{\tilde{\omega},\omega}}$ is introduced to guarantee the constraint condition $\sum_{\tilde{\omega} \in \Omega} \gamma_{\tilde{\omega},\omega} = 1$, and the corresponding Lagrange function is designed as follows:

$$Q_{\lambda_{\gamma_{\tilde{\omega},\omega}}} = Q + \lambda_{\gamma_{\tilde{\omega},\omega}} \left(1 - \sum_{\tilde{\omega} \in \Omega} \gamma_{\tilde{\omega},\omega} \right),$$

and by setting the derivative of $Q_{\lambda_{\gamma_{\tilde{\omega},\omega}}}$ w.r.t $\gamma_{\tilde{\omega},\omega}$ equal to zero, we derive the formula as follows:

$$\begin{aligned} \frac{\partial Q_{\lambda_{\gamma_{\tilde{\omega},\omega}}}}{\partial \gamma_{\tilde{\omega},\omega}} &= \sum_{\mathbf{x} \in \mathcal{S}} p(\omega|\mathbf{x}, \tilde{\omega}) \frac{1}{\gamma_{\tilde{\omega},\omega}} - \lambda_{\gamma_{\tilde{\omega},\omega}} = 0 \\ &\Rightarrow \lambda_{\gamma_{\tilde{\omega},\omega}} \gamma_{\tilde{\omega},\omega} = \sum_{\mathbf{x} \in \mathcal{S}} p(\omega|\mathbf{x}, \tilde{\omega}) \\ \text{intergrate } \gamma_{\tilde{\omega},\omega} &\Rightarrow \lambda_{\gamma_{\tilde{\omega},\omega}} \sum_{\tilde{\omega} \in \Omega} \gamma_{\tilde{\omega},\omega} = \lambda_{\gamma_{\tilde{\omega},\omega}} \\ &= \sum_{\mathbf{x} \in \mathcal{S}} \sum_{\tilde{\omega} \in \Omega} p(\omega|\mathbf{x}, \tilde{\omega}) \end{aligned} ,$$

and we plug $\lambda_{\gamma_{\tilde{\omega},\omega}}$ back and solve for $\gamma_{\tilde{\omega},\omega}$ as follows:

$$\gamma_{\tilde{\omega},\omega} = \frac{\sum_{\mathbf{x} \in \mathcal{S}} p(\omega|\mathbf{x}, \tilde{\omega})}{\sum_{\mathbf{x} \in \mathcal{S}} \sum_{\tilde{\omega} \in \Omega} p(\omega|\mathbf{x}, \tilde{\omega})}. \tag{13}$$

Updating class probability: A Lagrange multiplier λ_{π_ω} is introduced to guarantee the constraint condition $\sum_{\omega \in \Omega} \pi_\omega = 1$, and a Lagrange function is designed as follows:

$$Q_{\lambda_{\pi_\omega}} = Q + \lambda_{\pi_\omega} \left(1 - \sum_{\omega \in \Omega} \pi_\omega \right),$$

and by setting the derivative of $Q_{\lambda_{\pi_\omega}}$ w.r.t π_ω equal to zero, we derive the formula as follows:

$$\begin{aligned} \frac{\partial Q_{\lambda_{\pi_\omega}}}{\partial \pi_\omega} &= \sum_{\tilde{\omega} \in \Omega} I(\tilde{\omega} = \omega) \sum_{\mathbf{x} \in \mathcal{S}} p(\omega|\mathbf{x}, \tilde{\omega}) \frac{1}{\pi_\omega} - \lambda_{\pi_\omega} = 0 \\ &\Rightarrow \lambda_{\pi_\omega} \pi_\omega = \sum_{\tilde{\omega} \in \Omega} I(\tilde{\omega} = \omega) \sum_{\mathbf{x} \in \mathcal{S}} p(\omega|\mathbf{x}, \tilde{\omega}) \\ \text{intergrate } \pi_\omega &\Rightarrow \lambda_{\pi_\omega} \sum_{\omega \in \Omega} \pi_\omega = \sum_{\tilde{\omega} \in \Omega} I(\tilde{\omega} = \omega) \sum_{\mathbf{x} \in \mathcal{S}} \sum_{\omega \in \Omega} p(\omega|\mathbf{x}, \tilde{\omega}) \end{aligned} ,$$

and we plug λ_{π_ω} back and solve for π_ω as follows:

$$\pi_\omega = \frac{\sum_{\tilde{\omega} \in \Omega} I(\tilde{\omega} = \omega) \sum_{\mathbf{x} \in \mathcal{S}} p(\omega|\mathbf{x}, \tilde{\omega})}{\sum_{\tilde{\omega} \in \Omega} I(\tilde{\omega} = \omega) \sum_{\mathbf{x} \in \mathcal{S}} \sum_{\omega \in \Omega} p(\omega|\mathbf{x}, \tilde{\omega})}. \tag{14}$$

Predicting posterior probability: To predict an unlabeled instance \mathbf{x}_u , the posterior probability of each class is calculated as follows:

$$\begin{aligned} p(\omega|\mathbf{x}_u) &= \frac{p(\mathbf{x}_u|\omega, \theta_\omega) \pi_\omega}{\sum_{\omega \in \Omega} p(\mathbf{x}_u|\omega, \theta_\omega) \pi_\omega} \\ &= \frac{\pi_\omega \sum_{m \in \mathcal{M}} w_{\omega,m} g(\mathbf{x}|\omega, \boldsymbol{\mu}_{\omega,m}, \boldsymbol{\Sigma}_{\omega,m})}{\sum_{\omega \in \Omega} \pi_\omega \sum_{m \in \mathcal{M}} w_{\omega,m} g(\mathbf{x}|\omega, \boldsymbol{\mu}_{\omega,m}, \boldsymbol{\Sigma}_{\omega,m})} \end{aligned}$$

The maximum class posterior probability decides the class of this unlabeled instance.

For convenience, we summarize the overall updating process in Algorithm 1

Algorithm 1 GMDA

Input: $\Theta = \{\theta_\omega\}_{\omega \in \Omega}$, $\Gamma = \{\gamma_{\tilde{\omega}, \omega}\}_{\tilde{\omega} \in \Omega}$, and $\Pi = \{\pi_\omega\}_{\omega \in \Omega}$.

Initialize: $w_{\omega, m}, \mu_{\omega, m}, \Sigma_{\omega, m}$ is obtained by employing k -means method, and the initialization value of π_ω is:

$$\pi_\omega = \frac{\sum_{\tilde{\omega} \in \Omega} I(\tilde{\omega})}{|\Omega|}.$$

if iter < itermax **do**

$p(\omega = k | \mathbf{x}_n, \tilde{\omega} = j)$ and $\Pr(m | \mathbf{x}_n, \omega_n = k)$ are calculated according to (8) and (9).

$\mu_{\omega, m}, \Sigma_{\omega, m}, w_{\omega, m}, \gamma_{\tilde{\omega}, \omega}, \pi_\omega$ is updated according to (10)–(14).

end

Output: Θ, Γ , and Π .

2 Convergence analysis

2.1 General solution

The log-likelihood function for the general decision problem with randomly substituted labels is given by

$$\begin{aligned} L_0 &= \frac{1}{|\mathcal{S}|} \sum_{\mathbf{x} \in \mathcal{S}} \log \tilde{p}(\mathbf{x} | \psi(\mathbf{x})) \pi(\psi(\mathbf{x})) \\ &= \frac{1}{|\mathcal{S}|} \sum_{\mathbf{x} \in \mathcal{S}} \log \sum_{\omega \in \Omega} [p(\mathbf{x} | \omega) p(\omega | \psi(\mathbf{x})) \pi(\psi(\mathbf{x}))]. \end{aligned} \tag{15}$$

If we denote $\mathcal{S}_{\tilde{\omega}}$ the training set of data vectors with the observed label $\tilde{\omega}$:

$$\begin{aligned} \mathcal{S}_{\tilde{\omega}} &= \{\mathbf{x} \in \mathcal{S} \ominus : \psi(\mathbf{x}) = \tilde{\omega}\}, \tilde{\omega} \in \Omega, \mathcal{S} \\ &= \bigcup_{\tilde{\omega} \in \Omega} \mathcal{S}_{\tilde{\omega}}, |\mathcal{S}| = \sum_{\tilde{\omega} \in \Omega} |\mathcal{S}_{\tilde{\omega}}|, \end{aligned}$$

then, using the relation $\tilde{\omega} = \psi(\mathbf{x})$, we can express the log-likelihood function (15) equivalently in the form

$$L_0 = \frac{1}{|\mathcal{S}|} \sum_{\tilde{\omega} \in \Omega} \sum_{\mathbf{x} \in \mathcal{S}_{\tilde{\omega}}} \left[\log \left(\sum_{\omega \in \Omega} p(\mathbf{x} | \omega) p(\omega | \tilde{\omega}) \right) + \log \pi(\tilde{\omega}) \right],$$

and further

$$L_0 = \sum_{\tilde{\omega} \in \Omega} \frac{|\mathcal{S}_{\tilde{\omega}}|}{|\mathcal{S}|} \log \pi(\tilde{\omega}) + \frac{1}{|\mathcal{S}|} \sum_{\mathbf{x} \in \mathcal{S}} \log [p(\mathbf{x} | \omega) p(\omega | \psi(\mathbf{x}))]. \tag{16}$$

We recall that for any two probability distributions $p(\omega)$ and $\pi(\omega)$, the well-known Kullback–Leibler information divergence is nonnegative:

$$I(p, \pi) = \sum_{\omega \in \Omega} p(\omega) \log \frac{p(\omega)}{\pi(\omega)} \geq 0,$$

and the inequality (11) can be rewritten in the form

$$\sum_{\omega \in \Omega} p(\omega) \log \pi(\omega) \leq \sum_{\omega \in \Omega} p(\omega) \log p(\omega). \tag{17}$$

Consequently, the left-hand part of the inequality (17) is maximized by setting $\pi(\omega) = p(\omega)$, $\omega \in \Omega$. For the same reason, the first sum in (16) is maximized by the maximum likelihood estimate $\pi(\tilde{\omega}) = |\mathcal{S}_{\tilde{\omega}}|/|\mathcal{S}|$, $\tilde{\omega} \in \Omega$ in terms of relative frequencies. In the following, we repeatedly make use of this practically useful consequence of the inequality (17).

The second term in Convergence analysis L_0 can be maximized by EM algorithm. If we define the conditional weights

$$\begin{aligned} q(\omega | \mathbf{x}, \psi(\mathbf{x})) &= \frac{p(\mathbf{x} | \omega) p(\omega | \psi(\mathbf{x}))}{\sum_{\omega \in \Omega} p(\mathbf{x} | \omega) p(\omega | \psi(\mathbf{x}))} \\ \omega \in \Omega, \sum_{\omega \in \Omega} q(\omega | \mathbf{x}, \psi(\mathbf{x})) &= 1, \mathbf{x} \in \mathcal{S}, \end{aligned} \tag{18}$$

then the second part of the log-likelihood function (16)

$$L = \frac{1}{|\mathcal{S}|} \sum_{\mathbf{x} \in \mathcal{S}} \log \left[\sum_{\omega \in \Omega} p(\mathbf{x} | \omega) p(\omega | \psi(\mathbf{x})) \right] \tag{19}$$

can be expanded in the form [25, 26]:

$$\begin{aligned} L &= \sum_{\omega \in \Omega} \frac{1}{|\mathcal{S}|} \sum_{\mathbf{x} \in \mathcal{S}} \\ &\{ q(\omega | \mathbf{x}, \psi(\mathbf{x})) \log [p(\mathbf{x} | \omega) p(\omega | \psi(\mathbf{x}))] \\ &\quad - q(\omega | \mathbf{x}, \psi(\mathbf{x})) \log q(\omega | \mathbf{x}, \psi(\mathbf{x})) \} \end{aligned} \tag{20}$$

Similarly, denoting by apostrophe the mixture components and mixture parameters in the next iteration of EM algorithm, we can write

$$\begin{aligned} L' &= \frac{1}{|\mathcal{S}|} \sum_{\mathbf{x} \in \mathcal{S}} \log \left[\sum_{\omega \in \Omega} p'(\mathbf{x} | \omega) p'(\omega | \psi(\mathbf{x})) \right] \\ &= \sum_{\omega \in \Omega} \frac{1}{|\mathcal{S}|} \sum_{\mathbf{x} \in \mathcal{S}} \{ q(\omega | \mathbf{x}, \psi(\mathbf{x})) \log [p'(\mathbf{x} | \omega) p'(\omega | \psi(\mathbf{x}))] \\ &\quad - q(\omega | \mathbf{x}, \psi(\mathbf{x})) \log q'(\omega | \mathbf{x}, \psi(\mathbf{x})) \} \end{aligned} \tag{21}$$

where

$$q'(\omega | \mathbf{x}, \psi(\mathbf{x})) = \frac{p'(\mathbf{x} | \omega) p'(\omega | \psi(\mathbf{x}))}{\sum_{\omega \in \Omega} p'(\mathbf{x} | \omega) p'(\omega | \psi(\mathbf{x}))}, \omega \in \Omega, \mathbf{x} \in \mathcal{S} \ominus \tag{22}$$

Now, the increment of EM algorithm in one iteration can be expressed as follows:

$$L' - L = \sum_{\omega \in \Omega} \frac{1}{|\mathcal{S}|} \sum_{\mathbf{x} \in \mathcal{S}} q(\omega|\mathbf{x}, \psi(\mathbf{x})) \log \left[\frac{p'(\mathbf{x}|\omega)p'(\omega|\psi(\mathbf{x}))}{p(\mathbf{x}|\omega)p(\omega|\psi(\mathbf{x}))} \right] + \frac{1}{|\mathcal{S}|} \sum_{\mathbf{x} \in \mathcal{S}} \sum_{\omega \in \Omega} q(\omega|\mathbf{x}, \psi(\mathbf{x})) \log \frac{q(\omega|\mathbf{x}, \psi(\mathbf{x}))}{q'(\omega|\mathbf{x}, \psi(\mathbf{x}))}. \tag{23}$$

The last sum in Eq. (23) is again the well-known non-negative Kullback–Leibler information divergence:

$$I(q, q') = \sum_{\omega \in \Omega} q(\omega|\mathbf{x}, \psi(\mathbf{x})) \log \frac{q(\omega|\mathbf{x}, \psi(\mathbf{x}))}{q'(\omega|\mathbf{x}, \psi(\mathbf{x}))} \geq 0, \tag{24}$$

and therefore, we can write

$$L' - L \geq \frac{1}{|\mathcal{S}|} \sum_{\mathbf{x} \in \mathcal{S}} \sum_{\omega \in \Omega} q(\omega|\mathbf{x}, \psi(\mathbf{x})) \log \left[\frac{p'(\mathbf{x}|\omega)p'(\omega|\psi(\mathbf{x}))}{p(\mathbf{x}|\omega)p(\omega|\psi(\mathbf{x}))} \right]. \tag{25}$$

Thus, for the sake of the monotonic property of EM algorithm, we have to guarantee the inequality

$$L' - L \geq \sum_{\omega \in \Omega} \frac{1}{|\mathcal{S}|} \sum_{\mathbf{x} \in \mathcal{S}} \left[q(\omega|\mathbf{x}, \psi(\mathbf{x})) \log \frac{p'(\mathbf{x}|\omega)}{p(\mathbf{x}|\omega)} \right] + \sum_{\omega \in \Omega} \frac{1}{|\mathcal{S}|} \sum_{\mathbf{x} \in \mathcal{S}} \left[q(\omega|\mathbf{x}, \psi(\mathbf{x})) \log \frac{p'(\omega|\psi(\mathbf{x}))}{p(\omega|\psi(\mathbf{x}))} \right] \geq 0. \tag{26}$$

Here, the sum over $\mathbf{x} \in \mathcal{S}$ in the second term can be decomposed into summing over $\mathbf{x} \in \mathcal{S}_{\tilde{\omega}}$:

$$L' - L = \sum_{\omega \in \Omega} \frac{1}{|\mathcal{S}|} \sum_{\mathbf{x} \in \mathcal{S}} \left[q(\omega|\mathbf{x}, \psi(\mathbf{x})) \log \frac{p'(\mathbf{x}|\omega)}{p(\mathbf{x}|\omega)} \right] + \sum_{\tilde{\omega} \in \Omega} \frac{|\mathcal{S}_{\tilde{\omega}}|}{|\mathcal{S}|} \sum_{\omega \in \Omega} \left[\frac{1}{|\mathcal{S}_{\tilde{\omega}}|} \sum_{\mathbf{x} \in \mathcal{S}_{\tilde{\omega}}} q(\omega|\mathbf{x}, \tilde{\omega}) \right] \log \frac{p'(\omega|\tilde{\omega})}{p(\omega|\tilde{\omega})}. \tag{27}$$

Now, if we define the EM iteration equations in the form

$$p'(\omega|\tilde{\omega}) = \frac{1}{|\mathcal{S}_{\tilde{\omega}}|} \sum_{\mathbf{x} \in \mathcal{S}_{\tilde{\omega}}} q(\omega|\mathbf{x}, \tilde{\omega}), \quad \omega \in \Omega, \quad \tilde{\omega} \in \Omega \tag{28}$$

$$p'(\cdot|\omega) = \arg \max_{p(\cdot|\omega)} \left\{ \frac{1}{|\mathcal{S}|} \sum_{\mathbf{x} \in \mathcal{S}} q(\omega|\mathbf{x}, \psi(\mathbf{x})) \log p(\mathbf{x}|\omega) \right\} \quad \omega \in \Omega, \tag{29}$$

then the monotonic property of EM algorithm is guaranteed because, by substitution (28), the second term in (26) is nonnegative as a sum of nonnegative Kullback–Leibler information divergences:

$$\sum_{\omega \in \Omega} p'(\omega|\tilde{\omega}) \log \frac{p'(\omega|\tilde{\omega})}{p(\omega|\tilde{\omega})} \geq 0, \quad \tilde{\omega} \in \Omega, \tag{30}$$

and Eq. (29) implies the inequalities

$$\frac{1}{|\mathcal{S}|} \sum_{\mathbf{x} \in \mathcal{S}} q(\omega|\mathbf{x}, \psi(\mathbf{x})) \log p'(\mathbf{x}|\omega) \geq \frac{1}{|\mathcal{S}|} \sum_{\mathbf{x} \in \mathcal{S}} q(\omega|\mathbf{x}, \psi(\mathbf{x})) \log p(\mathbf{x}|\omega), \quad \omega \in \Omega,$$

which can be rewritten in the form:

$$\frac{1}{|\mathcal{S}|} \sum_{\mathbf{x} \in \mathcal{S}} q(\omega|\mathbf{x}, \psi(\mathbf{x})) \log \frac{p'(\mathbf{x}|\omega)}{p(\mathbf{x}|\omega)} \geq 0, \quad \omega \in \Omega. \tag{31}$$

Consequently, the first term in (27) is nonnegative and in view of the above inequalities (30), (31), the EM iteration equations in the general form (23), (28), and (29) imply the basic monotonic property of EM algorithm [25, 26].

2.1.1 Gaussian Classes with Noisy Labels

Assuming a particular type, e.g., Gaussian class-conditional densities

$$p(\mathbf{x}|\omega) = f(\mathbf{x}|\boldsymbol{\mu}_\omega, \boldsymbol{\Sigma}_\omega), \quad \omega \in \Omega, \tag{32}$$

we can write Eq. (29) in a more specific form

$$\{\boldsymbol{\mu}'_\omega, \boldsymbol{\Sigma}'_\omega\} = \arg \max_{\{\boldsymbol{\mu}_\omega, \boldsymbol{\Sigma}_\omega\}} \left[\frac{1}{|\mathcal{S}|} \sum_{\mathbf{x} \in \mathcal{S}} q(\omega|\mathbf{x}, \psi(\mathbf{x})) \log f(\mathbf{x}|\boldsymbol{\mu}_\omega, \boldsymbol{\Sigma}_\omega) \right] \quad \omega \in \Omega. \tag{33}$$

As the maximized expression in Eq. (33) is a weighted likelihood function, we can easily verify [26] that the explicit solution can be expressed as a weighted analogy of the standard maximum likelihood estimate. In particular, let $F(\mathbf{x}|\boldsymbol{\mu})$ be a probability density with a parameter $\boldsymbol{\mu}$ having a standard maximum likelihood estimate $\hat{\boldsymbol{\mu}}$:

$$L_{\boldsymbol{\mu}} = \frac{1}{|\mathcal{S}|} \sum_{\mathbf{x} \in \mathcal{S}} \log f(\mathbf{x}|\boldsymbol{\mu}) \rightarrow \max \Rightarrow \hat{\boldsymbol{\mu}} = \frac{1}{|\mathcal{S}|} \sum_{\mathbf{x} \in \mathcal{S}} \mathbf{x}. \tag{34}$$

If $N(\mathbf{x})$ is the number of repeated occurrences of $\mathbf{x} \in \mathcal{X}$ in \mathcal{S} and $q(\mathbf{x})$ denotes the relative frequency of $\mathbf{x} \in \mathcal{S}$:

$$q(\mathbf{x}) = \frac{N(\mathbf{x})}{|\mathcal{S}|}, \quad \sum_{\mathbf{x} \in \mathcal{X}} q(\mathbf{x}) = 1, \quad (\mathbf{x} \notin \mathcal{S} \Rightarrow q(\mathbf{x}) = 0),$$

then Eq. (34) can be rewritten equivalently in the form

$$L_{\boldsymbol{\mu}} = \sum_{\mathbf{x} \in \mathcal{X}} q(\mathbf{x}) \log F(\mathbf{x}|\boldsymbol{\mu}) \rightarrow \max \Rightarrow \hat{\boldsymbol{\mu}} = \sum_{\mathbf{x} \in \mathcal{X}} q(\mathbf{x}) \mathbf{x}. \tag{35}$$

From the comparison of Eqs. (34) and (35), it follows that the weighted likelihood (35) is maximized by the

corresponding weighted maximum likelihood estimate (for a detailed proof in [26]). Consequently, in view of (33), we can write:

$$\boldsymbol{\mu}'_{\omega} = \frac{1}{\sum_{\mathbf{x} \in \mathcal{S}} q(\omega|\mathbf{x}, \psi(\mathbf{x}))} \sum_{\mathbf{x} \in \mathcal{S}} q(\omega|\mathbf{x}, \psi(\mathbf{x})) \mathbf{x}, \quad \omega \in \Omega \tag{36}$$

$$\boldsymbol{\Sigma}'_{\omega} = \frac{1}{\sum_{\mathbf{x} \in \mathcal{S}} q(\omega|\mathbf{x}, \psi(\mathbf{x}))} \sum_{\mathbf{x} \in \mathcal{S}} q(\omega|\mathbf{x}, \psi(\mathbf{x})) \mathbf{x}\mathbf{x}^T - \boldsymbol{\mu}'_{\omega} \boldsymbol{\mu}'_{\omega}{}^T, \quad \omega \in \Omega. \tag{37}$$

We can conclude that the problem of parameter estimation for the Gaussian classes with noisy labels can be solved by repeating the EM iteration Eqs. (18), (28), (36), and (37).

2.1.2 Class-conditional Gaussian Mixtures with Noisy Labels

The Gaussian assumption (32) is well known to be rather restrictive and can be essentially relaxed by approximating the unknown class-conditional densities $p(\mathbf{x}|\omega)$ by Gaussian mixtures. In particular, we assume

$$p(\mathbf{x}|\omega) = \sum_{m \in \mathcal{M}_{\omega}} w_{m\omega} f(\mathbf{x}|\boldsymbol{\mu}_{m\omega}, \boldsymbol{\Sigma}_{m\omega}), \tag{38}$$

$$\sum_{m \in \mathcal{M}_{\omega}} w_{m\omega} = 1, \quad \omega \in \Omega$$

where $m \in \mathcal{M}_{\omega}$ denotes the component’s index set of the class-conditional mixture $P(\mathbf{x}|\omega)$. Making substitution (38) in (19), we obtain the log-likelihood function in the following more general form:

$$L = \frac{1}{|\mathcal{S}|} \sum_{\mathbf{x} \in \mathcal{S}} \log \left[\sum_{\omega \in \Omega} \sum_{m \in \mathcal{M}_{\omega}} p(\omega|\psi(\mathbf{x})) w_{m\omega} f(\mathbf{x}|\boldsymbol{\mu}_{m\omega}, \boldsymbol{\Sigma}_{m\omega}) \right].$$

If we introduce the conditional component weights:

$$h(m, \omega|\mathbf{x}, \psi(\mathbf{x})) = \frac{p(\omega|\psi(\mathbf{x})) w_{m\omega} f(\mathbf{x}|\boldsymbol{\mu}_{m\omega}, \boldsymbol{\Sigma}_{m\omega})}{\sum_{\omega \in \Omega} \sum_{m \in \mathcal{M}_{\omega}} p(\omega|\psi(\mathbf{x})) w_{m\omega} f(\mathbf{x}|\boldsymbol{\mu}_{m\omega}, \boldsymbol{\Sigma}_{m\omega})}, \quad m \in \mathcal{M}_{\omega}, \omega \in \Omega, \mathbf{x} \in \mathcal{S} \tag{39}$$

then, in analogy with (20), we can expand the log-likelihood criterion (36) in the form:

$$L = \frac{1}{|\mathcal{S}|} \sum_{\mathbf{x} \in \mathcal{S}} \sum_{\omega \in \Omega} \sum_{m \in \mathcal{M}_{\omega}} h(m, \omega|\mathbf{x}, \psi(\mathbf{x})) \cdot \log [p(\omega|\psi(\mathbf{x})) w_{m\omega} f(\mathbf{x}|\boldsymbol{\mu}_{m\omega}, \boldsymbol{\Sigma}_{m\omega})] - h(m, \omega|\mathbf{x}, \psi(\mathbf{x})) \log h(m, \omega|\mathbf{x}, \psi(\mathbf{x}))$$

s.t. $\sum_{\omega \in \Omega} \sum_{m \in \mathcal{M}_{\omega}} h(m, \omega|\mathbf{x}, \psi(\mathbf{x})) = 1.$

Again, in analogy with (17)–(25), we come to the inequality

$$L' - L \geq \frac{1}{|\mathcal{S}|} \sum_{\mathbf{x} \in \mathcal{S}} \sum_{\omega \in \Omega} \sum_{m \in \mathcal{M}_{\omega}} h(m, \omega|\mathbf{x}, \psi(\mathbf{x})) \log \left[\frac{p'(\omega|\psi(\mathbf{x})) w_{m\omega} f(\mathbf{x}|\boldsymbol{\mu}'_{m\omega}, \boldsymbol{\Sigma}'_{m\omega})}{p(\omega|\psi(\mathbf{x})) w_{m\omega} f(\mathbf{x}|\boldsymbol{\mu}_{m\omega}, \boldsymbol{\Sigma}_{m\omega})} \right] \geq 0 \tag{40}$$

to be guaranteed for the sake of the monotonic property of EM algorithm. The right-hand side of (40) can be satisfied separately in two parts:

$$\frac{1}{|\mathcal{S}|} \sum_{\mathbf{x} \in \mathcal{S}} \sum_{\omega \in \Omega} \sum_{m \in \mathcal{M}_{\omega}} h(m, \omega|\mathbf{x}, \psi(\mathbf{x})) \log \frac{f(\mathbf{x}|\boldsymbol{\mu}'_{m\omega}, \boldsymbol{\Sigma}'_{m\omega})}{f(\mathbf{x}|\boldsymbol{\mu}_{m\omega}, \boldsymbol{\Sigma}_{m\omega})} \geq 0, \tag{41}$$

$$\sum_{\omega \in \Omega} \frac{1}{|\mathcal{S}|} \sum_{\mathbf{x} \in \mathcal{S}} \sum_{m \in \mathcal{M}_{\omega}} h(m, \omega|\mathbf{x}, \psi(\mathbf{x})) \log \frac{p'(\omega|\psi(\mathbf{x})) w'_{m\omega}}{p(\omega|\psi(\mathbf{x})) w_{m\omega}} \geq 0. \tag{42}$$

The first inequality (41) is satisfied if we define the new parameters $\boldsymbol{\mu}'_{m\omega}, \boldsymbol{\Sigma}'_{m\omega}$ by Eq. (33):

$$\{\boldsymbol{\mu}'_{m\omega}, \boldsymbol{\Sigma}'_{m\omega}\} = \arg \max_{\{\boldsymbol{\mu}_{m\omega}, \boldsymbol{\Sigma}_{m\omega}\}} \left[\sum_{\mathbf{x} \in \mathcal{S}} h(m, \omega|\mathbf{x}, \psi(\mathbf{x})) \log f(\mathbf{x}|\boldsymbol{\mu}_{m\omega}, \boldsymbol{\Sigma}_{m\omega}) \right], \quad \omega \in \Omega, \tag{43}$$

Again, using the weighted likelihood analogy of the standard maximum likelihood estimate [25, 26], we can write the following explicit solution of Eq. (43):

$$\boldsymbol{\mu}'_{m\omega} = \frac{1}{\sum_{\mathbf{x} \in \mathcal{S}} h(m, \omega|\mathbf{x}, \psi(\mathbf{x}))} \sum_{\mathbf{x} \in \mathcal{S}} h(m, \omega|\mathbf{x}, \psi(\mathbf{x})) \mathbf{x}, \quad m \in \mathcal{M}_{\omega}, \omega \in \Omega \tag{44}$$

$$\boldsymbol{\Sigma}'_{m\omega} = \frac{1}{\sum_{\mathbf{x} \in \mathcal{S}} h(m, \omega|\mathbf{x}, \psi(\mathbf{x}))} \cdot \sum_{\mathbf{x} \in \mathcal{S}} h(m, \omega|\mathbf{x}, \psi(\mathbf{x})) \mathbf{x}\mathbf{x}^T - \boldsymbol{\mu}'_{m\omega} \boldsymbol{\mu}'_{m\omega}{}^T, \quad m \in \mathcal{M}_{\omega}, \omega \in \Omega \tag{45}$$

The inequality (42) can be further decomposed as follows:

$$\sum_{\tilde{\omega} \in \Omega} \frac{|\mathcal{S}_{\tilde{\omega}}|}{|\mathcal{S}|} \sum_{\omega \in \Omega} \left[\frac{1}{|\mathcal{S}_{\tilde{\omega}}|} \sum_{\mathbf{x} \in \mathcal{S}_{\tilde{\omega}}} \sum_{m \in \mathcal{M}_{\omega}} h(m, \omega|\mathbf{x}, \tilde{\omega}) \right] \log \frac{p'(\omega|\tilde{\omega})}{p(\omega|\tilde{\omega})} \geq 0, \tag{46}$$

$$\sum_{\omega \in \Omega} \sum_{m \in \mathcal{M}_{\omega}} \left[\frac{1}{|\mathcal{S}|} \sum_{\mathbf{x} \in \mathcal{S}} h(m|\omega, \mathbf{x}, \psi(\mathbf{x})) \right] \log \frac{w'_{m\omega}}{w_{m\omega}} \geq 0, \tag{47}$$

where

$$h(m|\omega, \mathbf{x}, \psi(\mathbf{x})) = \frac{h(m, \omega|\mathbf{x}, \psi(\mathbf{x}))}{\sum_{m \in \mathcal{M}_{\omega}} h(m, \omega|\mathbf{x}, \psi(\mathbf{x}))}.$$

Considering the inequality (17), we can write again the EM iteration equations for the parameters $p'(\omega|\tilde{\omega})$, $w'_{m\omega}$ in explicit form. In particular, the inequality (45) is satisfied if we set

$$p'(\omega|\tilde{\omega}) = \frac{1}{|\mathcal{S}_{\tilde{\omega}}|} \sum_{\mathbf{x} \in \mathcal{S}_{\tilde{\omega}}} \sum_{m \in \mathcal{M}_{\omega}} h(m, \omega|\mathbf{x}, \tilde{\omega}), \omega \in \Omega, \tilde{\omega} \in \Omega. \tag{48}$$

And the second inequality (47) is satisfied if we define the new component weights $w'_{m\omega}$ by equation:

$$\begin{aligned} w'_{m\omega} &= \frac{1}{|\mathcal{S}|} \sum_{\mathbf{x} \in \mathcal{S}} h(m|\omega, \mathbf{x}, \psi(\mathbf{x})) \\ &= \frac{1}{|\mathcal{S}|} \sum_{\mathbf{x} \in \mathcal{S}} \frac{h(m, \omega|\mathbf{x}, \psi(\mathbf{x}))}{\sum_{m \in \mathcal{M}_{\omega}} h(m, \omega|\mathbf{x}, \psi(\mathbf{x}))} \quad m \in \mathcal{M}_{\omega}, \omega \in \Omega, \end{aligned} \tag{49}$$

The EM algorithm for the problem of estimating class-conditional Gaussian mixtures with noisy labels can be summarized in terms of the iteration Eqs. (39), (44), (46), (48), and (49).

3 Related work

The problem of discriminant analysis has been studied by researchers from many disciplines, such as physical, biological and social sciences, cognitive science, psychology, engineering, and medicine [34]. Recently, the discriminant analysis with label noise has gained substantial research attention. Various solution strategies have been proposed to prevent a learning algorithm from overfitting the noisy data, and the robust classifiers with capability to diminish the effect of label noise to a certain extent have obtained varying levels of success [9–11, 35–40]. For instance, in [38], the emphasis functions that combine both sample errors and their proximity to the classification border are explored. Long and Servedio demonstrated in [11] that for a broad class of convex potential functions, any boosting algorithm was highly susceptible to random classification noise. They also emphasized that the result was unsuitable for non-convex potential function. In [39], a comprehensive empirical investigation using neural network algorithms to learn from imbalanced data with labeling errors was explored.

Lee and Liu transformed the learning problem with positive and unlabeled examples into a problem of learning with noise by labeling all unlabeled examples as negative and using logistic regression to learn from the weighting noisy examples [40]. In [41], based on consistency assurance that the label noise ultimately did not hinder the search for the optimal classifier of the noise-free sample,

the study proved that any surrogate loss function could be used for classification with noisy labels by using importance reweighting. [41] also showed that the noise rate that could be estimated was upper bounded by the conditional probability of the noisy sample. Bootkrajang and Kabán built a discriminative model by modeling class noise distributions and reinterpreted existing discriminative models from the class noise perspective [8]. They proved that the error of label-noise robust logistic regression was bounded and that label-noise robust logistic regression behaved in the same way as logistic regression when label noise did not exist or when the label flipping was symmetric. They also demonstrated that the weighting mechanism of label-noise robust logistic regression improved upon logistic regression with asymmetric label flipping. However, in [8], the loss function did not define the latent true label but defined the observed noisy label instead. Rooyen et al. proposed in [12] a convex classification calibrated unhinged loss and proved that it is robust under symmetric label noise. The loss further avoided minimization of any convex potential over a linear function class that could result in classification performance equivalent to random guessing. In [42], corruption problems that were classified as mutually contaminated distributions were considered, and authors argued that optimized balanced error on corrupted data was equivalently optimized as the binary label error on clean data.

Based on the boundary conditional class noise assumption, instead of modeling data generation or conditional class probability both for symmetric and asymmetric cases, Jun and Cai assumed that the class noise was distributed as an unnormalized Gaussian and an unnormalized Laplace centered on the linear class boundary and proposed Gaussian noise model and Laplace noise model, respectively [13]. They then further reinterpreted logistic regression and probit regression by using the proposed class noise probability.

Table 1 Characteristics of the datasets

Dataset	Characteristics		
	Samples	Dimensionality	Classes
Synth1	2000	30	5
Synth2	1000	2	2
Breast	106	9	6
Iris	150	4	3
Wine	178	13	3
Heart	267	22	2
Boston	506	13	2
Waveform	5000	21	3

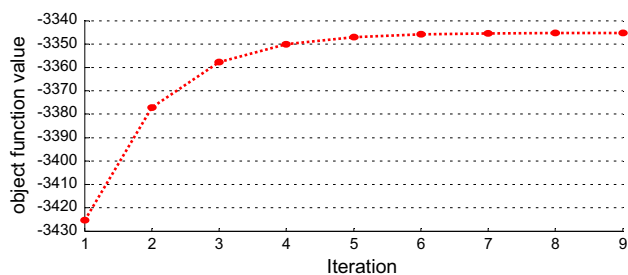


Fig. 1 Convergence of the proposed method using EM algorithm

Previous theoretical work on the label noise problem assumed that the two classes were separable, and the label noise was independent of the true class label or that the noise proportions for each class were known [14, 42] introduced a general mixture proportion estimation framework for classification with label noise that eliminated these assumptions. When the class-conditional distributions overlapped and the label noise was not symmetric, [14, 42] presented assumptions ensuring identifiability and the existence of a consistent estimator of the optimal risk and given associated estimation strategies. For any arbitrary pair of contaminated distributions, a unique pair of non-contaminated distributions satisfied the proposed assumptions. Scott argued in [15] that a solution to mixture proportion estimation led to solutions to various weakly supervised learning problems, such as anomaly detection, learning from positive and unlabeled examples, domain adaptation, and classification with label noise. He established a rate of convergence for mixture proportion estimation under an appropriate distributional assumption based on surrogate risk minimization and showed that this rate of convergence can derive the consistency of the algorithm and provide a practical implementation of mixture proportion estimation and demonstrate its efficacy in classification with noisy labels [15].

By modeling the corruption process through a Markov kernel and defining the corrupted learning problem to be the corrupted experiment, Brendan and Williamson developed a general framework for tackling corrupted learning problems as well as introduced minimax upper and lower bounds on the risk of learning in the presence of corruption [43].

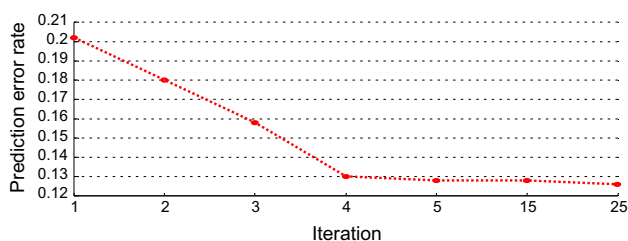


Fig. 2 Error rates at different iterations

Manwani and Sastry study in [44] noise tolerance under risk minimization. They assume that the actual training set given to the learning algorithm was obtained from the noise-free dataset, the class label of each example is corrupted and that a learning method was noise tolerant if the classifiers learned with noise-free data and with noisy data, and both have the same classification accuracy on the noise-free data. They showed that risk minimization under 0–1 loss function was a promising approach for learning from noisy training data and that Fisher linear discriminant and linear least squares under squared error loss were noise tolerant under uniform noise, but not under non-uniform noise. The risk minimization under other loss functions was not noise tolerant [16].

A great deal of research has been conducted on both theory and applications for such label noise problem. Despite much attention paid to discriminant analysis for noisy data [17], the investigation focused on the instances of generating a single Gaussian model. Furthermore, symmetric and asymmetric label noise was introduced to describe the contaminated distribution of corrupted binary labels. However, the instances that belonged to the same class usually were ruled by multiple GMM because of the presence of non-Gaussian distribution, which is mixed proportionally by Gaussian distribution of different means and variances. However, to the best of our knowledge, the discriminant analysis with noisy labels based on GMM has received limited research attention mainly because of mathematical difficulties. In particular, the commonly used approaches, such as matrix analysis, are no longer directly applicable to deal with both symmetric and asymmetric label noise problem because the presence of asymmetric label noise cannot be expressed in the normal form. In this paper, therefore, we intend to tackle such an important yet challenging problem. [45] presents a novel deep self-learning framework, which does not rely on any assumption on the distribution of the noisy labels and train a robust network on the real noisy datasets without extra supervision.

Similar to our approach, Bouveyron also proposed to use the explicit global mixture model of more than two classes [46]; however, Bouveyron's method is totally different from our approach. Bouveyron's approach compares the supervised information given by the learning data with an unsupervised modeling based on the Gaussian mixture model; if some learning data have wrong labels, the comparison of the supervised information with an unsupervised modeling of the data allows to detect the inconsistent labels. Then, it is possible afterward to build a supervised classifier by giving a low confidence to the learning observations with inconsistent labels.

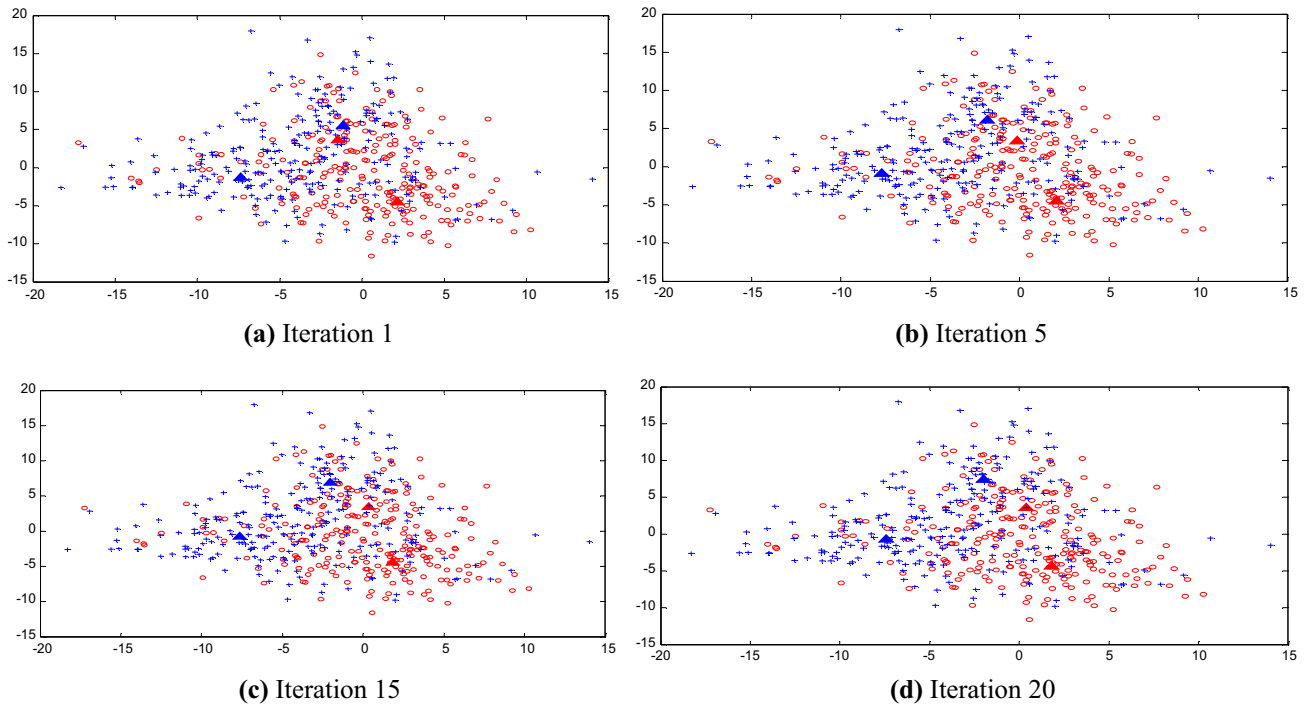


Fig. 3 Model’s mixture centers at different iterations

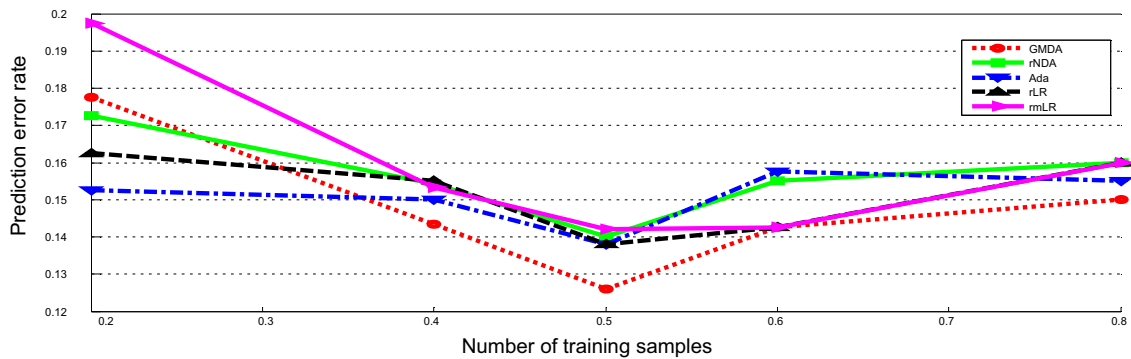


Fig. 4 Effect of the number of samples

4 Experiments and discussion

4.1 Datasets and preprocessing

Synthetic datasets and real-world datasets are used in our experiments. Table 1 presents a summary of the datasets.

Two synthetic datasets are created randomly by our MATLAB code. We apply the following real-world datasets: Boston, Breast Issue, SPECT Heart, Waveform, Wine, and Iris. Real-world datasets are UCI datasets [47].

The datasets are equally divided into training data and test data. The original class labels are treated as true labels. Symmetric and asymmetric label errors are injected into the datasets artificially. As the label noise was generated

randomly, the label noise rate and label error rate were not equal.

4.2 Results and discussion

First, we illustrate the convergence of object function value of the proposed method. We use the synth1 dataset with 20% label errors as a representative. The objective function value is almost stable after five iterations in Fig. 1. As shown in Fig. 2, the error rate of prediction after five iterations is 12.80% and 12.80% after ten iterations. The error rate changes rapidly during the first four iterations, and the error rate stabilizes after five iterations. The proposed method using the EM approach is convergent and

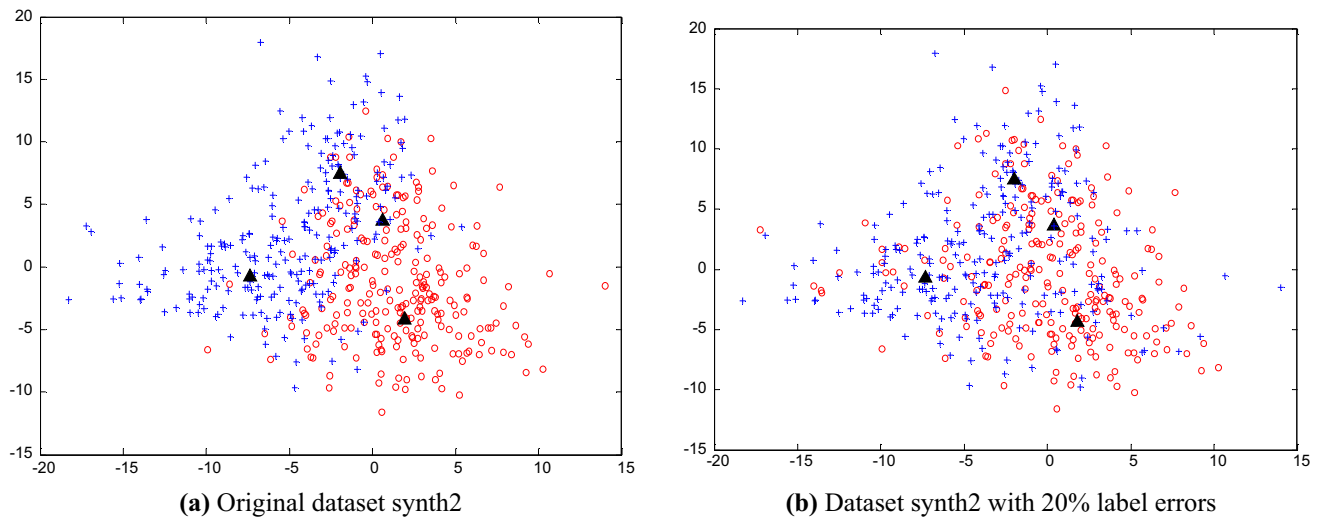


Fig. 5 Dataset synth2 and estimated mixture centers obtained by using the proposed method

Table 2 Results on dataset synth2 using the proposed method

Dataset	Results		
	Flipping probability	Class probability	Error rate (%)
Original synth2	$\begin{bmatrix} 1 & 2.49e-05 \\ 3.03e-04 & 1 \end{bmatrix}$	[0.4994 0.5006]	13.00
Synth2 with 20% label error	$\begin{bmatrix} 0.7825 & 0.2175 \\ 0.1977 & 0.8023 \end{bmatrix}$	[0.5101 0.4899]	12.60

effective. The change of the mixture models centers at different iterations is illustrated in Fig. 3.

We compare the proposed method to rNDA, AdaBoost, rLR, and rmLR. As the maximum likelihood used in our method is totally dependent on the training data, we change the training sample size from 20 to 80% to determine the effect of the size of the training data on the performance of classifiers. Figure 4 shows the effect of the number of training samples on error rates of the predictions of the methods. All the methods mentioned are affected by the number of training samples, and all methods perform best with 50% samples. We run the following experiments with 50% samples.

Figure 5 shows the original dataset synth2 and the estimated mixture centers obtained by our method in comparison with 20% label errors synth2 and its estimated mean vectors obtained by our method. Table 2 shows the parameters and the error rates of predictions obtained by our method using the original dataset synth2 and synth2 with 20% label errors. The diagonal elements $\gamma_{ij}(i = j)$ indicate the probability of labels that are not flipped. Table 2 shows that the unflipped probabilities of original dataset synth2 are extremely close to 1, whereas that of 20% label errors data are close to 0.8, and the real

probability of each class is 0.5. The estimated class probabilities are all close to 0.5. The results confirm the reality. The error rates of predictions using the two datasets mentioned are 13.00% and 12.60%, respectively, which differ slightly. In the noisy case, prediction is better than the original case because our method has already considered flipping.

Figure 6 shows the results of two synthetic datasets and six real-world datasets using the proposed method and the other four methods mentioned. All results are presented in Tables 3, 4, and 5. The trend is that at a higher label noise level, a higher prediction error rate is obtained. The AdaBoost method, which is a label-noise robust method, is affected most by the label errors and has mostly the largest error rate. The rLR and the rmLR methods have similar performance. The proposed method has a much better performance and the lowest error rate in most cases and outperforms others mostly both in the symmetric and asymmetric noise cases. The AdaBoost, rLR, and rmLR methods cannot predict when samples of one class label are all flipped in binary cases.

On the other hand, Fig. 6 also shows that the proposed method significantly outperforms other methods on synthetic datasets synth1 with larger dimension. We run

Table 3 Prediction error rate on two synthetic datasets and six real-world datasets with different symmetric label noise rates

Dataset	Method	Symmetric label noise rate					
		0	0.1	0.2	0.3	0.4	0.5
Synth1	Ada	0.03	0.056	0.061	0.078	0.116	0.224
	rNDA	0.035	0.064	0.064	0.065	0.065	0.065
	GMDA	0.005	0.003	0.003	0.003	0.030	0.003
Synth2	rmLR	0.142	0.138	0.138	0.184	0.504	0.504
	rLR	0.147	0.138	0.138	0.17	0.186	0.504
	Ada	0.142	0.14	0.142	0.184	0.226	0.318
	rNDA	0.138	0.138	0.138	0.266	0.524	0.532
	GMDA	0.128	0.120	0.126	0.166	0.19	0.182
	Issue	Ada	0.396	0.333	0.381	0.523	0.571
Breast	rNDA	0.305	0.404	0.357	0.5	0.504	0.542
	GMDA	0.377	0.238	0.357	0.357	0.500	0.504
	Iris	Ada	0.026	0.05	0.183	0.200	0.333
Iris	rNDA	0.026	0.033	0.033	0.05	0.100	0.08
	GMDA	0.013	0.016	0.033	0.05	0.083	0.08
	Wine	Ada	0.078	0.123	0.112	0.157	0.236
rNDA		0.044	0.011	0.044	0.044	0.044	0.078
GMDA		0.033	0.022	0.044	0.033	0.045	0.076
Heart	rmLR	0.548	0.654	0.691	0.737	0.6	0.5
	rLR	0.533	0.635	0.672	0.7	0.65	0.6
	Ada	0.270	0.261	0.299	0.537	0.32	0.537
	rNDA	0.609	0.626	0.607	0.637	0.39	0.7
	GMDA	0.248	0.261	0.261	0.289	0.287	0.271
	Boston	rmLR	0.130	0.138	0.128	0.193	0.336
rLR		0.130	0.183	0.113	0.188	0.316	0.638
Ada		0.17	0.158	0.163	0.183	0.292	0.549
rNDA		0.233	0.198	0.198	0.198	0.445	0.717
GMDA		0.205	0.183	0.188	0.183	0.188	0.185
Wave form		Ada	0.188	0.201	0.223	0.223	0.290
	rNDA	0.293	0.248	0.249	0.251	0.252	0.253
	GMDA	0.222	0.228	0.231	0.232	0.244	0.244

The best performances are indicated in the bold

Table 4 Prediction error rate on two synthetic datasets and six real-world datasets with different asymmetric label noise rates

Dataset	Method	Asymmetric label noise rate						
		0	0.1	0.2	0.3	0.4	0.5	
Synth1	Ada	0.03	0.073	0.085	0.138	0.211	0.221	
	rNDA	0.035	0.035	0.067	0.067	0.065	0.79	
	GMDA	0.005	0.005	0.005	0.005	0.026	0.191	
Synth2	rmLR	0.142	0.138	0.148	0.138	0.192	–	
	rLR	0.147	0.138	0.142	0.134	0.23	–	
	Ada	0.142	0.142	0.296	0.468	0.494	–	
	rNDA	0.138	0.128	0.144	0.144	0.148	0.504	
	GMDA	0.128	0.128	0.132	0.13	0.144	0.52	
	Breast	Ada	0.396	0.333	0.357	0.343	0.452	0.509
Issue		rNDA	0.305	0.238	0.285	0.437	0.523	0.5
GMDA		0.377	0.261	0.214	0.25	0.404	0.415	
Iris	Ada	0.026	0.016	0.183	0.177	0.433	–	
	rNDA	0.026	0.016	0.016	0.022	0.033	–	
	GMDA	0.013	0.016	0.016	0.022	0.033	–	
Wine	Ada	0.078	0.140	0.112	0.197	0.408	–	
	rNDA	0.044	0.056	0.056	0.042	0.056	–	
	GMDA	0.033	0.042	0.042	0.042	0.056	–	
Heart	rmLR	0.548	0.448	0.336	0.355	0.700	–	
	rLR	0.533	0.420	0.411	0.336	0.672	–	
	Ada	0.270	0.299	0.307	0.392	0.719	–	
	rNDA	0.609	0.364	0.271	0.364	0.729	–	
	GMDA	0.248	0.261	0.261	0.261	0.327	–	
	Boston	rmLR	0.130	0.163	0.173	0.178	0.198	0.336
rLR		0.130	0.163	0.183	0.173	0.193	0.347	
Ada		0.17	0.178	0.262	0.415	0.485	0.442	
rNDA		0.233	0.198	0.198	0.198	0.445	0.233	
GMDA		0.205	0.168	0.183	0.183	0.163	0.221	
Wave form		Ada	0.188	0.196	0.286	0.339	0.390	–
	rNDA	0.293	0.23	0.231	0.493	0.500	–	
	GMDA	0.222	0.227	0.226	0.280	0.296	–	

The best performances are indicated in the bold

another experiment to determine the effect of the dimension of the datasets. We create a set of datasets consisting of 1000 samples, three classes, and dimension from 5 to 30. Then, 20% label errors are artificially injected into these datasets.

Figure 7 shows the results on different datasets with different dimensions using different methods. The trend is that with a higher dimension, a lower error rate is obtained. The proposed method performs much better than the others in all dimensionality cases.

We employ a set of synthetic datasets to study the effect of the mixture number of the datasets and that of the

component number of the model. Each dataset contains 1000 samples with ten dimensions and comprises three classes, component numbers from 1 to 5 for each class. Figure 7 shows that the more components in each class the dataset has, the larger the error rate is. In almost all cases, the proposed method notably outperforms the other methods.

According to Fig. 6 and Tables 3, 4, and 5, our method achieves better performance in a multi-class case than in a two-class case. We run one more experiment to investigate the effect of the number of classes. Datasets used are created artificially as well; they all contain 1000 samples

Table 5 Win/draw/lose

Dataset	Method	Win/draw/lose	
		Symmetric	Asymmetric
Synth1	Ada	0/0/6	0/0/6
	rNDA	0/0/6	0/0/6
	GMDA	6/0/0	6/0/0
Synth2	rmLR	0/0/6	0/0/6
	rLR	1/0/5	0/0/6
	Ada	0/0/6	0/0/6
	rNDA	0/0/6	1/1/4
Breast	GMDA	5/0/1	4/1/1
	Ada	0/0/6	0/0/6
Issue	rNDA	1/1/4	2/0/4
	GMDA	4/1/1	4/0/2
Iris	Ada	0/0/6	0/0/5
	rNDA	0/3/3	0/4/1
	GMDA	3/3/0	1/4/0
Wine	Ada	0/0/6	0/0/5
	rNDA	1/2/3	0/2/3
	GMDA	3/2/1	3/2/0
Heart	rmLR	0/0/6	0/0/5
	rLR	0/0/6	0/0/5
	Ada	0/1/5	0/0/5
	rNDA	0/0/6	0/0/5
Boston	GMDA	5/1/0	5/0/0
	rmLR	1/1/4	1/2/3
	rLR	1/1/4	1/2/3
	Ada	0/1/5	0/0/6
	rNDA	0/0/6	0/0/6
Wave form	GMDA	2/1/3	2/0/4
	Ada	4/0/2	2/0/3
	rNDA	0/0/6	0/0/5
	GMDA	2/0/4	3/0/2

with ten dimensions. Figure 8 shows that our proposed method is affected much less than the other methods.

Our previous experiments on synthetic datasets have been conducted on the premise of taking the correct mixture number. In this paper, we tested the datasets using an invalid mixture number. We take a three-component dataset as a representative and set the component number of the model from 1 to 6. From Figs. 9 and 10, we can discern that an invalid mixture number makes an increment on error rate, whereas rNDA and AdaBoost have error rates of 0.5040 and 0.4660, respectively. Our method still performs remarkably better even when an invalid mixture number is used. Poor performance is expected when the number of the component is 1.

The EM method we employed finds the local optimum each time; the initial cluster center is significant. The initial values used in this paper are obtained by k-means method, which is very sensitive to initial cluster centers. A bad initial cluster center leads to poor cluster performance and affects the error rate of the proposed method. Moreover, the difference between samples in the same class is much smaller than that between classes; thus, obtaining proper initial values for our method is difficult. The mixture model proposed in this paper is finite; the number of mixture components is provided in advance and cannot be changed to adapt to a simpler or more complicated situation. Thus, the estimation of the number of components and adapted mixture number remains to be studied.

5 Experimental results on large-scale datasets

To verify the performance of our proposed approach on large-scale datasets, we employ six synthetic datasets to study the effect of the mixture number of the datasets and that of the component number of the model. Each dataset contains 15,000 samples with 200 dimensions and comprises two classes, component numbers from 1 to 5 for each class. The six datasets together with their sizes N and number of features D are listed in Table 6. More specifically, first, we randomly generate six synthetic datasets for verification goal. The dataset is a mixture of two types of labels, with the covariance range from 0 to 250, which means that the correlation between these two types of labels is from uncorrelation to the maximum correlation. Similarly, for the division of datasets according to Table 6, we have carried out tenfold, fivefold, and threefold cross-validation, respectively. In addition, we add noise labels based on 0%, 10%, 20%, 30%, 40%, and 50% of the total number of labels. Table 7 shows the error rate with 5, 10, and 3 cross-validation for all comparison methods. In Table 7, (a1), (b1), and (c1) summarize the error rates on synthetic datasets with label correlation group, and (a2), (b2), and (c2) summarize the error rates on synthetic datasets with label uncorrelation group. Figure 11 shows the learning performances for all comparison methods; similarly, in Fig. 11, (a1), (b1), and (c1) are the learning performances on synthetic datasets with label correlation group, and (a2), (b2), and (c2) are the learning performances on synthetic datasets with label uncorrelation group.

From Table 7 and Fig. 11, we can see that:

- (a) The number of these two types of tags in synthetic datasets is comparable to each other. They belong to the synthetic datasets with relatively balanced class

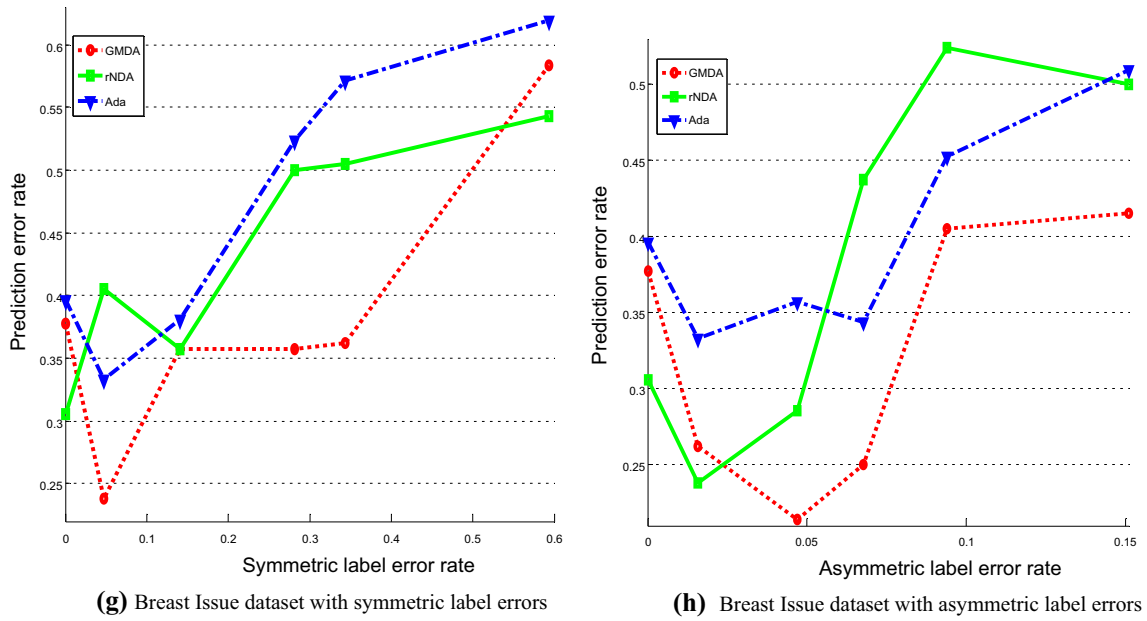


Fig. 6 continued

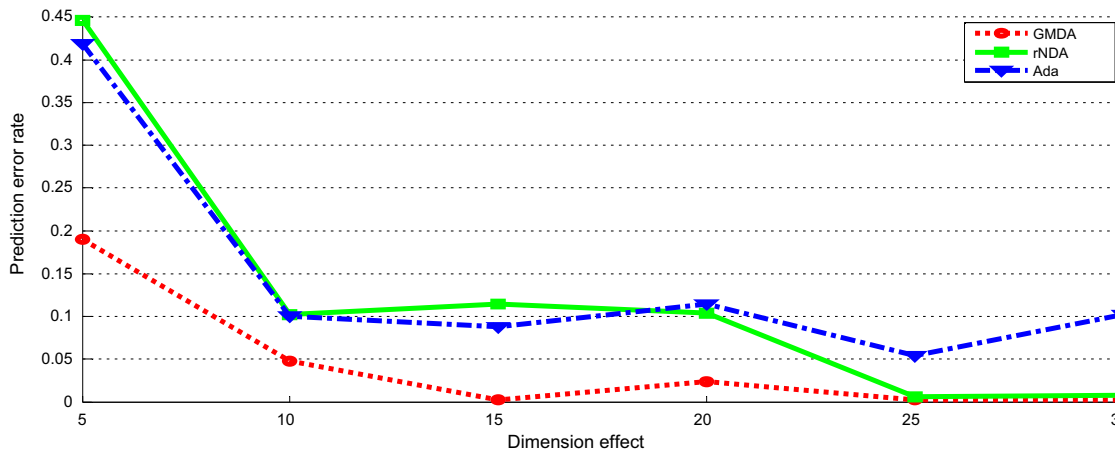


Fig. 7 Effect of the dimension of datasets

Fig. 8 Effect of the number of components of the model (the number of dataset mixtures is 3)

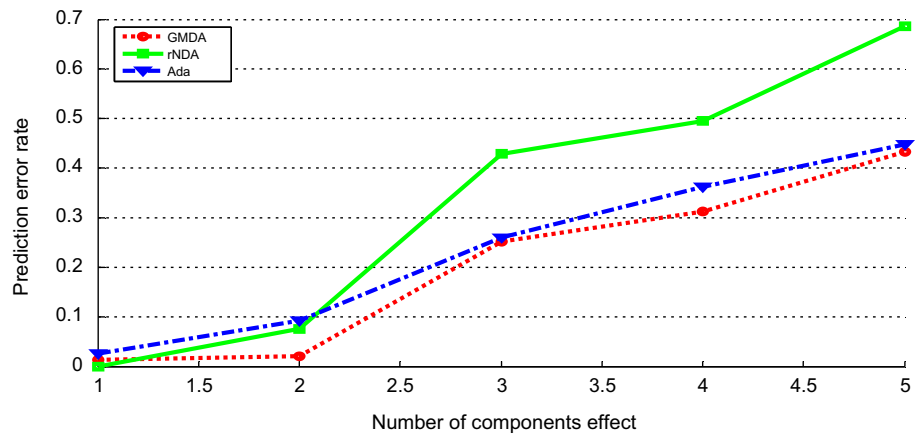


Fig. 9 Effect of the number of classes

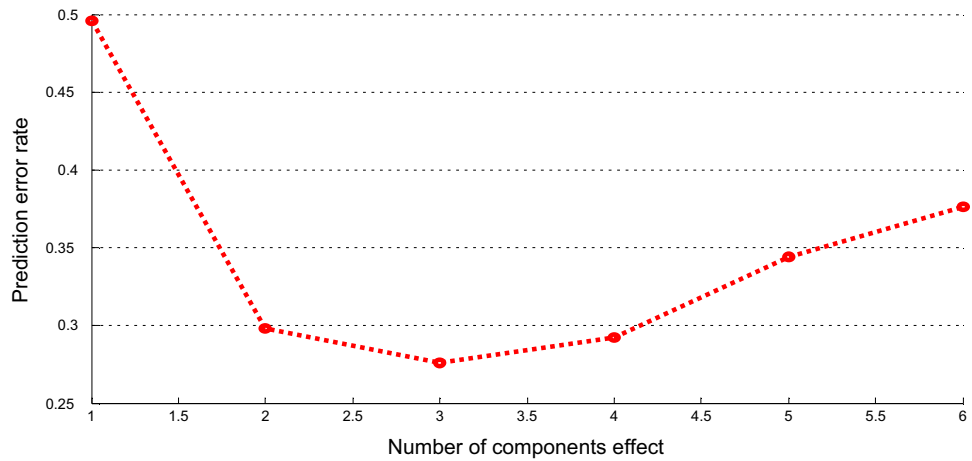


Fig. 10 Effect of the number of classes

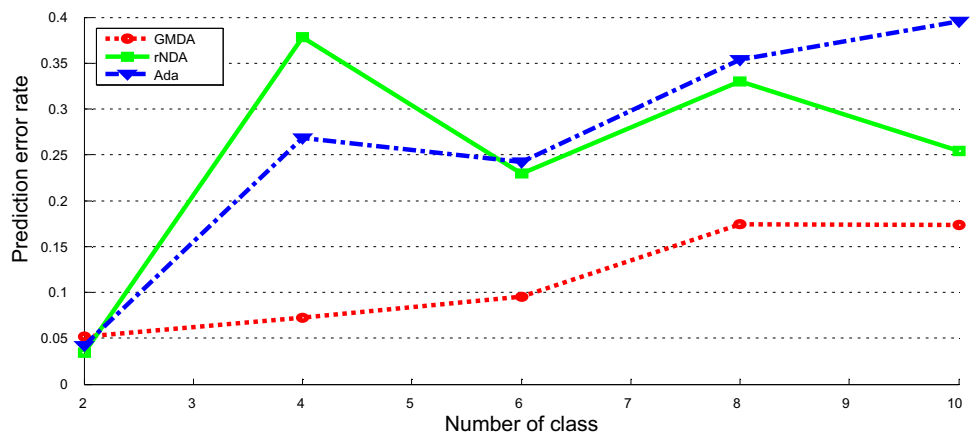


Table 6 Characteristics of the large-scale datasets

Dataset	Characteristics					
	N	D	Covariance	Cross-validation	Number of samples in class 1	Number of samples in class 2
Synth1	15,000	200	250	5	7592	7408
Synth2	15,000	200	0	5	6619	8381
Synth3	15,000	200	250	10	8434	6566
Synth4	15,000	200	0	10	6619	8381
Synth5	15,000	200	250	3	7707	7293
Synth6	15,000	200	0	3	7796	7204

size and large sample size, and it has certain representativeness.

- (b) When the characteristics difference between the two classes is small, that is, the covariance between two classes is equal to 250, the effect of the increase in noise labels is not very obvious. Compared with other methods, the GMDA has a certain capability of noise resistance, and its error rate is up and down at 50%. Secondly, with the increase in noise tags, the error rate of the GMDA decreases by about 1%.

Similar to the GMDA, under these synthetic datasets, the other comparison algorithms also fluctuate at specific values. We suspect that this is caused by the randomly generated synthetic datasets, and this small fluctuation does not affect the evaluation of the noise resistance performance of various models.

- (c) When the characteristics difference between the two classes is large, that is, the covariance between two classes is equal to 0, the addition of noise tags has different degrees of impact on these algorithms.

Table 7 Experimental results in error rate on six different correlated synthetic datasets with different cross-validation processes

Method	Label noise rate					
	0.0	0.1	0.2	0.3	0.4	0.5
<i>(a1) error rate with 5-cross-validation on correlated synth1 dataset</i>						
GMDA	0.5023	0.5023	0.5023	0.5023	0.5023	0.4977
rNDA	0.5063	0.5063	0.5063	0.5063	0.5063	0.4937
Ada	0.4953	0.4853	0.4997	0.5007	0.5090	0.5047
rLR	0.4927	0.4907	0.5057	0.5017	0.5110	0.5087
rmLR	0.4890	0.4877	0.4953	0.5080	0.5080	0.5090
<i>(b1) error rate with 10-cross-validation on correlated synth3 dataset</i>						
GMDA	0.5073	0.5073	0.4927	0.4927	0.4927	0.4927
rNDA	0.5067	0.5067	0.4933	0.4933	0.4933	0.4933
Ada	0.4520	0.4520	0.4520	0.4520	0.5480	0.5480
rLR	0.5033	0.5087	0.4973	0.4807	0.4867	0.4947
rmLR	0.5120	0.4987	0.4973	0.4740	0.4827	0.4967
<i>(c1) error rate with 3-cross-validation on correlated synth5 dataset</i>						
GMDA	0.5060	0.5060	0.5060	0.5060	0.5060	0.4938
rNDA	0.4944	0.5056	0.5056	0.5056	0.5056	0.5056
Ada	0.4812	0.4812	0.4812	0.4812	0.5188	0.5188
rLR	0.4838	0.4956	0.4884	0.4818	0.4934	0.5144
rmLR	0.4846	0.4908	0.4904	0.4950	0.4904	0.5152
<i>(a2) error rate with 5-cross-validation on uncorrelated synth2 dataset</i>						
GMDA	0.5570	0.5570	0.5570	0.5577	0.9640	0.5570
rNDA	0.4427	0.4427	0.4427	0.4427	0.4427	0.4427
Ada	0.4427	0.4427	0.4427	0.5573	0.5573	0.4427
rLR	0.1653	0.1103	0.3853	0.5573	0.9483	0.1653
rmLR	0.2730	0.2730	0.4303	0.5477	0.7920	0.2730
<i>(b2) error rate with 10-cross-validation on uncorrelated synth4 dataset</i>						
GMDA	0.0420	0.5440	0.5453	0.5440	0.5473	0.9613
rNDA	0.4560	0.4560	0.4560	0.4560	0.4560	0.4560
Ada	0.4560	0.4560	0.4560	0.4560	0.5440	0.5440
rLR	0.0647	0.2313	0.2320	0.2133	0.4267	0.9353
rmLR	0.0600	0.2313	0.3453	0.3733	0.4473	0.9400
<i>(c2) error rate with 3-cross-validation on uncorrelated synth6 dataset</i>						
GMDA	0.0424	0.5572	0.5572	0.5570	0.5673	0.9628
rNDA	0.4422	0.4422	0.4422	0.4422	0.4422	0.4422
Ada	0.4422	0.4422	0.4422	0.4422	0.5578	0.5578
rLR	0.0754	0.1754	0.1562	0.1902	0.4370	0.9264
rmLR	0.0708	0.2686	0.2376	0.3258	0.4577	0.9112

The best performances are indicated in the bold

Generally speaking, the more the noise tags are, the higher the classification error rate is. But when we look the label noise rate at the interval [0.2, 0.4], these algorithms all are with good anti-noise performance and have ability of active noise cancelation. It is worthy of note that ADA algorithm is different from the trend curve of other algorithms. When the difference between the two kinds of tags is small, the fluctuation is large. When the difference between the two kinds of tags is large, the fluctuation is small.

We can choose the appropriate algorithm according to the actual dataset.

In sum up, when the characteristics difference between the two classes in mixture model is obvious, it is meaningful to analyze and compare the experimental results. Whereas the correlation between the two classes in mixture model is large, the effect of the increase in noise labels is vague and limited. As increase in noise labels, prediction error rate does not change much.

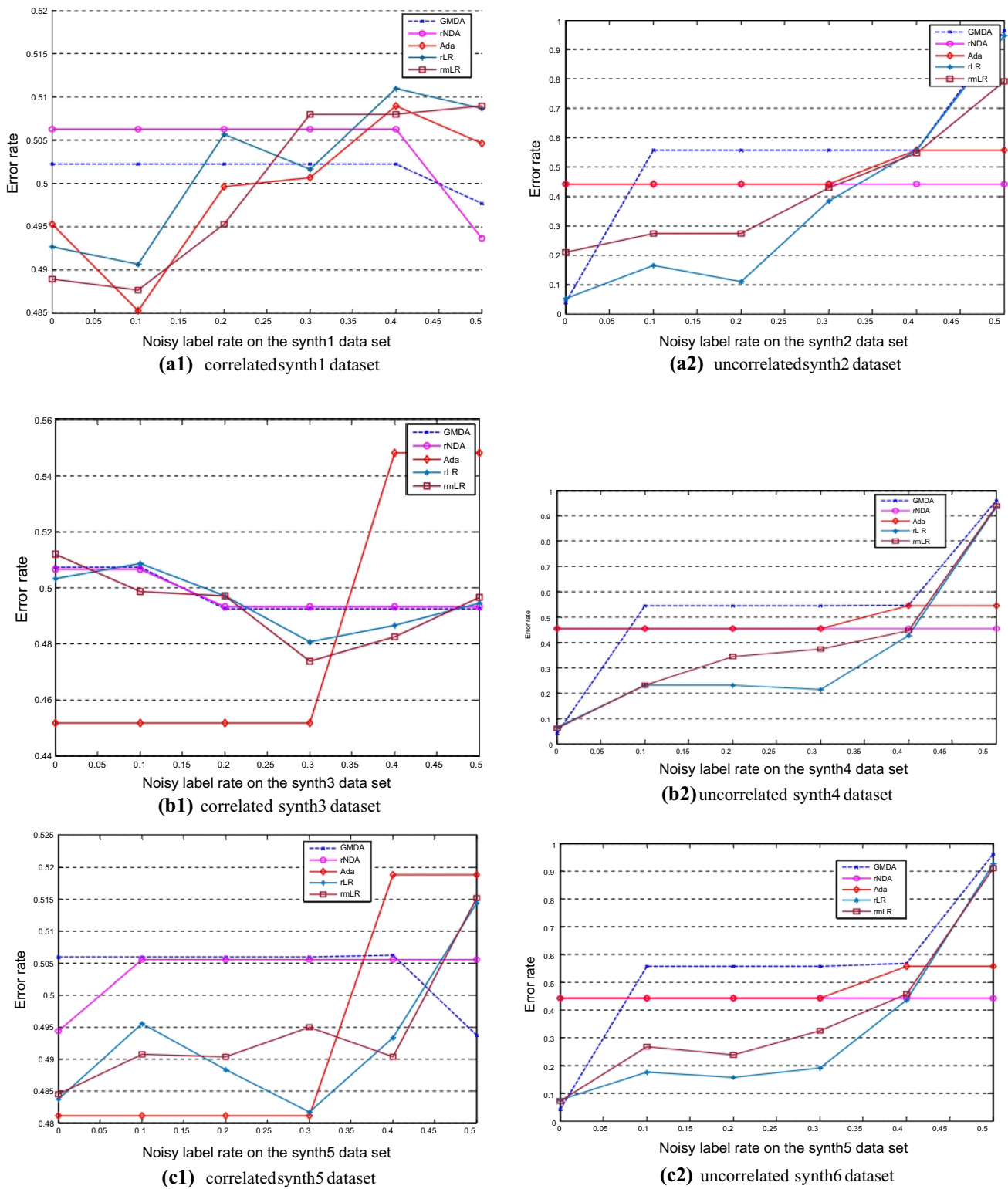


Fig. 11 Experimental results in error rate on six synthetic datasets with different cross-validation processes

6 Conclusion

This paper presented a discriminant analysis based on Gaussian mixture models and applied to classification in the presence of label noise. We derived the updating formulas of the parameters. The experiments on two synthetic datasets and several real-world datasets showed that the proposed method was convergent and effective and mostly outperformed the other methods. Compared with the other methods, our method was less affected by the factors discussed in the preceding sections.

We found that the number of training samples affected the performance significantly, that is, the number of training samples is increased if necessary. If the samples were insufficient for maximum likelihood to estimate, Bayes estimation was used, where prior information was utilized, or domain adaptation learning was used, where a source dataset that was akin to a target dataset was used to help.

The number of components of a model given in advance may not be adapted to all the classes; it might lead to further calculation on a simpler case or less approximation on a more complicated case. Therefore, we considered a more flexible and adaptable infinite mixture model that estimates the hidden number of components from the training datasets.

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

Compliance with ethical standards

Conflict of interest We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work, and there is no professional or other personal interest of any nature or kind in any product, service, and/or company that could be construed as influencing the position presented in, or the review of, the manuscript entitled “GMM Discriminant Analysis with Noisy Label for Each Class.”

References

1. ttleton DF, Orriols-Puig A, Fornells A (2010) A study of the effect of different types of noise on the precision of supervised learning techniques. *Artif Intell Rev*, 33(4): 275–306
2. Frenay B, Verleysen M (2014) Classification in the presence of label noise: a survey. *IEEE Transactions on Neural Networks and Learning Systems* 25(5):845–869
3. Brodley CE, Friedl MA (1999) Identifying mislabeled training data. *J Artif Intell Res* 11(1):131–167
4. Libralon GL, Carvalho ACPDLFD, Lorena AC (2009) Pre-processing for noise detection in gene expression classification data”. *J Brazil Comput Soc* 15(1):3–11
5. Abellán J, Masegosa AR (2010) Bagging decision trees on data sets with classification noise. In: *Proceedings of foundations of information and knowledge systems, 6th international symposium, FoIKS 2010, Sofia, Bulgaria, Springer*, pp 248–265
6. Mathews H, Mayyay V, Volfovsky A, Reeves G (2019) Gaussian mixture models for stochastic block models with non-vanishing noise. [arXiv:1911.01855v1,2019](https://arxiv.org/abs/1911.01855v1)
7. Hastie T, Tibshirani R, Friedman J (2001) *The elements of statistical learning*. Springer, New York
8. Bootkrajang J, Kabán A (2012) Label-noise robust logistic regression and its applications. In: *Proceedings machine learning and knowledge discovery in databases - European conference, ECML PKDD 2012, Bristol*, pp 143–158, 2012
9. Kearns Michael (1998) Efficient noise-tolerant learning from statistical queries. *J ACM (JACM)* 45(6):983–1006
10. Wessels Y, de Ridder LF, Reinders MJ (2007) Classification in the presence of class noise using a probabilistic kernel fisher method. *Pattern Recognit* 40(12):3349–3357
11. Long Philip M, Servedio Rocco A (2010) Random classification noise defeats all convex potential boosters. *Mach Learn* 78(3):287–304
12. van Rooyen B, Menon AK, Williamson RC (2015) Learning with symmetric label noise: the importance of being unhinged. *arXiv preprint arXiv:1505.07634*
13. Du J, Cai Z (2015) Modelling class noise with symmetric and asymmetric distributions. In: *Proceedings of the twenty-ninth AAAI conference on artificial intelligence, Austin, Texas, USA, AAAI Press*, pp 2589–2595
14. Scott C, Blanchard G, Handy G (2013) Classification with asymmetric label noise: consistency and maximal denoising. In: *Proceeding of the 26th annual conference on learning theory, Princeton University, NJ*, pp 489–511
15. Scott C, Blanchard G, Handy G, et al (2013) Classification with asymmetric label noise: consistency and maximal denoising. *arXiv preprint arXiv:1303.1208*
16. Manwani N, Sastry PS (2013) Noise tolerance under risk minimization. *IEEE Trans Cybern* 43(3):1146–1151
17. Goldman SA, Sloan RH (1995) Can PAC learning algorithms tolerate random attribute noise? *Algorithmica* 14(1):70–84
18. Zhang W, Rekaya R, Bertrand K (2006) A method for predicting disease subtypes in presence of misclassification among training samples using gene expression: application to human breast cancer. *Bioinformatics* 22(3):317–325
19. Abu Shanab A, Khoshgoftaar TM, Wald R (2012) Robustness of threshold-based feature rankers with data sampling on noisy and imbalanced data. In: *Proceedings of the twenty-fifth international florida artificial intelligence research society conference, Marco Island, Florida: AAAI Press*
20. Rätsch G, Onoda T, Müller K (2001) Soft margins for AdaBoost. *Mach Learn* 42(3):287–320
21. Wilson DR, Martinez TR (2000) Reduction techniques for instance-based learning algorithms. *Mach Learn* 38(3):257–286
22. Brodley CE, Friedl MA (1996) Identifying and eliminating mislabeled training instances. In: *Proceedings of the thirteenth national conference on artificial intelligence and eighth innovative applications of artificial intelligence conference, AAAI 96, Portland, Oregon: AAAI Press*, pp. 799–805
23. Bootkrajang J (2013) *Supervised learning with random labelling errors*. University of Birmingham, Birmingham
24. Moon TK (1996) The expectation-maximization algorithm. *IEEE Signal Process Mag* 13(6):47–60
25. Dempster AP, Laird NM, Rubin DB (1977) Maximum Likelihood from Incomplete Data via the EM Algorithm. *J R Stat Soc Ser B* 39(1):1–38
26. Jiří G (1982) On numerical evaluation of maximum-likelihood estimates for finite mixtures of distributions. *Kybernetika (Prague)* 18(3):173–190

27. Lathuiliere S, Mesejo P, Alameda-Pineda X, Horaud R (2018) DeepGUM: learning deep robust regression with a gaussian-uniform mixture model. [arXiv:1808.09211v1](https://arxiv.org/abs/1808.09211v1)
28. Anonymous authors (2020) DIVIDEMIX: learning with noisy labels as semi-supervised learning. Under review as a conference paper at ICLR 2020
29. Kim Y, Yim J, Yun J, Kim J (2019) NLNL: negative learning for noisy labels. [arXiv:1908.07387v1](https://arxiv.org/abs/1908.07387v1), 2019
30. Lawrence ND, Scholkopf B (2012) Estimating a kernel fisher discriminant in the presence of label noise. In: Proceedings of the 29th international conference on machine learning, ICML 2012, Edinburgh, Scotland, UK:Omnipress, 306–313,2012
31. Melnykov V, Melnykov I (2012) Initializing the EM algorithm in Gaussian mixture models with an unknown number of components. *Comput Stat Data Anal* 56(6):1381–1395
32. Dasgupta S (1999) Learning mixtures of Gaussians. In: Proceedings of 40th annual symposium on foundations of computer science, FOCS '99, New York, USA, IEEE, pp 634–644
33. Bishop CM (1995) *Neural networks for pattern recognition*. Oxford University Press, New York
34. Ji S, Ye J (2008) Generalized linear discriminant analysis: a unified framework and efficient model selection. *IEEE Trans. Neural Netw.* 19(10):1768–1782
35. Durrant RJ, Kaban A (2012) Error bounds for kernel Fisher linear discriminant in Gaussian Hilbert space. In: Proceedings of the fifteenth international conference on artificial intelligence and statistics, AISTATS 2012, La Palma, Canary Islands: JMLR.org, pp 337–345
36. Angluin Dana, Laird Philip (1988) Learning from noisy examples. *Mach Learn* 2(4):343–370
37. Natarajan N, Dhillon I, Ravikumar P, Tewari A (2013) Learning with noisy labels. In: Proceedings of 27th annual conference on neural information processing systems, Lake Tahoe, Nevada, United States, pp 1196–1204
38. Gómez-Verdejo Vanessa, Ortega-Moral Manuel, Arenas-García Jerónimo, Figueiras-Vidal Aníbal R (2006) Boosting by weighting critical and erroneous samples. *Neurocomputing* 69(7-9):679–685
39. Khoshgoftaar Taghi M, Hulse JV, Napolitano A (2010) Supervised neural network modeling: an empirical investigation into learning from imbalanced data with labeling errors. *IEEE Trans Neural Netw* 21(5):813–830
40. Lee WS, Liu B (2003) Learning with positive and unlabeled examples using weighted logistic regression. In: Proceedings of the twentieth international conference (ICML 2003), Washington, DC, USA, AAAI Press, pp 448–455
41. Liu T, Tao D (2014) Classification with noisy labels by importance reweighting. [arXiv preprint arXiv:1411.7718](https://arxiv.org/abs/1411.7718)
42. Menon A, Rooyen BV, Ong CS, et al (2015) Learning from corrupted binary labels via class-probability estimation. In: Proceedings of the 32nd international conference on machine learning, ICML 2015, Lille, France, vol 37, pp 125–134
43. Scott C (2015) A rate of convergence for mixture proportion estimation, with application to learning from noisy labels. In: Proceedings of the eighteenth international conference on artificial intelligence and statistics. San Diego, California, USA, pp 838–846
44. van Rooyen B, Williamson RC (2015) Learning in the presence of corruption. [arXiv preprint arXiv:1504.00091](https://arxiv.org/abs/1504.00091)
45. Han J, Luo P, Wang X (2019) Deep self-learning from noisy labels. [arXiv:1908.02160v1](https://arxiv.org/abs/1908.02160v1)
46. Bouveyron JC, Girard S (2009) Robust supervised classification with mixture models: learning from data with uncertain labels. *Pattern Recognit* 42(11):2649–2658
47. Lichman M (2013) UCI machine learning repository. University of California, School of Information and Computer Science, Irvine, CA. <http://archive.ics.uci.edu/ml>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.