Pattern Recognition 42 (2009) 2659–2670

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Parameter estimation of Poisson mixture with automated model selection through BYY harmony learning

Jinwen Ma*, Jianfeng Liu, Zhijie Ren

Department of Information Science, School of Mathematical Sciences and LAMA, Peking University, Beijing 100871, China

ARTICLE INFO

Article history:
Received 2 November 2007
Received in revised form 17 January 2009
Accepted 31 March 2009

Keywords:
Bayesian Ying–Yang (BYY) harmony learning
Poisson mixture
Gradient learning algorithm
Automated model selection
Texture classification

ABSTRACT

Finite mixture is widely used in the fields of information processing and data analysis. However, its model selection, i.e., the selection of components in the mixture for a given sample data set, has been still a rather difficult task. Recently, the Bayesian Ying–Yang (BYY) harmony learning has provided a new approach to the Gaussian mixture modeling with a favorite feature that model selection can be made automatically during parameter learning. In this paper, based on the same BYY harmony learning framework for finite mixture, we propose an adaptive gradient BYY learning algorithm for Poisson mixture with automated model selection. It is demonstrated well by the simulation experiments that this adaptive gradient BYY learning algorithm can automatically determine the number of actual Poisson components for a sample data set, with a good estimation of the parameters in the original or true mixture where the components are separated in a certain degree. Moreover, the adaptive gradient BYY learning algorithm is successfully applied to texture classification.

1. Introduction

As a powerful probabilistic model, finite mixture distribution has been adopted extensively in a wide variety of practical situations where data can be viewed as arising from two or more populations linearly mixed in certain proportions (e.g. [1–3]). In fact, there are a variety of finite mixtures. Among them, Gaussian mixture is the most well known and frequently used. Clearly, the sample data subject to a Gaussian mixture should be continuous. However, there are many available discrete data which can be regarded being generated from a finite mixture model. Certainly, we can transform these discrete data into continuous ones via some appropriate techniques and still use the Gaussian mixture model to analyze them, e.g. like the correspondence analysis of categorical data. But such a transformation approach has serious limitations because some useful information can be lost during the transformation. Moreover, the Gaussian or normal assumption may not be appropriate for some practical problems, especially in the cases of count data. For these reasons, more attention is being focused on the finite mixtures whose components are not Gaussian densities. Actually, Poisson mixture is a typical non-Gaussian finite mixture with a variety of practical applications such as biological and medical data modeling (e.g. [4,5]), analysis of user accesses to web pages [6], text mining [7], shopper classification [8] and gray level texture classification [9]. Theoretically, Poisson mixtures are identifiable [10] and owns a great number of properties (refer to the recent review [11]).

In the conventional finite mixture modeling or parameter learning, in particular for Poisson mixture, it is usually assumed that the number k of components in the mixture is pre-known. In this situation, several statistical or unsupervised learning methods have been established for parameter estimation or learning. The most classical method of this kind may be the method of moments (e.g. [12,13]). But it can only solve the problem of parameter estimation for the mixture with two components or a number of simple components. Clearly, the maximum likelihood principle can be applied to the parameter estimation of finite mixture in general, which led to the EM algorithm for finite mixture [14,15]. Although the EM algorithm has certain good convergence behaviors [16–18], it still has some weaknesses or limitations. Essentially, the EM algorithm is a local searching approach, thus "bad" initialization can make it get trapped in a local solution. Moreover, the Bayesian inference can also be utilized to solve the parameter estimation of finite mixture [3], but it is rather difficult to set up a set of reasonable prior distributions for the parameters in the mixture. On the other hand, the parameter estimation problem of finite mixture can be solved with the minimum distance principle with the help of defining certain distance measure between the underlying (or actual) and the estimated distributions (e.g. [2,19]). Although many such distances have been established, it is usually rather difficult to implement the minimum distance principle for the parameter estimation of finite mixture efficiently.

* Corresponding author.
E-mail address: jwma@math.pku.edu.cn (J. Ma).

0031-3203/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2009.03.029

When the number of components is not known in advance, the finite mixture modeling becomes complicated and difficult because the selection of an appropriate number of components must be made jointly with the estimation of the parameters [20]. Since the number of components is just a scale of the finite mixture model, its selection for the mixture to model a sample data set is usually referred to as the model selection. Certainly, as a typical class of finite mixtures, Poisson mixtures face the same compound modeling problem of parameter estimation or learning and model selection. Although this problem might be solved by choosing a best number k^* of components as the clusters in the sample data set via one of information, coding and statistical selection criteria such as Akaike's Information Criterion [21] or its extensions (e.g. [22,23]), MDL [24], MML [25], likelihood ratio test (LRT) [26] and the Bootstrapping methods [27,28], the process of evaluating a criterion incurs a large computational cost since we need to repeat the entire parameters learning process at a large number of different values of k. Moreover, all the existing theoretical selection criteria have their limitations and often lead to a wrong result.

Since 1990s, there have appeared some new approaches to solve this compound mixture modeling problem. One approach was to use a kind of stochastic simulation to infer the optimal mixture model. The two typical implementations are the methods of Dirichlet processes [29] and reversible jump Markov chain Monte Carlo (RJMCMC) [30]. However, these stochastic simulation methods generally require a large number of samples through different sampling rules. Another approach was the unsupervised learning [31] on finite mixture which introduces certain competitive learning mechanism into the EM algorithm such that the model selection can be made during the parameter learning by annihilating the components with very small mixing proportions during the parameter learning via the MML principle. As a matter of fact, our proposed approach in the current paper follows a similar rule on the model selection and thus we will use the unsupervised learning algorithm for Poisson mixture as the counterpart to compare our proposed learning algorithm.

The Bayesian Ying–Yang (BYY) harmony learning system and theory, which was first proposed in 1995 [32] and then systematically developed and summarized in the recent years in [33–35], has provided another new approach to solving this complicated compound problem of finite mixture modeling. Actually, the BYY harmony learning acts as a general statistical learning framework, which is useful not only for understanding several existing major learning approaches but also for tackling the learning problem on a set of finite samples with a new learning mechanism that makes model selection automatically during parameter learning. Particularly, we can implement such a mechanism of parameter learning with automated model selection on a certain BYY system for finite mixture via maximizing a harmony function, which is reduced from the harmony measure between the Ying and Yang machines in the BYY learning system. In fact, it was already shown in [36] that the compound mixture modeling problem for Gaussian mixture can be solved through the maximization of a harmony function on a specific BI-directional architecture (BI-architecture) of the BYY system for the Gaussian mixture model via a gradient learning rule such that an appropriate number of Gaussians can be automatically allocated for the sample data set, with the mixing proportions of the extra Gaussians attenuating to zero. Moreover, the adaptive, conjugate, natural gradient and fixed-point learning algorithms [37–39] were further proposed to improve the efficiency of the harmony function maximization. On the other hand, an annealing learning algorithm was also proposed via the maximization of the harmony function on the back-directional architecture (B-architecture) of the BYY system for Gaussian mixture with automated model selection [40]. Methodically, this BYY harmony learning approach can be further applied to any types of non-Gaussian mixtures.

In the current paper, we extend the BYY harmony learning mechanism of parameter learning with automated model selection from Gaussian mixture to Poisson mixture. Under a BI-architecture of the BYY learning system for Poisson mixture, an adaptive gradient learning algorithm or rule for maximizing the harmony function is constructed to achieve the parameter learning or estimation of Poisson mixture with automated model selection. It is demonstrated well by the simulation experiments that the adaptive gradient BYY learning algorithm can make model selection automatically during the parameter learning on the sample data as long as the actual Poisson components in the original mixture are separated in a certain degree. Moreover, it is successfully applied to texture classification.

The rest of this paper is organized as follows. We begin with the introduction of Poisson distribution and mixture in Section 2. Then, after an introduction of the BYY learning system and the harmony function for Poisson mixture, we derive the adaptive gradient BYY learning algorithm in Section 3. The proposed BYY learning algorithm is further demonstrated by simulation experiments in Section 4 and applied to the gray images based texture classification in Section 5. Finally, we make a brief conclusion in Section 6.

2. Poisson distribution and mixture

We begin to briefly introduce Poisson distribution and mixture, respectively and leave some details to [15,41].

2.1. Poisson distribution

As is well known, Poisson distribution is a typical discrete probability model for count events or data, such as the number of telephone calls you receive in an hour, the number of dandelions per square meter on the college playing field and the number of cars per mile broken down on the hard shoulder of the motor-way. Mathematically, a univariate Poisson (probability) distribution is defined as follows:

$$p(x|\theta) = \frac{\theta^x}{x!}e^{-\theta}, \quad x = 0, 1, 2, \ldots, \tag{1}$$

where the parameter $\theta > 0$, and $x!$ is the factorial of x. If a discrete random variable takes the probability as in Eq. (1) at each possible value, we say that it is a Poisson random variable X and follows a Poisson distribution with parameter θ.

For defining a multivariate Poisson distribution, we introduce a random vector $Y = [Y_1, Y_2, \ldots, Y_m]^T$, where each Y_i is an independent Poisson random variable with parameter θ_i. Then, a multivariate Poisson distribution can be defined as the probability distribution of the random vector $\mathbf{A}Y$, where $\mathbf{A}$ belongs to a special class of 0–1 matrixes (i.e., their elements are either zeros or ones). Supposing that the matrix $\mathbf{A}$ is such a $q \times m$ 0–1 matrix, the vector of random variables $X = \mathbf{A}Y$ (i.e., $X = [X_1, X_2, \ldots, X_q]^T$) follows a multivariate Poisson distribution [41]. Theoretically, the 0–1 matrix $\mathbf{A}$ should reflect the covariance between any two random variables X_i and X_j, which usually makes $\mathbf{A}$ take the following form:

$$\mathbf{A} = [A_1, A_2, \ldots, A_m],$$

where each A_i is a $q \times \binom{q}{i}$ matrix whose columns are all the q-dimensional 0–1 vectors containing exactly i ones and $q - i$ zeros. For instance, at $q = 2$, we have

$$A_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

With the help of such a matrix $\mathbf{A}$, we can then define the multivariate Poisson distribution from a group of independent Poisson variables. An implication of the above definition is that a multivariate Poisson

distribution can be defined via a multivariate reduction technique. For example, in the case of $q=2$, we can use three independent Poisson random variables Y_i, with parameters θ_i, for $i \in S=\{1,2,0\}$, and obtain the following representation formulae respectively for X_1, X_2:

$$X_1 = Y_1 + Y_0,$$

$$X_2 = Y_2 + Y_0.$$

In this situation, matrix $\mathbf{A}$ is given by

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix},$$

and $Y=(Y_1,Y_2,Y_0)^T$. Particularly, the probability distribution or function of $X=[X_1,X_2]^T$ at each point $x=(x_1,x_2)^T$ is given as follows:

$$p(x_1,x_2|\theta_1,\theta_2,\theta_0) = e^{-(\theta_1+\theta_2+\theta_0)} \frac{\theta_1^{x_1}}{x_1!}\frac{\theta_2^{x_2}}{x_2!} \sum_{i=0}^{min(x_1,x_2)} \binom{x_1}{i}\binom{x_2}{i} i! \left(\frac{\theta_0}{\theta_1\theta_2}\right)^i. \quad (2)$$

Clearly, Y_0 acts as a correlation term between X_1 and X_2. Moreover, the correlation can be seen from the expressions of mean vector and covariance matrix of the random vector X, which can be computed as follows:

$$E(X) = \mathbf{A}M, \quad (3)$$

$$Var(X) = \mathbf{A}\mathbf{\Sigma}\mathbf{A}^{\mathbf{T}}, \quad (4)$$

where

$$M = E(Y) = (\theta_1, \theta_2, \ldots, \theta_m)^T$$

and $\mathbf{\Sigma}$ is the covariance matrix of Y given by

$$\mathbf{\Sigma} = E[(Y-E(Y))(Y-E(Y))^T] = E[(Y-M)(Y-M)^T] = diag[\theta_1,\theta_2,\ldots,\theta_m].$$

For practical applications with the multivariate Poisson model, we generally assume that either there is a common correlation-generating term for any two random variables or simply all the variables are independent in the random vector. For the former correlation case, we let $X=(X_1,X_2,\ldots,X_m)^T$ be constructed from $Y=(Y_0,Y_1,\ldots,Y_m)^T$ via the simple relations: $X_i=Y_i+Y_0 (i=1,\ldots,m)$, where each Y_i is an independent Poisson random variable with parameter θ_i. In this way, the probability distribution or function of X at each point $x=(x_1,x_2,\ldots,x_m)^T$ can be expressed as follows:

$$p(x_1,x_2,\ldots,x_m|\theta_0,\theta_1,\ldots,\theta_m) = \exp\left(-\sum_{i=0}^{m}\theta_i\right)\prod_{i=1}^{m}\frac{\theta_i^{x_i}}{x_i!}\sum_{l=0}^{s}\prod_{j=1}^{m}\binom{x_j}{l} l!\left(\frac{\theta_0}{\prod_{k=1}^{m}\theta_k}\right)^l, \quad (5)$$

where $s=\min(x_1,x_2,\ldots,x_m)$.

For the latter independent case, supposing that all the random variables X_i are independent Poisson random variables with parameters θ_i, we can easily get the probability distribution or function of $X=(X_1,X_2,\ldots,X_m)^T$ at each point $x=(x_1,x_2,\ldots,x_m)^T$ as follows:

$$p(x_1,x_2,\ldots,x_m|\theta_1,\theta_2,\ldots,\theta_m) = \prod_{i=1}^{m}\frac{\theta_i^{x_i}}{x_i!}e^{-\theta_i}. \quad (6)$$

2.2. Poisson mixture

In many practical applications, the observed data can be considered being generated from a number of components that are linearly mixed in certain proportions. That is, the observed data are subject to a finite mixture distribution. The major task is then to solve the compound mixture modeling problem of model selection and parameter estimation, i.e., to determine the number of components and estimate the parameters of the component distributions as well as the mixing proportions, only with a set of sample data. Theoretically, we consider the following finite mixture model:

$$q(x|\Theta_k) = \sum_{j=1}^{k}\alpha_j q(x|\theta_j), \quad (7)$$

where $q(x|\theta_j)$ are component probability densities or distributions with parameters θ_j, k is the number of components in the mixture, x denotes the variable or variable vector, and $\alpha_j \geqslant 0$ are mixing proportions of the components with the constraint that $\sum_{j=1}^{k}\alpha_j = 1$. For clarity, we let $\Theta_k=\{\alpha_j,\theta_j\}_{j=1}^{k}$ be the set of all parameters in the mixture model.

If all $q(x|\theta_j)$ in Eq. (7) are Poisson probability distributions, the finite mixture is called a Poisson mixture. As a typical class of finite mixtures, Poisson mixtures are not only important in statistics, but also widely used in practical applications. As discussed in the previous section, many efforts have been made on the model selection and parameter estimation of Poisson mixture. As a matter of fact, the EM algorithm is probably the most frequently used method to estimate the parameters of the Poisson mixture with a sample data set [14,15]. However, the EM algorithm is constructed under a framework of maximum likelihood and thus is unable to make model selection for Poisson mixture only with a set of sample data. In the following, based on the BYY harmony learning, we will construct a BYY harmony learning algorithm for Poisson mixture to make model selection automatic during parameter learning.

3. Adaptive gradient BYY learning algorithm

In this section, we further introduce the BI-architecture of the BYY learning system on which the harmony learning turns into the parameter learning with automated model selection on the finite mixture model [34,36], and then derive the adaptive gradient BYY learning algorithm for Poisson mixture.

3.1. BYY learning system and harmony function for Poisson mixtures

A BYY system describes each observation $x \in \mathcal{X} \subset \Re^n$ and its corresponding inner representation $y \in \mathcal{Y} \subset \Re^m$ via the two types of Bayesian decomposition of the joint density: $p(x,y)=p(x)p(y|x)$ and $q(x,y)=q(y)q(x|y)$, which are called Yang machine and Ying machine, respectively. Given a data set $D_x=\{x_t\}_{t=1}^{N}$ from the Yang or observable space, the goal of harmony learning on a BYY learning system is to extract the hidden probabilistic structure of x with the help of y from specifying all aspects of $p(y|x)$, $p(x)$, $q(x|y)$ and $q(y)$ via a harmony learning principle implemented by maximizing the functional

$$H(p||q) = \int p(y|x)p(x)\ln[q(x|y)q(y)]\,dx\,dy, \quad (8)$$

which is essentially equivalent to minimizing the Kullback–Leibler divergence between the Yang and Ying machines, i.e., $p(x,y)$ and $q(x,y)$, because

$$KL(p\|q) = \int p(y|x)p(x)\ln\frac{p(y|x)p(x)}{q(x|y)q(y)}\,dx\,dy = -H(p\|q) - H(p), \quad (9)$$

where $H(p)$ is the entropy of $p(x,y)$ and invariant to $q(x,y)$.

If both $p(y|x)$ and $q(x|y)$ are parametric, i.e., from a family of probability densities with parameter θ, the BYY learning system is said to have a BI-directional architecture (BI-architecture). For the Poisson mixture model with a given sample set $D_x = \{x_t\}_{t=1}^N$, we can utilize the following specific BI-architecture of the BYY learning system. The inner representation y is discrete in $\mathscr{Y} = \{1, 2, \ldots, k\}$ (i.e., with $m = 1$), and the observation x is also discrete from a Poisson mixture distribution. On the Ying space, we let $q(y = j) = \alpha_j \geqslant 0$ with $\sum_{j=1}^k \alpha_j = 1$. On the Yang space, we suppose that $p(x)$ is a blind Poisson mixture probability distribution, with a set of sample data D_x being generated from it. Moreover, in the Ying path, we let each $q(x|y = j) = q(x|\theta_j)$ be a Poisson probability distribution with parameter θ_j consisting of all its parameters, while the Yang path is constructed under the Bayesian principle by the following parametric form:

$$p(y = j|x) = \frac{\alpha_j q(x|\theta_j)}{q(x|\Theta_k)}, \quad q(x|\Theta_k) = \sum_{j=1}^k \alpha_j q(x|\theta_j), \tag{10}$$

where $\Theta_k = \{\alpha_j, \theta_j\}_{j=1}^k$ and $q(x|\Theta_k)$ is just a Poisson mixture that will approximate the true Poisson mixture $p(x)$ hidden in the sample data D_x via the harmony learning on the BYY learning system.

With all these component densities in Eq. (8), we have

$$H(p||q) = E_{p(x)}\left[\sum_{j=1}^k \frac{\alpha_j q(X|\theta_j)}{\sum_{i=1}^k \alpha_i q(X|\theta_i)} \ln[\alpha_j q(X|\theta_j)]\right], \tag{11}$$

that is, it becomes the expectation of a random variable $\sum_{j=1}^k ((\alpha_j q(X|\theta_j))/(\sum_{i=1}^k \alpha_i q(X|\theta_i))) \ln[\alpha_j q(X|\theta_j)]$, where X is just the random variable (or vector) subject to $p(x)$. Based on the given sample data set D_x, we get an estimate of $H(p||q)$ as the following harmony function for Poisson mixtures with the parameter set Θ_k:

$$J(\Theta_k) = \frac{1}{N}\sum_{t=1}^N \sum_{j=1}^k \frac{\alpha_j q(x_t|\theta_j)}{\sum_{i=1}^k \alpha_i q(x_t|\theta_i)} \ln[\alpha_j q(x_t|\theta_j)]. \tag{12}$$

Based on Eqs. (8), (11) and (12), we actually have a new derivation of the harmony function $J(\Theta_k)$ in finite mixture. With help of probability theory and statistics, this derivation is more reasonable and clearer than the previous derivation in [34,36,37].

According to the BYY harmony learning principle [34,35], the maximization of the harmony function $J(\Theta_k)$ is able to make model selection automatic during parameter learning for Poisson mixture with a sample data set in which the number N of sample points is large enough and the actual components are separated in a certain degree. That is, in such a situation, as long as we set k to be larger than the number k^* of actual Poissons in the sample data, it can make k^* Poissons from the estimated mixture match the actual Poissons, respectively, and force the mixing proportions of the other $k-k^*$ extra Poissons to attenuate to zero. In order to do so, we will construct an adaptive gradient BYY learning algorithm to search the maximum of $J(\Theta_k)$ in the next subsection.

3.2. Derivation of the adaptive gradient BYY learning rule

For convenience of derivation of $J(\Theta_k)$, we introduce a group of intermediate variables $U_j(x) = \alpha_j q(x|\theta_j)$ for $j = 1, 2, \ldots, k$, as we previously did in [37]. In this way, $J(\Theta_k)$ takes a simple and structural form:

$$J(\Theta_k) = \frac{1}{N}\sum_{t=1}^N J_t(\Theta_k), \quad J_t(\Theta_k) = \sum_{j=1}^k \frac{U_j(x_t)}{\sum_{i=1}^k U_i(x_t)} \ln U_j(x_t). \tag{13}$$

Moreover, in order to get rid of the constrains on α_j, we utilize the following so-called softmax representation:

$$\alpha_j = \frac{e^{\beta_j}}{\sum_{i=1}^k e^{\beta_i}}, \quad j = 1, 2, \ldots, k, \tag{14}$$

where $-\infty < \beta_1, \ldots, \beta_k < +\infty$.

With the above preparations, we can get the derivatives of $J(\Theta_k)$ with respect to β_j and θ_j at sample point x_t as follows:

$$\begin{aligned}\frac{\partial J_t(\Theta_k)}{\partial \beta_j} &= \sum_{i=1}^k \frac{\partial J_t(\Theta_k)}{\partial U_i(x_t)} \frac{\partial U_i(x_t)}{\partial \beta_j} \\ &= \frac{1}{q(x_t|\Theta_k)} \sum_{i=1}^k \left[1 - \sum_{l=1}^k (p(l|x_t) - \delta_{il}) \ln U_l(x_t)\right] \\ &\quad \times (\delta_{ij} - \alpha_j) U_i(x_t),\end{aligned} \tag{15}$$

$$\begin{aligned}\frac{\partial J_t(\Theta_k)}{\partial \theta_j} &= \sum_{i=1}^k \frac{\partial J_t(\Theta_k)}{\partial U_i(x_t)} \frac{\partial U_i(x_t)}{\partial \theta_j} \\ &= \frac{1}{q(x_t|\Theta_k)} \left[1 - \sum_{l=1}^k (p(l|x_t) - \delta_{jl}) \ln U_l(x_t)\right] \alpha_j \frac{\partial q(x_t|\theta_j)}{\partial \theta_j},\end{aligned} \tag{16}$$

where δ_{ij} is the Kronecker function.

Letting

$$\lambda_i(t) = 1 - \sum_{l=1}^k (p(l|x_t) - \delta_{il}) \ln U_l(x_t), \quad i = 1, \ldots, k \tag{17}$$

and according to Eqs. (15) and (16), we have the following general adaptive gradient rules at sample x_t:

$$\nabla_{\beta_j} J_t(\Theta_k) = \frac{1}{q(x_t|\Theta_k)} \sum_{i=1}^k \lambda_i(t)(\delta_{ij} - \alpha_j) U_i(x_t), \tag{18}$$

$$\nabla_{\theta_j} J_t(\Theta_k) = \frac{\lambda_j(t)\alpha_j}{q(x_t|\Theta_k)} \frac{\partial q(x_t|\theta_j)}{\partial \theta_j}. \tag{19}$$

As each $q(x_t|\theta_j)$ takes the form of univariate or multivariate Poisson distribution, we can get the adaptive gradient rules of $J(\Theta_k)$ for the mixture of univariate or multivariate Poisson distributions according to the above general adaptive gradient rules. That is, we need only to replace these $\partial q(x_t|\theta_j)/\partial \theta_j$, i.e., the derivative of $q(x_t|\theta_j)$ with respect to θ_j, by their particular expressions. Now, we give them for the multivariate Poisson distributions in the two cases as follows.

(1) *The correlation case* (*defined by Eq.* (5)): Letting $x_t = (x_{t1}, x_{t2}, \ldots, x_{tm})^T$ and $\theta_j = (\theta_{j0}, \theta_{j1}, \ldots, \theta_{jm})^T$, we then have

$$\frac{\partial q(x_t|\theta_j)}{\partial \theta_j} = \left(\frac{\partial q(x_t|\theta_j)}{\partial \theta_{j0}}, \frac{\partial q(x_t|\theta_j)}{\partial \theta_{j1}}, \ldots, \frac{\partial q(x_t|\theta_j)}{\partial \theta_{jm}}\right)^T. \tag{20}$$

If there are some components of x_t which are zero, i.e., there is a nonempty subset H of $\{1, 2, \ldots, m\}$ such that $x_{th} = 0$ iif $h \in H$, the derivatives of $q(x_t|\theta_j)$ with respect to these θ_{jh} (i.e., $h \in H$) take the following simple form (noting that $0! = 1$):

$$\frac{\partial q(x_t|\theta_j)}{\partial \theta_{j0}} = \frac{\partial q(x_t|\theta_j)}{\partial \theta_{jh}} = -\exp\left(-\sum_{i=0}^m \theta_{ji}\right) \prod_{i=1}^m \frac{\theta_{ji}^{x_{ti}}}{x_{ti}!}. \tag{21}$$

As for the derivatives of $q(x_t|\theta_j)$ with respect to $\theta_{jh'}$ where $h' \notin H$, we have

$$\frac{\partial q(x_t|\theta_j)}{\partial \theta_{jh'}} = \exp\left(-\sum_{i=0}^m \theta_{ji}\right)\left(\frac{x_{th'}}{\theta_{jh'}} - 1\right) \prod_{i=1}^m \frac{\theta_{ji}^{x_{ti}}}{x_{ti}!}. \tag{22}$$

On the other hand, if there is no zero component in x_t, we have the derivatives of $q(x_t|\theta_j)$ with respect to each θ_{ih} as follows:

$$\frac{\partial q(x_t|\theta_j)}{\partial \theta_{jh}} = q(x_t|\theta_j)\left(\frac{x_{th}}{\theta_{jh}} - 1\right) - V_h(\theta_j), \quad (h > 0) \tag{23}$$

$$\frac{\partial q(x_t|\theta_j)}{\partial \theta_{j0}} = V_0(\theta_j) - q(x_t|\theta_j), \tag{24}$$

where

$$V_n(\theta_j) = \exp\left(-\sum_{i=0}^{m}\theta_{ji}\right)\prod_{i=1}^{m}\frac{\theta_{ji}^{x_{ti}}}{x_{ti}!}\sum_{i=1}^{s}\prod_{l=1}^{m}\binom{x_{tl}}{i} i!\left(\frac{\theta_{j0}}{\prod_{k=1}^{m}\theta_{jk}}\right)^{i}\frac{i}{\theta_{jn}}$$

for $n = 0, 1, \ldots, m$.

(2) *The independent case* (*defined by Eq.* (6)): In this situation, the sample x_t is represented in the same way, but there is no component θ_{j0} in θ_j. When some component x_{th} of x_t is zero, the derivative of $q(x_t|\theta_j)$ with respect to the corresponding θ_{jh} takes the following simple form:

$$\frac{\partial q(x_t|\theta_j)}{\partial \theta_{jh}} = -\prod_{i=1}^{m}\frac{\theta_{ji}^{x_{ti}}}{x_{ti}!}e^{-\theta_{ji}}. \tag{25}$$

Otherwise, for a general component x_{th} of x_t that is not zero, the derivative with respect to the corresponding θ_{jh} takes the following slightly complicated expression:

$$\frac{\partial q(x_t|\theta_j)}{\partial \theta_{jh'}} = \left(\frac{x_{th'}}{\theta_{jh'}} - 1\right)\prod_{i=1}^{m}\frac{\theta_{ji}^{x_{ti}}}{x_{ti}!}e^{-\theta_{ji}}. \tag{26}$$

For the situation where each component is expressed by Eq. (1), i.e., a univariate Poisson distribution, it is certainly a special independent case of Eq. (6).

Summing up all these derivations, we finally obtain the adaptive gradient BYY learning rule for Poisson mixture as follows:

$$\beta_j^{new} = \beta_j^{old} + \eta\nabla_{\beta_j}J_t(\Theta_k), \tag{27}$$

$$\theta_j^{new} = \theta_j^{old} + \eta\nabla_{\theta_j}J_t(\Theta_k), \tag{28}$$

where $\eta(>0)$ denotes the learning rate that starts from a reasonable initial value and then reduces to zero with the iteration number n in such a way that $0 \leqslant \eta(n) \leqslant 1$, and

$$\lim_{n\to\infty}\eta(n) = 0, \quad \sum_{n=1}^{\infty}\eta(n) = \infty. \tag{29}$$

The typical example of the learning rate satisfying Eq. (29) is $\eta(n) = \eta_0/n$, where η_0 is a positive constant.

4. Simulation results and comparisons

In this section, simulation experiments are carried out to demonstrate the performance of the adaptive gradient BYY learning algorithm for Poisson mixture for both model selection and parameter estimation on a sample data set from a Poisson mixture, being compared with that of the unsupervised learning algorithm [31] for Poisson mixture.

4.1. Sample data sets

To test our proposed adaptive gradient BYY learning algorithm for Poisson mixture, we generate eight typical sample data sets $\mathscr{S}_1, \mathscr{S}_2, \ldots, \mathscr{S}_8$ from finite mixtures of Poisson distributions of different types (e.g. univariate, bivariate, trivariate, correlation and independent Poisson distributions) with different sample sizes or mixing proportions, and the parameters of these Poisson mixtures are summarized in Table 1.

Table 1
The parameters of the eight (original) Poisson mixtures to generate sample data sets for simulation experiments.

Data set	Poissons	θ_1^i	θ_2^i	θ_0^i/θ_3^i	α_i	N_i
$\mathscr{S}_1(k^*=1)$	Poisson 1	6.0			1.0	300
$\mathscr{S}_2(k^*=2)$	Poisson 1	1.0			0.20	100
	Poisson 2	10.0			0.80	400
$\mathscr{S}_3(k^*=2)$	Poisson 1	1.0	2.0		0.20	100
	Poisson 2	7.0	8.0		0.80	400
$\mathscr{S}_4(k^*=2)$	Poisson 1	3.0	2.0		0.50	250
	Poisson 2	12.0	14.0		0.50	250
$\mathscr{S}_5(k^*=2)$	Poisson 1	1.0	2.0	1.0	0.40	200
	Poisson 2	11.0	12.0	8.0	0.60	300
$\mathscr{S}_6(k^*=3)$	Poisson 1	1.0	2.0		1/3	200
	Poisson 2	10.0	11.0		1/3	200
	Poisson 3	23.0	22.0		1/3	200
$\mathscr{S}_7(k^*=3)$	Poisson 1	3.0	2.0		1/3	200
	Poisson 2	9.0	10.0		1/3	200
	Poisson 3	15.0	16.0		1/3	200
$\mathscr{S}_8(k^*=3)$	Poisson 1	1.0	2.0	1.0	1/3	200
	Poisson 2	4.0	5.0	6.0	1/3	200
	Poisson 3	10.0	12.0	11.0	1/3	200

Specifically, the first sample data set $\mathscr{S}_1$ contains the sample data only from one univariate Poisson distribution, just as a degenerated Poisson mixture, while the other sample data sets contain the sample data from two or three Poisson distributions which always have some overlap, but keep separated in certain sense. The generating Poisson variables are univariate, bivariate and trivariate for $\mathscr{S}_1$ to $\mathscr{S}_2$, $\mathscr{S}_3$ to $\mathscr{S}_7$ and $\mathscr{S}_8$, respectively. Moreover, the sample data in $\mathscr{S}_5$ are generated from a correlation Poisson mixture, while the sample data in any of the other sample data sets are generated from an independent Poisson mixture. Particularly for $\mathscr{S}_5$, the two component Poisson variables are correlated via a common Poisson variable with parameter θ_0 as described previously in Section 2.2. For illustration, we sketch the sample data or points in each of $\mathscr{S}_3, \ldots, \mathscr{S}_8$, respectively, in Fig. 1, from which we can observe that the component Poisson distributions in $\mathscr{S}_3, \ldots, \mathscr{S}_6$ are strongly separated, while those in $\mathscr{S}_7$ and $\mathscr{S}_8$ are overlapped in a higher degree.

4.2. Simulation results

We implement the adaptive gradient BYY learning algorithm for Poisson mixture on these eight synthetic data sets with $k \geqslant k^*$. The parameters of the adaptive gradient BYY learning algorithm are initialized randomly in some intervals under the constraints. Particularly in our experiments, we initialize the parameters as follows. At first, we randomly set the initial values of the mixing proportions α_j under the constraints that $\alpha_j \geqslant 0$ and $\sum_{j=1}^{k}\alpha_j$. Then, according to these given mixing proportions, we divide all the samples into k classes. In the independent case of Poisson mixture, each component parameter vector θ_j is initialized as the corresponding sample mean vector of the j-th class. In the correlation case for $\mathscr{S}_5$, according to Eqs. (3) and (4), we have the following probability relations on each Poisson component j:

$$E(X_j) = (\theta_{j1} + \theta_{j0}, \theta_{j2} + \theta_{j0})^T, \tag{30}$$

$$Var(X_j) = \begin{pmatrix} \theta_{j1} + \theta_{j0} & \theta_{j0} \\ \theta_{j0} & \theta_{j2} + \theta_{j0} \end{pmatrix}. \tag{31}$$

When we use the sample mean vector and covariance matrix of $\mathscr{S}_5$ instead of the mean vector in Eq. (30) and the covariance matrix in Eq. (31), respectively, we can get the two equations from which the

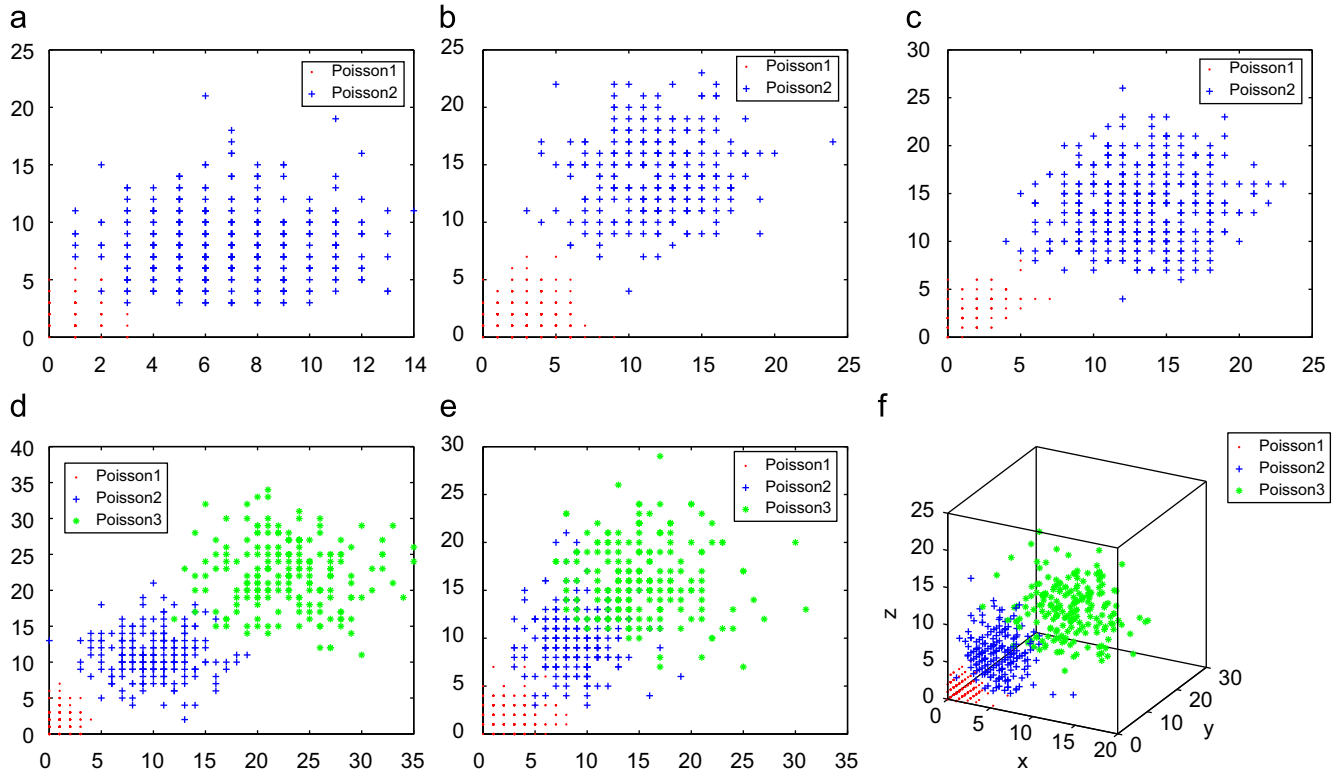

Fig. 1. Six sets of sample data used in the experiments: (a) $\mathscr{S}_3$, (b) $\mathscr{S}_4$, (c) $\mathscr{S}_5$, (d) $\mathscr{S}_6$, (e) $\mathscr{S}_7$ and (f) $\mathscr{S}_8$.

component parameters θ_{ji} can be easily solved. Actually, we get the statistical estimates of the component parameters and make them as their initial values in our learning algorithm. As a result, we set the initial values of the component parameters as their statistical estimates on the sample data. In order to speed up the convergence, we also add an annihilation mechanism on the components in our learning algorithm in the same way of the unsupervised learning algorithm on finite mixture [31]. That is, at each iteration, if the mixing proportion of a component is less than a threshold value $\varepsilon > 0$, we discard or annihilate this component from the mixture directly. In the simulation experiments, we set ε=0.005 by experience. Learning is stopped once the terminating criterion $|J(\Theta_k^{new}) - J(\Theta_k^{old})| < 10^{-5}$ is satisfied. Actually, we find that our learning algorithm always converge in all attempts.

The average experimental results of the adaptive gradient BYY learning algorithm on the eight sample data sets are listed in Table 2. Typically, k=3k^* for $\mathscr{S}_1$ and k=2k^* for all the other sample data sets. For each sample data set, we conduct 100 experiments by selecting different initial values of the mixing proportions. In all the cases, the adaptive gradient BYY learning algorithm leads to the correct model selection, i.e., finally allocating the correct number of Poissons for the sample data set. In the meantime, it also results in an estimate for each parameter in the original or true Poisson mixture which generated the sample data. The average results of the parameter estimations as well as the standard deviations from 100 experiments for each sample data set are listed in Table 2. As compared with the original parameters in Table 1, we can find that the adaptive gradient BYY learning algorithm always obtains a stable and accurate estimate for each parameter in the independent Poisson mixture. However, for the correlation Poisson mixture, the parameter estimations from the adaptive gradient BYY learning algorithm are still acceptable, but their stabilities and accuracies are decreased considerably, which is probably caused by the correlation feature of this special mixture architecture.

Table 2
The average experimental results of the adaptive gradient BYY learning algorithm on the eight sample data sets.

Data set	Poissons	$\bar{\theta}_1^i \pm \sigma_1^i$	$\bar{\theta}_2^i \pm \sigma_2^i$	$\bar{\theta}_3^i \pm \sigma_3^i$	$\bar{\alpha}_i \pm \sigma_i$
$\mathscr{S}_1(k=3)$	Poisson 1	6.0 ± 0.001			1 ± 0
$\mathscr{S}_2(k=4)$	Poisson 1	0.91 ± 0.001			0.20 ± 0.001
	Poisson 2	9.9 ± 0.002			0.80 ± 0.001
$\mathscr{S}_3(k=4)$	Poisson 1	0.92 ± 0.001	2.1 ± 0.001		0.19 ± 0.001
	Poisson 2	6.9 ± 0.001	8.2 ± 0.003		0.81 ± 0.001
$\mathscr{S}_4(k=4)$	Poisson 1	3.1 ± 0.001	2.0 ± 0.001		0.5 ± 0.001
	Poisson 2	11.9 ± 0.005	14.3 ± 0.006		0.5 ± 0.001
$\mathscr{S}_5(k=4)$	Poisson 1	1.0 ± 0.013	2.0 ± 0.014	0.91 ± 0.012	0.40 ± 0.001
	Poisson 2	10.7 ± 1.21	11.6 ± 1.23	8.2 ± 1.20	0.6 ± 0.001
$\mathscr{S}_6(k=6)$	Poisson 1	1.0 ± 0.001	1.9 ± 0.001		0.33 ± 0.001
	Poisson 2	9.9 ± 0.010	11.0 ± 0.008		0.34 ± 0.001
	Poisson 3	23.1 ± 0.021	22.6 ± 0.031		0.33 ± 0.001
$\mathscr{S}_7(k=6)$	Poisson 1	2.8 ± 0.004	2.0 ± 0.004		0.33 ± 0.003
	Poisson 2	8.7 ± 0.09	9.8 ± 0.11		0.34 ± 0.009
	Poisson 3	15.9 ± 0.14	16.6 ± 0.11		0.33 ± 0.009
$\mathscr{S}_8(k=6)$	Poisson 1	0.96 ± 0.001	1.88 ± 0.002	0.98 ± 0.001	0.33 ± 0.001
	Poisson 2	4.2 ± 0.005	5.2 ± 0.009	5.7 ± 0.01	0.35 ± 0.001
	Poisson 3	10.5 ± 0.007	12.3 ± 0.009	11.0 ± 0.007	0.32 ± 0.001

4.3. Comparisons with the unsupervised learning algorithm

We further compare the adaptive gradient BYY learning algorithm with the unsupervised learning algorithm [31] for Poisson mixture on both the implementation time and the performance on parameter estimation or parameter estimation accuracy. For short, the proposed adaptive gradient BYY learning algorithm for Poisson

Table 3
The comparisons of the AGL-BYY and UL-MML algorithms on the average implementation times.

Data set	Initial value of k	UL-MML (s)	AGL-BYY (s)
$\mathscr{S}_2(N=500)$	$2k^*-1$	0.8988	1.9341
	$2k^*$	2.3256	1.7743
	$2k^*+1$	4.0159	2.2281
$\mathscr{S}_3(N=500)$	$2k^*-1$	1.3563	1.9877
	$2k^*$	3.0686	1.7986
	$2k^*+1$	6.13325	1.9381
$\mathscr{S}_4(N=500)$	$2k^*-1$	1.3969	1.8689
	$2k^*$	0.8672	1.8789
	$2k^*+1$	3.7414	2.4684
$\mathscr{S}_6(N=600)$	$2k^*-1$	2.9780	3.8177
	$2k^*$	4.7775	5.2666
	$2k^*+1$	28.926	5.6242
$\mathscr{S}_7(N=600)$	$2k^*-1$	4.0789	6.7984
	$2k^*$	23.8398	7.3975
	$2k^*+1$	29.7119	8.3292

mixture is referred to as the AGL-BYY algorithm, while the unsupervised learning algorithm for Poisson mixture based on the MML criterion is referred to as the UL-MML algorithm. For simplicity, we only consider the sets of sample data from the independent Poisson mixtures. For both algorithms, the threshold value for the annihilation mechanism is set as 0.01. On each sample data set, we implement the two algorithms for 100 times with different initial settings and compare their average implementation times and the average parameter estimation accuracies. The AGL-BYY algorithm is implemented with the initial setting as above in the previous subsection. In this situation, the AGL-BYY algorithm can still always lead to the correct model selection. However, if the initial mixing proportions are randomly set as above, the UL-MML algorithm often leads to a wrong result on model selection. In order to overcome this weakness, we set each initial mixing proportion by $1/k$, i.e., $\alpha_j = 1/k$. (Here k is severed as k_{max} in the unsupervised learning algorithm on finite mixture [31].) Accordingly, the sample data are equally divided into k classes and each component parameter vector θ_j is also initialized as the corresponding sample mean vector of the j-th class. With such a parameter initial setting, the UL-MML algorithm generally leads to a correct model selection. Certainly, by different divisions of the sample data, we can get different parameter initial settings. For comparison, we only record the experimental results of the UL-MML algorithm that are successful on model selection.

We first compare the AGL-BYY and UL-MML algorithms on implementation time or convergence speed. The average implementation times (seconds) of the AGL-BYY and UL-MML algorithms on the typical sample data sets $\mathscr{S}_2, \mathscr{S}_3, \mathscr{S}_4, \mathscr{S}_6$ and $\mathscr{S}_7$ for $k = 2k^*-1, 2k^*$ and $2k^*+1$, respectively, are listed in Table 3. From those data, it can be found that the implementation times of the two algorithms generally varies with the structure of the sample data set. For the cases of low overlap between the components in the Poisson mixture or sample data, either algorithm can converge more quickly than the other one. But the AGL-BYY algorithm converges much quickly than the UL-MML algorithm on $\mathscr{S}_7$ which has a relative higher overlap. Moreover, it can be further found that as the overlap becomes stronger, the implementation times for the two algorithms become larger. However, according to the results on $\mathscr{S}_7$, the implementation time of the UL-MML algorithm increases more quickly than that of the AGL-BYY algorithm as the overlap becomes larger. On the other hand, it can also be found that the implementation time of the UL-MML algorithm increases very quickly with the increase in k, but the implementation time of the AGL-BYY algorithm increases slowly with the increase in k.

Table 4
The comparisons of the AGL-BYY and UL-MML algorithms on the parameter estimation accuracies.

Data set	Algorithm	Δ_α	Δ_θ
$\mathscr{S}_2(N=500,k=4)$	UL-MML	5.37e−8	9.40e−5
	AGL-BYY	9.59e−5	0.0068
$\mathscr{S}_3(N=500,k=4)$	UL-MML	6.31e−4	0.0126
	AGL-BYY	0.0014	0.0096
$\mathscr{S}_4(N=500,k=4)$	UL-MML	7.91e−7	0.0016
	AGL-BYY	3.20e−7	0.0016
$\mathscr{S}_6(N=600,k=6)$	UL-MML	3.93e−6	0.0031
	AGL-BYY	2.29e−4	0.0033
$\mathscr{S}_7(N=600,k=6)$	UL-MML	0.0014	0.0065
	AGL-BYY	0.0023	0.0110

We then compare the performances of the two algorithms on parameter estimation. According to the experimental results of each algorithm on a sample data set, for a 1-dimensional parameter β we can compute $\tilde{\beta}$, the radio of the estimated parameter to the actual or true parameter, and then define $\Delta_\beta = \|\tilde{\beta} - 1\|^2$ as a measure of the normalized mean-square error (NMSE) between the estimated parameter and the actual parameter. Thus, Δ_β can be used as a criterion for evaluating the performance of the algorithm on the parameter estimation or simply the parameter estimation accuracy. In order to reflect the general performance of the algorithm, we compute the average value of the parameter estimates from the 100 experiments and then get Δ_β with this average parameter estimate. Moreover, we add the NMSEs for all the parameters of the Poissons in the mixture as Δ_θ and the NMSEs for all the mixing proportions as Δ_α. Actually, the total NMSEs of two kinds of the average parameter estimates of the AGL-BYY and UL-MML algorithms on the sample data sets $\mathscr{S}_2, \mathscr{S}_3, \mathscr{S}_4, \mathscr{S}_6$ and $\mathscr{S}_7$ are listed, respectively, in Table 4. From those data, it can be found that both the AGL-BYY and UL-MML algorithms lead to a very good parameter estimation on Poisson mixture. But the parameter estimation accuracy of the UL-MML algorithm is relatively better than the AGL-BYY algorithm. Actually, this can be explained from the theoretical foundations of the two algorithms. The UL-MML algorithm uses the MML criterion to make model selection and leads to a maximum likelihood estimator for the parameter under the EM framework, i.e., a consistent estimator. But the AGL-BYY algorithm implements the BYY harmony learning which is essentially a rewarding and penalization mechanism on the parameters [37] and leads to a small deviation on the parameter estimation [39].

As for the first sample data set $\mathscr{S}_1$, the AGL-BYY and UL-MML algorithms behave quite differently. In fact, the AGL-BYY algorithm always leads to the true Poisson distribution, which is already presented in Table 2. But the UL-MML algorithm with $k=3$ generally converges to a mixture of two Poisson distributions. That is, it cannot make a correct model selection as we expect.

By the above comparisons of the AGL-BYY and UL-MML algorithms as well as the other experimental results, we can find that, on the model selection, the AGL-BBY algorithm is more efficient than the UL-MML algorithm. Moreover, the implementation time of the AGL-BYY algorithm increases slowly with k. As for parameter estimation, the AGL-BYY algorithm converges to a quite good result, although its parameter estimation accuracy is lower than that of the UL-MML algorithm.

4.4. Further discussions

Finally, we discuss the adaptive gradient BYY learning algorithm on the sample data sets in the other or special situations. In order to

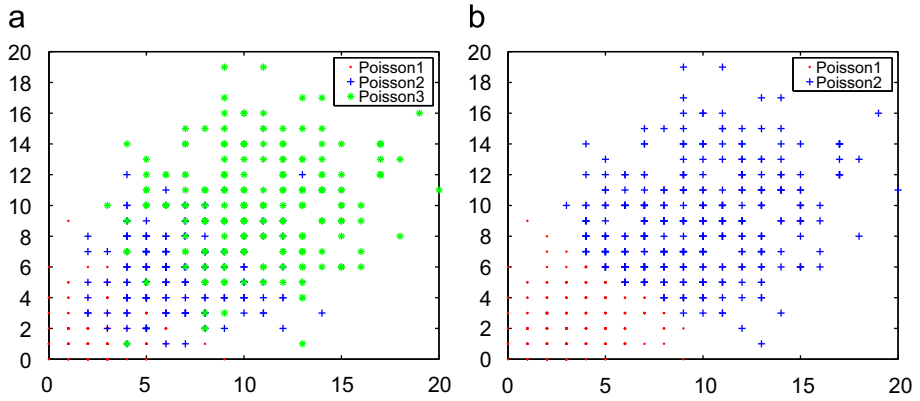

Fig. 2. The sketch of $\mathscr{S}_9$ and the experimental result. (a) $\mathscr{S}_9$ and (b) the classification result of the adaptive gradient BYY learning algorithm on $\mathscr{S}_9$.

do so, we begin to discuss when the algorithm can lead to the correct model selection. Actually, the problem of correct model selection via the BYY harmony learning was already investigated theoretically on Gaussian mixture in [42] and it was proved that the maximization of the harmony function can lead to the correct model selection as long as the overlap among the actual components in the original or true mixture is small enough. Although we cannot extend this result to Poisson mixture, many experimental results show that it is also true for Poisson mixture. That is, as long as the overlap among the actual Poissons in the original mixture keeps a low level, the maximization of the BYY harmony function, Eq. (12), leads to the correct model selection and therefore the adaptive gradient BYY learning algorithm can converge with the correct model selection. On the other hand, when any two actual Poissons are strongly overlapped, the adaptive gradient BYY learning algorithm may leads to a wrong result. For illustration, we give an example of such sample data sets as $\mathscr{S}_9$, being shown in Fig. 2(a), and its classification result of the adaptive gradient BYY learning algorithm is shown in Fig. 2(b). Actually, since the three Poissons in $\mathscr{S}_9$ are strongly overlapped, the adaptive gradient BYY learning algorithm with $k=6$ converges to a mixture of two Poissons in Fig. 2(b) which cannot match the actual Poissons in Fig. 2(a).

By the simulation experiments, we also find that the annihilation mechanism on the components not only speeds up the convergence of the algorithm, but also improve the convergent result of the algorithm, especially on the sample data set with a relatively high overlap among the actual Poissons. Actually, the conventional adaptive gradient learning algorithm given by Eqs. (27) and (28) often leads to a wrong model selection on $\mathscr{S}_7$. But the adaptive gradient BYY learning algorithm with the annihilation mechanism can always get a good result on $\mathscr{S}_7$ as given in Table 2. As for the threshold value ε, it can be selected from a relatively large interval like [0.001, 0.12] for our eight typical sample data sets. Actually, when it is relatively large, the algorithm converges much quickly. Otherwise, when it is relatively small, the parameter estimation accuracy can be improved. However, if we set it very low in the practical applications, the algorithm may keep a number of extra components with very small mixing proportions which try to approximate the noises in the sample data and thus alleviate the generalization ability of the learning system. Therefore, it is reasonable to set it relatively large like 0.01 in the practical applications.

As the overlap among the actual Poissons in the sample data is small enough, just as those of $\mathscr{S}_2, \ldots, \mathscr{S}_8$, the adaptive gradient BYY learning algorithm can always lead to the correct model selection and accurate parameter estimation even with large k. Actually, we conduct the simulation experiments on each of the eight data sets for k from k^* to $\sqrt{N}$ and find out that our algorithm always converges correctly. Generally, we consider that $\sqrt{N}$ is the largest upper bound of the true number k^* of clusters or components in the sample data. Thus, the convergent result of our algorithm does not rely on the selection of k as long as $k \geqslant k^*$ under the low overlap assumption.

In the case of a small set of sample data under such a less overlap condition, it is also shown by the simulation experiments that the adaptive gradient BYY learning algorithm can still lead to the correct model selection and an accurate parameter estimation. For example, from each of $\mathscr{S}_3, \mathscr{S}_4$ and $\mathscr{S}_7$, we can get a small size sample data set which contains only 10 points per each cluster or component, and then conduct the algorithm on this small set of sample data. The experimental results show that our algorithm can still make the correct model selection and obtain an acceptable parameter estimation. However, the unsupervised leaning algorithm for Poisson mixture often leads to a wrong model selection on each of these small sets of sample data. Thus, the adaptive gradient BYY learning algorithm is more robust than the unsupervised learning algorithm on a small size sample data set.

By the other simulation experiments we further find that the adaptive gradient BYY learning algorithm works well for both model selection and parameter estimation on the set of sample data with a large number components (e.g. 15 Poissons) or in a higher dimensional space (e.g. 20-dimensional space) as long as the actual Poissons are separated at a degree as those in the above sample data sets. It is even found by the simulation experiments that the parameter estimation accuracy maintains a similar level with the increase in the space dimension or number of components in the mixture.

In a summary, the adaptive gradient BYY learning algorithm can be implemented efficiently for parameter estimation on Poisson mixture with automated model selection as long as the actual Poisson components in the sample data are separated in a certain degree. Moreover, it outperforms the unsupervised learning algorithm [31] for Poisson mixture on automated model selection. Thus, the BYY harmony learning can be applied to the Poisson mixtures just as it have been applied to the Gaussian mixtures. However, our proposed BYY harmony learning algorithm for Poisson mixture does not work as efficiently as the BYY harmony learning algorithms for Gaussian mixture in some cases. The reasons may be twofold. First, Poisson distributions are discrete on nonnegative integers, and thus there is always certain overlap between any two Poisson distributions. Moreover, the difference between the sample sets of two Poisson distributions is relatively small and the sample data may provide less information for the algorithm. Second, for a Poisson distribution, its mean and variance are only based on one type of parameters θ. Thus, it is difficult to control the distribution location and shape through this unique parameter during the learning. On the other hand, Gaussian distribution has a pair of independent mean and variance parameters and it is easy to control the distribution

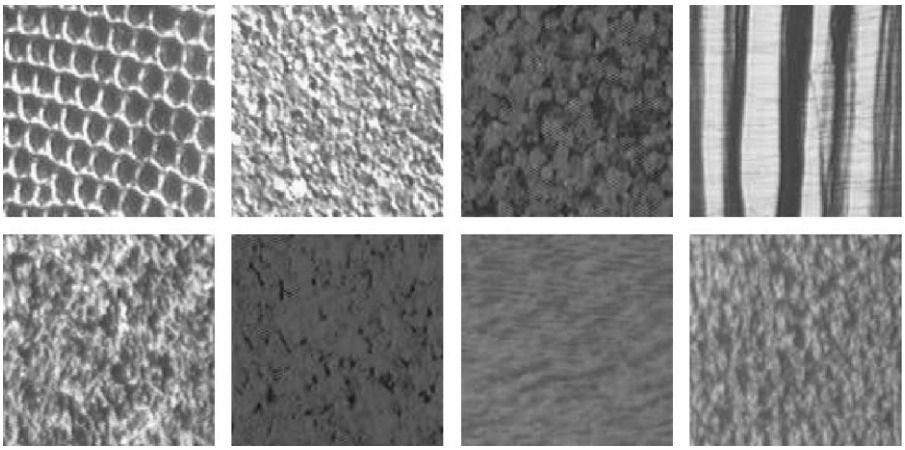

Fig. 3. The gray images: D03, D04, D29, D51, D09, D32, D38 and D24.

location and shape by learning these two kinds of independent parameters.

5. Application to texture classification

In this section, for practical usage and test, we apply the adaptive gradient BYY learning algorithm for Poisson mixture to texture classification or recognition on gray-scale Brodatz textures,[1] and compare it with two current state-of-the-art approaches.

5.1. Poisson mixture learning based texture classification

As is well known, texture is a basic characteristic of an image and texture classification is a fundamental problem in image analysis and computer vision [43]. Actually, the task of texture classification (or recognition) is to classify images into a number of classes with different textures and the major difficulty relies on the description and measure of texture on an image. In the literature, the main focus is on the extraction techniques of texture features. In this work, we use the Poisson mixture model to directly describe the gray images with a particular texture so that our adaptive gradient BYY learning algorithm can be applied to texture classification on gray images.

We adopt the spatial analysis technique for gray images which is fundamental for texture classification [44]. Via this technique, a discrete gray image can be decomposed as a series of gray-level planes defined as follows:

$$f_g(u,v)=\begin{cases}1, & f(u,v)=g,\\ 0, & f(u,v)\neq g,\end{cases} \tag{32}$$

where (u,v) denotes a pair of discrete coordinates for a pixel at the image, $f(u,v)$ is the discrete gray level of the image at the pixel (u,v), and $f_g(u,v)$ is just the gray plane with gray-level g. In this way, the stochastic nature of a gray image can be described through the distributions of the numbers of different gray-level points within a patch, i.e., a square region or block, picked up in the image. Supposing that $\{0,1,\ldots,G-1\}$ is the set of gray levels in consideration, as we pick up a patch B of $L\times L$ pixels in the image, we get a random feature vector $\mathbf{W}=(W_0,\ldots,W_{G-1})^T$, where each component W_g is just the number of points in B with gray-level g, being computed by

$$W_g=\sum_{u,v\in B} f_g(u,v). \tag{33}$$

[1] Brodatz texture images from http://www.cipr.rpi.edu/resource/stills/brodatz.html.

With this random feature vector $\mathbf{W}$, we can construct a Bayesian classifier for texture classification. Firstly, we pick up a number of patches of a fixed size (i.e., $L\times L$ pixels) from the given gray images with different texture classes, being denoted by $\{1,2,\ldots,C\}$. Then, we can get the feature vectors $\mathbf{W}$s for those image patches with the help of Eq. (33) as well as their corresponding texture indexes. Based on these sample data, we can train the Bayesian classifier through the posteriori probability given as follows:

$$P(T_c|\mathbf{W})=\frac{P(\mathbf{W}|T_c)P(T_c)}{\sum_{i=1}^{C}P(\mathbf{W}|T_i)P(T_i)}, \tag{34}$$

where T_c represents texture class c, $P(T_c)$ is the prior probability correspondingly, and $P(\mathbf{W}|T_c)$ denotes the conditional probability of $\mathbf{W}$ given T_c. If all the components in the posteriori probability formula are trained to be known, the Bayesian classifier just assigns the texture class index c^* gaining the highest $P(T_c|\mathbf{W})$ to $\mathbf{W}$ or the image patch it represents.

Actually, if we can maintain that $P(T_c)$ is equally distributed, i.e., $P(T_c)=1/C$, we need to learn only the parameters of those conditional probability distributions $P(\mathbf{W}|T_c)$. For simplicity we assume that the numbers of points of G gray levels within an image patch, i.e., the numbers of black points (with the value one) in G gray-level planes, are independent and then the conditional probability $P(\mathbf{W}|T_c)$ takes the following simple product form:

$$P(\mathbf{W}|T_c)=\prod_{i=0}^{G-1}P(W_i|\Theta_{i,c}), \tag{35}$$

where $P(W_i|\Theta_{i,c})$ is the probability distribution of W_i being parameterized by $\Theta_{i,c}$. As each W_i takes the values of nonnegative integers, $P(W_i|\Theta_{i,c})$ can be expressed or at least approximated by a Poisson mixture model, i.e.,

$$P(W_i|\Theta_{i,c})=\sum_{j=1}^{k}\alpha_j q(W_i|\theta_{i,c,j}), \tag{36}$$

where k is the number of Poisson components. In this way, our task of texture classification relies on parameter estimation and model selection of the Poisson mixture for each W_i. So, the texture classification problem has been transformed into a learning problem on Poisson mixture with a sample data set. In this case, the number of Poisson components in each Poisson mixture is not clear. Thus, our adaptive gradient BYY learning algorithm can be implemented to solve this compound Poisson mixture modeling problem for the training of the Bayesian classifier on texture classification.

Table 5
The results of the texture classification on the eight gray images.

DC	RC1	RC2	RC3	RC4	RC5	RC6	RC7	RC8	CAR (%)
DC1	254	2	0	0	0	0	0	0	99.22
DC2	1	249	0	0	6	0	0	0	97.27
DC3	0	0	256	0	0	0	0	0	100
DC4	5	5	1	237	6	0	0	2	92.58
DC5	0	22	0	0	234	0	0	0	91.41
DC6	0	0	0	0	0	256	0	0	100
DC7	0	0	0	0	0	0	256	0	100
DC8	0	1	0	0	10	0	0	245	95.70

Here DCi represents data class i, RCi represents resulted class i and CAR represents classification accuracy rate.

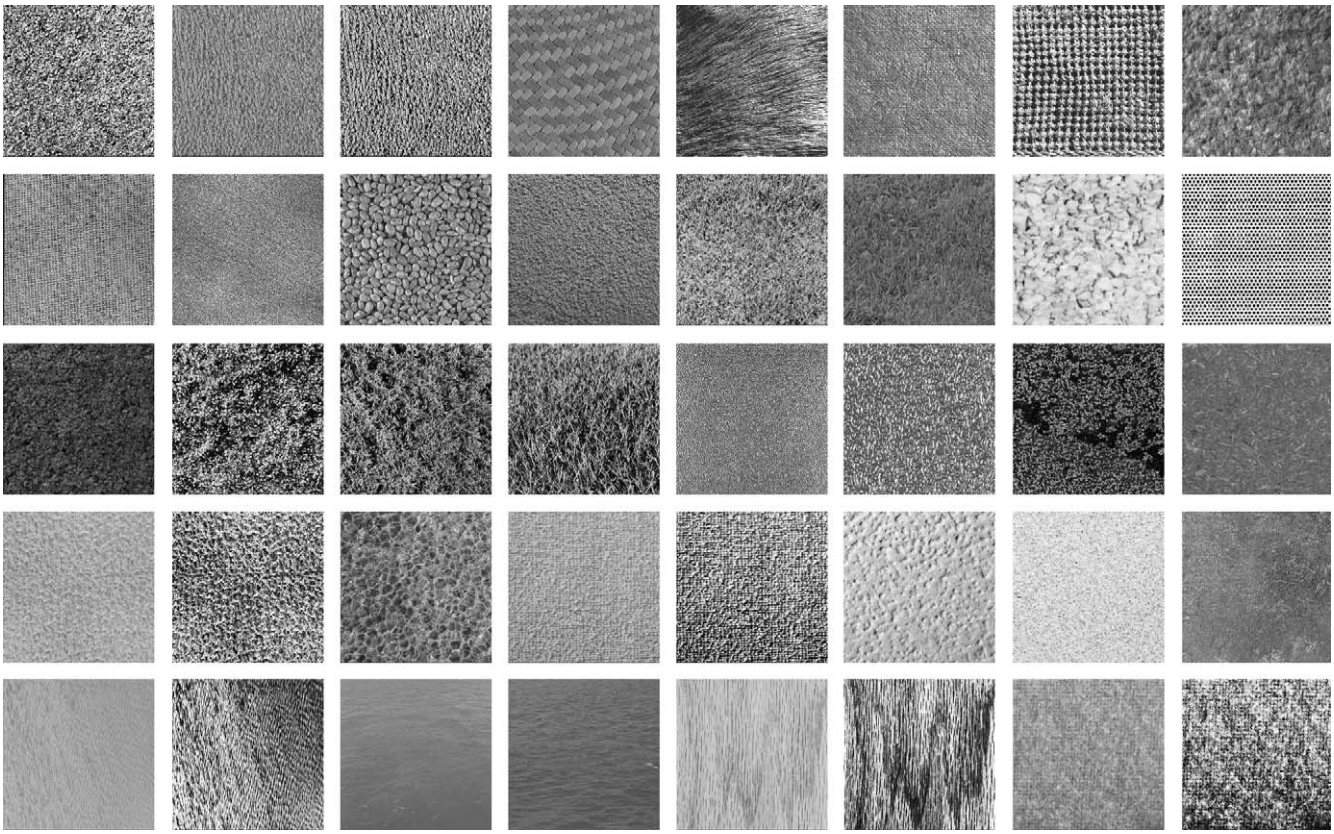

Fig. 4. The 40 gray images used in [45,46].

5.2. Experimental results and comparisons

To test the effectiveness of the adaptive gradient BYY learning algorithm on the Poisson mixture based texture classification, we apply it to learning each $P(W_i|\Theta_{i,c})$ for eight typical gray texture images (denoted by D03, D04, D29, D51, D09, D32, D38 and D24) with 128×128 pixels and 0-255 gray levels from the Brodatz image base, which are, respectively, shown in Fig. 3. So, the number of texture classes is eight, i.e., $C = 8$.

We begin to randomly pick up 128 image patches of 16×16 pixels from each texture image, and compute their feature vectors to form a training data set. On the other hand, we randomly pick up the other 256 image patches of 16×16 pixels from each texture image and compute their feature vectors to form a testing data set. During our training, if some component W_i is always 0 for every texture class, it is obvious that these $P(W_i|\Theta_{i,c})$ have no influence on the texture classification and we need not consider these degenerated but unchanged Poisson mixtures. So, the number of gray levels for the Poisson mixture modeling may be smaller than 256. As a matter of fact, this number is only 121 in this case, i.e., $G=121$. For the initial parameter setting and stop criterion, we follow the same rules as those in the simulation experiments given previously. Particularly, we set $k = 4$ for all the Poisson mixtures in consideration. As for the annihilation mechanism in this case, we always set $\varepsilon = 0.01$.

After the implementation of the adaptive gradient BYY learning algorithm on each of the eight training data sets, we have estimated 121 Poisson mixtures which indeed have different numbers of Poisson components. This can be seen from the statistical histograms of the Poisson component number based on the 121 estimated Poisson mixtures in each case of the eight texture classes. For every component X_i in each of the eight cases, there are at most three Poisson components in the estimated mixture. Thus, the initial setting of $k=4$ is large enough. Actually, it is demonstrated by the experiments that the algorithm returns the same results as $k > 4$. With all these estimated Poisson mixtures, we can obtain the trained Bayesian classifier and check it on the testing data set. Specifically, the results of the

Table 6
The comparison results of the BYY learning method and the spectral histograms method.

Size of patches	Average classification accuracy rates of the two methods	
	Spectral histogram method (%)	BYY learning method (%)
24×24	89.65	90.48
32×32	92.50	94.65

texture classification on the testing data set from a general training experiment are listed in Table 5.

From those results, we can find that the total accuracy rate of texture classification on the whole eight texture classes is 97.2%, but the accuracy rates of texture classification on the first, second, third, sixth and seventh classes are quite high and three of them even reach at 100%, that is, all the testing samples in these three classes are classified correctly. Though the texture classification accuracy rates on the other three classes are relatively low (92.58%, 91.42% and 95.70%, respectively), they are still acceptable because there really exists certain similarity between the textures of some two images (like d04 and d09), especially in certain local regions.

For comparison, we select two current state-of-the-art approaches on texture classification. One is the spectral histogram method [45] which employs the spectral histogram of sub-band images obtained from a bank of given filters on an image window as the feature statistic for texture classification. The other is the bit-plane probability (BP) signature method [46] which models the wavelet sub-band histograms of an image window as the product Bernoulli distributions (PBD) and uses the bit-plane probabilities or parameters in these PBD models, i.e., the BP signatures, as the feature statistic for texture classification. For convenience, we consider the set of 40 gray 256×256 images from the Brodatz image base used in both [45,46], which are, respectively, shown in Fig. 4. Actually, this gray image or texture set is challenging because there are significant variations within some texture and some of them are very similar to each other.

In comparison with the spectral histogram method, we randomly pick up two disjoint sets of 256 image patches from each texture image to form a training and test data sets, respectively. As suggested in [45], we test on two sizes of patches. One size is 24×24 pixels, while the other is 32×32 pixels. As there are 40 gray images, $G=256$. The average classification accuracy rates of our BYY learning method on the two sizes of patches over 100 trials are listed in Table 6, being with those of the spectral histogram method obtained in [45]. It can be found from Table 6 that our BYY learning method is slightly better than the spectral histogram method on texture classification.

In order to compare our BYY learning method with the BP signature method on these 40 gray images, we conduct the above BYY learning experiments with the image patches of 48×48 pixels and find out that the average classification accuracy rate reaches at 97.59%. According to the fact that the classification accuracy rate of a texture classification method increases with the size of image patches, the average classification accuracy rate of our BYY learning method on the image patches of 128×128 pixels should be larger than 97.59%, which is considerably higher than 96.8%, the average classification accuracy rate of the BP signature method for these 40 gray images on the image patches of 128×128 pixels obtained in [46]. Thus, we can believe that our BYY learning method is more efficient than the BP signature method on texture classification.

In summary, our adaptive gradient BYY learning algorithm is able to automatically determine the number of actual Poisson components in the real-world data from a gray image and construct a good Poisson mixture model for them, which can be successfully applied to the texture classification and even better than the two current state-of-the-art texture classification methods.

6. Conclusions

We have applied the Bayesian Ying–Yang harmony learning mechanism to the Poisson mixture modeling for parameter learning with automated model selection by constructing an adaptive gradient BYY learning algorithm for Poisson mixture. It is demonstrated by the simulation experiments that with a sample data set, our proposed adaptive gradient learning algorithm not only determines the number of actual Poisson components automatically during parameter learning, but also obtains a good estimation of the parameters in the original Poisson mixture, as long as those actual Poisson components are separated in a certain degree. Moreover, it outperforms the unsupervised learning algorithm for Poisson mixture on model selection. Moreover, the adaptive gradient learning algorithm can be used to build an efficient Bayesian classifier for texture classification on gray images.

Acknowledgments

This work was supported by the Natural Science Foundation of China for Projects 60471054 and 60771061. A preliminary version of this work or the algorithm was presented at the International Conference on Intelligent Computing (ICIC'07), August 21–24, 2007, Qingdao, China, LNCS, vol. 4681, pp. 1059–1069.

References

[1] G.J. Mclachlan, K.E. Basford, Mixture Models: Inference and Applications to Clustering, Marcel Dekker, New York, 1988.
[2] D.M. Titterington, A.F.M. Smith, U.E. Makov, Statistical Analysis of Finite Mixture Distributions, Wiley, New York, 1985.
[3] S. Frühwirth-Schnatter, Finite Mixture and Markov Switching Models, Springer Series in Statistics, Springer, New York, 2005.
[4] V.T. Farewell, D.A. Sprott, The use of a mixture model in the analysis of count data, Biometrics 44 (1988) 1191–1194.
[5] D.K. Pauler, M.D. Escobar, J.A. Sweeney, J. Greenhouse, Mixture models for eye-tracking data: a case study, Statistics in Medicine 15 (1996) 1365–1376.
[6] S. Gunduz, O.M. Tamer, A Poisson model for user accesses to web pages, in: Proceedings of the Eighteenth International Symposium on Computer and Information Sciences ISCIS'2003, Lecture Notes in Computer Science, vol. 2869, Springer, Berlin, pp. 332–339.
[7] J. Li, H. Zha, Two-way Poisson mixture models for simultaneous document classification and word clustering, Computational Statistics & Data Analysis 50 (1) (2006) 163–180.
[8] T. Brijs, T. Karlis, G. Swinnen, et al., A multivariate Poisson mixture model for marketing applications, Statistica Neerlandica 58 (3) (2004) 322–348.
[9] D.R. Carmichael, S.J. Clarke, L.M. Linnett, Spatial models for texture classification, in: Proceedings of IEE Colloquium on Texture Classification: Theory and Application, London, October 1994, pp. 9/1–9/5.
[10] W. Feller, On a general class of contagious distributions, The Annals of Mathematical Statistics 14 (1943) 389–400.
[11] D. Karlis, E. Xekalaki, Mixed Poisson distributions, International Statistical Review 73 (1) (2005) 35–58.
[12] K. Pearson, Contribution to the mathematical theory of evolution, Philosophical Transactions of the Royal Society of London Series A 185 (1894) 71–110.
[13] B.S. Everitt, Mixture distributions, in: S. Kotz, N.L. Johnson (Eds.), Encyclopedia of Statistical Sciences, vol. 5, Wiley, New York, 1985, pp. 559–569.
[14] R.A. Render, H.F. Walker, Mixture densities, maximum likelihood and the EM algorithm, SIAM Review 26 (2) (1984) 195–239.
[15] D. Karlis, An EM algorithm for multivariate Poisson distribution and related models, Journal of Applied Statistics 30 (1) (2003) 63–77.
[16] J. Ma, L. Xu, M.I. Jordan, Asymptotic convergence rate of the EM algorithm for Gaussian mixtures, Neural Computation 12 (12) (2000) 2881–2907.
[17] J. Ma, L. Xu, Asymptotic convergence properties of the EM algorithm with respect to the overlap in the mixture, Neurocomputing 68 (2005) 105–129.
[18] J. Ma, S. Fu, On the correct convergence of the EM algorithm for Gaussian mixtures, Pattern Recognition 138 (12) (2005) 2602–2611.
[19] W.C. Parr, Minimum distance estimation: a bibliography, Communications in Statistics Part A 10 (1981) 1205–1224.
[20] J.A. Hartigan, Distribution problems in clustering, in: J. Garrett (Ed.), Classification and Clustering, Academic Press, New York, 1977, pp. 45–72.
[21] H. Akaike, A new look at statistical model identification, IEEE Transactions on Automatic Control AC-19 (1974) 716–723.

[22] G. Scharz, Estimating the dimension of a model, The Annals of Statistics 6 (1978) 461–464.
[23] H. Bozdogan, Model selection and Akaike's information criterion: the general theory and its analytical extensions, Psychometrika 52 (1987) 345–370.
[24] J. Rissanen, Modeling by shortest data description, Automatica 14 (1978) 465–471.
[25] C. Wallace, D. Dowe, Minimum message length and Kolmogorov complexity, Computer Journal 42 (4) (1999) 270–283.
[26] S. Self, K. Liang, Asymptotic properties of maximum likelihood estimators and likelihood ratio tests under nonstandard conditions, Journal of the American Statistical Association 82 (1987) 605–610.
[27] D. Karlis, E. Xekalaki, On testing for the number of components in a mixed Poisson model, Annals of the Institute of Statistical Mathematics 51 (1) (1999) 149–162.
[28] P. Schlattmann, On bootstrapping the number of components in finite mixtures of Poisson distributions, Statistics and Computing 15 (3) (2005) 179–188.
[29] M.D. Escobar, M. West, Bayesian density estimation and inference using mixtures, Journal of the American Statistical Association 90 (430) (1995) 577–588.
[30] S. Recgardson, P.J. Green, On Bayesian analysis of mixtures with an unknown number of components, Journal of the Royal Statistical Society B 59 (4) (1997) 731–792.
[31] M.A.T. Figueiredo, A.K. Jain, Unsupervised learning of finite mixture models, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (3) (2002) 381–396.
[32] L. Xu, Ying–Yang machine: a Bayesian–Kullback scheme for unified learnings and new results on vector quantization, in: Proceedings of 1995 International Conference on Neural Information Processing (ICONIP'95), vol. 2, pp. 977–988.
[33] L. Xu, Bayesian Ying–Yang machine, clustering and number of clusters, Pattern Recognition Letters 18 (1997) 1167–1178.
[34] L. Xu, Best harmony, unified RPCL and automated model selection for unsupervised and supervised learning on Gaussian mixtures, three-layer nets and ME-RBF-SVM models, International Journal of Neural Systems 11 (1) (2001) 43–69.
[35] L. Xu, BYY harmony learning, structural RPCL, and topological self-organizing on mixture modes, Neural Networks 15 (2002) 1231–1237.
[36] J. Ma, T. Wang, L. Xu, A gradient BYY harmony learning rule on Gaussian mixture with automated model selection, Neurocomputing 56 (2004) 481–487.
[37] J. Ma, L. Wang, BYY harmony learning on finite mixture: adaptive gradient implementation and a floating RPCL mechanism, Neural Processing Letters 24 (1) (2006) 19–40.
[38] J. Ma, B. Gao, Y. Wang, Q. Cheng, Conjugate and natural gradient rules for BYY harmony learning on Gaussian mixture with automated model selection, International Journal of Pattern Recognition and Artificial Intelligence 19 (5) (2005) 701–713.
[39] J. Ma, X. He, A fast fixed-point BYY harmony learning algorithm on Gaussian mixture with automated model selection, Pattern Recognition Letters 29 (6) (2008) 701–711.
[40] J. Ma, J. Liu, The BYY annealing learning algorithm for Gaussian mixture with automated model selection, Pattern Recognition 40 (2007) 2029–2037.
[41] N.L. Johnson, S. Kotz, N. Balakrishnan, Discrete Multivariate Distributions, Wiley, New York, 1997.
[42] J. Ma, Automated model selection (AMS) on finite mixtures: a theoretical analysis, in: Proceedings of the 2006 International Joint Conference on Neural Networks (IJCNN), July 16–21, 2006, Vancouver, Canada, pp. 8255–8261.
[43] M. Tuceryan, A.K. Jain, Texture analysis, in: C.H. Chen, L.P. Pau, P.S.P. Wang (Eds.), Handbook of Pattern Recognition and Computer Vision, World Scientific, Singapore, 1993, pp. 235–276.
[44] B.D. Ripley, Spatial Statistics, Wiley, New York, 1981.
[45] X. Liu, D. Wang, Texture classification using spectral histograms, IEEE Transactions on Signal Processing 12 (6) (2003) 661–670.
[46] S.K. Choy, C.S. Tong, Statistical properties of bit-plane probability model and its application in supervised texture classification, IEEE Transactions on Signal Processing 17 (8) (2008) 1399–1405.

About the Author—JINWEN MA received the M.S. degree in applied mathematics from Xi'an Jiaotong University in 1988 and the Ph.D. degree in probability theory and statistics from Nankai University in 1992. From July 1992 to November 1999, he was a lecturer or associate professor at the Department or Institute of Mathematics, Shantou University. From December 1999, he became a full professor at the Institute of Mathematics, Shantou University. From September 2001, he has joined the Department of Information Science at the School of Mathematical Sciences, Peking University, where he is currently a full professor and Ph.D. advisor. During 1995 and 2003, he also visited several times at the Department of Computer Science and Engineering, the Chinese University of Hong Kong as a Research Associate or Fellow. He also worked as Research Scientist at Amari Research Unit, RIKEN Brain Science Institute, Japan from September 2005 to August 2006. He has published over 100 academic papers on neural networks, pattern recognition, bioinformatics and information theory.

About the Author—JIANFENG LIU received the B.S. degree in 2004 and M.S. degree in 2007 from the Department of Information Science at the School of Mathematical Sciences, Peking University. Currently, he is working at a software research institute. His main interests includes pattern recognition, learning theory and algorithm, and bioinformatics.

About the Author—ZHIJIE REN received the B.S. degree in 2007 from the Department of Information Science at the School of Mathematical Sciences, Peking University. Currently, she is a graduate student at the same department. Her main interests includes pattern recognition, learning theory and algorithm.