View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Springer - Publisher Connector

Pattern Anal Applic (2013) 16:55–67
DOI 10.1007/s10044-011-0256-4

THEORETICAL ADVANCES

# Expectation-maximization algorithms for inference in Dirichlet processes mixture

**T. Kimura · T. Tokuda · Y. Nakada · T. Nokajima · T. Matsumoto · A. Doucet**

Received: 25 April 2010 / Accepted: 12 November 2011 / Published online: 6 December 2011
© The Author(s) 2011. This article is published with open access at Springerlink.com

**Abstract** Mixture models are ubiquitous in applied science. In many real-world applications, the number of mixture components needs to be estimated from the data. A popular approach consists of using information criteria to perform model selection. Another approach which has become very popular over the past few years consists of using Dirichlet processes mixture (DPM) models. Both approaches are computationally intensive. The use of information criteria requires computing the maximum likelihood parameter estimates for each candidate model whereas DPM are usually trained using Markov chain Monte Carlo (MCMC) or variational Bayes (VB) methods. We propose here original batch and recursive expectation-maximization algorithms to estimate the parameters of DPM. The performance of our algorithms is demonstrated on several applications including image segmentation and image classification tasks. Our algorithms are computationally much more efficient than MCMC and VB and outperform VB on an example.

**Keywords** Clustering · Dirichlet processes · Expectation-maximization · Finite mixture models

T. Kimura · T. Tokuda · T. Nokajima · T. Matsumoto
Department of Electrical Engineering and Bioscience, Waseda University, Tokyo, Japan

Y. Nakada
College of Science and Engineering, Aoyama Gakuin University, Tokyo, Japan

A. Doucet (✉)
Departments of Computer Science and Statistics, University of British Columbia, Vancouver, BC, Canada
e-mail: arnaud@cs.ubc.ca

## 1 Introduction

Finite mixture models are used in numerous applications for density estimation and model-based clustering [14]. In many cases, the number of components of the mixture is unknown and needs to be estimated from the data. Two popular approaches have been developed to address this problem in the literature.

The standard approach consists of performing model selection using an information criterion such as Akaike information criterion or Bayesian information criterion. This requires computing the maximum likelihood estimates of the parameters for each model candidate. This is typically performed using the celebrated expectation-maximization (EM) algorithm [7] which allows us to find easily local maxima of the likelihood function. However, if the number of model candidates is large, then this approach is expensive.

Over the past few years, an alternative approach has become very popular in machine learning and pattern recognition. It relies on the class of Dirichlet process mixture (DPM) models. In this approach, a prior on the number of components of the mixture is implicitly introduced through the so-called stick-breaking construction [4, 10]. DPM models have attractive properties but are unfortunately difficult to learn and inference is typically carried out using VB [4] or Markov chain Monte Carlo (MCMC) methods [10]. Both approaches are very computationally intensive.

The main contributions of our paper is to present here original batch and recursive EM algorithms for parameter estimation in DPMs which allows us to do jointly parameter and model selection. We additionally propose an original method to select automatically the scale parameter of the DPM model which has a crucial influence on the inference results.

Batch EM algorithms need to compute an expectation w.r.t the whole data set before updating the parameters and

can be quite computationally intensive for large data sets. Our batch EM algorithm for DPM is no exception. To mitigate this problem, recursive EM ideas have appeared over the past few years where we compute an expectation w.r.t a single data point before updating the parameters [12, 17, 18, 20]. The algorithms discussed in these references have enjoyed some successes but suffer from several drawbacks. In particular, if the parameters we are interested in are constrained to a manifold—e.g., the simplex or the space of positive definite matrices—then the update rules described in earlier work rely on some complex reprojection steps. The recursive EM algorithm for DPMs proposed here follows the alternative approach initiated in [1, 16] for standard finite mixture of Gaussians. Such recursive EM approaches have surprisingly not been widely adopted, whereas they do not suffer from the problems encountered by the algorithms presented in [12, 17, 18, 20] and are a direct extension of the batch EM algorithm. We demonstrate these EM algorithms on various datasets and show that they can outperform state-of-the-art variational Bayesian approaches developed for DPM for a fraction of their computational complexity [4]. To be precise, we should mention that our EM algorithms are only applicable to a truncated version of the DPM model where the number of possible components is restricted to a large number. Similar truncation approaches have been adopted in [4, 10].

The rest of this paper is organized as follows: in Sect. 2, we review the class of finite mixture models and the batch and recursive EM algorithm in this framework. In Sect. 3, we present the class of DPM models and present original batch and recursive EM algorithms to fit a truncated version of the DPM models. We demonstrate the performance of our model and algorithms in Sect. 4.

## 2 Finite mixture models and EM algorithms

### 2.1 Finite mixture model

Let $y_1, \ldots, y_T \in \mathcal{Y}$ be independent and identically distributed random variables. We model the distribution of the observations using a parametric family of pdfs $\{g(y|\Phi); \Phi \in \mathbf{\Phi}\}$. We assume that $g(y|\Phi)$ is a mixture of $k$ components

$$g(y|\Phi) = \sum_{i=1}^{k} \pi_i f_i(y|\theta_i)$$

where $\Phi = (\pi_1, \ldots, \pi_k, \theta_1, \ldots, \theta_k)$, $f_i(y;\theta_i)$ the probability density function of the $i$th component, $\pi_i \geq 0$ and $\sum_{i=1}^{k} \pi_i = 1$. Further, we will introduce the missing data $x_1, \ldots, x_T$ where $x_n \in \{1, \ldots, k\}$ corresponds to the component associated with $y_n$; that is, we have

$$g(y|\Phi) = \sum_{i=1}^{k} f(x = i, y|\Phi)$$

where

$$f(x, y|\Phi) = \pi_x f_x(y|\theta_x).$$

We give here two illustrative examples:

*Example 1* Mixture of multivariate Gaussians.
For observations in $\mathcal{Y} = \mathbb{R}^d$, we have

$$g(y|\Phi) = \sum_{i=1}^{k} \pi_i \mathcal{N}(y; m_i, \Sigma_i)$$

where $\mathcal{N}(z; m, \Sigma)$ is the multivariate Gaussian density of argument $z$, mean $m$ and covariance $\Sigma$. In this case, we have $\Phi = \{(\pi_i, m_i, \Sigma_i);\ i = 1, \ldots, k\}$ and

$$\begin{aligned}\log f(x, y|\Phi) &= \log \pi_x - \frac{n}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma_x| \\ &\quad - \frac{1}{2}(y - m_x)^{\mathrm{T}}\Sigma_x^{-1}(y - m_x) \\ &= -\frac{n}{2}\log(2\pi) + \sum_{i=1}^{k}\left(\log \pi_i - \frac{1}{2}\log|\Sigma_i| \right. \\ &\quad \left. - \frac{1}{2}y^{\mathrm{T}}\Sigma_i^{-1}y - \frac{1}{2}m_i^{\mathrm{T}}\Sigma_i^{-1}m_i - y^{\mathrm{T}}\Sigma_i^{-1}m_i\right) \cdot \delta_i(x)\end{aligned}$$

where $\delta_i(x) = 1$ if $x = i$ and 0 otherwise.

*Example 2* Mixture of Poisson distributions.
We have $\mathcal{Y} = \mathbb{N}$ and

$$g(y|\Phi) = \sum_{i=1}^{k} \pi_i \mathcal{P}(y; \lambda_i)$$

where $\mathcal{P}(z; \lambda)$ is the Poisson distribution of argument $z$ and mean $\lambda$. In this case we have

$$\begin{aligned}\log f(x, y|\Phi) &= \log \pi_x - y\log(\lambda_x) - \log(y!) - \lambda_x \\ &= -\log(y!) + \sum_{i=1}^{k}(\log \pi_i - y\log(\lambda_i) - \lambda_i) \cdot \delta_i(x)\end{aligned}$$

In the rest of the paper, we will limit ourselves to the multivariate Gaussian case but the algorithms presented later on can be applied to any scenario where $f_i(y;\ \theta_i)$ belongs to the exponential family.

### 2.2 Batch and recursive EM algorithms for finite mixture models

#### 2.2.1 Batch EM

For any generic sequence $\{z_t\}$, we use the notation $z_{i:j} = (z_i, z_{i+1}, \ldots, z_j)$. Consider we are interested in maximizing the likelihood of the observations $y_{1:T}$. The EM algorithm

is an iterative algorithm which proceeds as follows at iteration $j$:

$$\Phi^{(j)} = \arg\max_{\Phi \in \mathbf{\Phi}} Q(\Phi, \Phi^{(j-1)})$$

where

$$Q(\Phi, \Phi^{(j-1)}) = \mathbb{E}[\log f(x_{1:T}, y_{1:T}|\Phi)|y_{1:T}, \Phi^{(j-1)}].$$

We can also easily modify this EM algorithm to maximize the penalized likelihood associated with a prior distribution $p(\Phi)$.

*Example 1* (continued) We have

$$Q(\Phi,\Phi^{(j-1)}) = \sum_{t=1}^{T}\sum_{i=1}^{k}\left(\log \pi_i - \frac{1}{2}\log|\Sigma_i| - \frac{1}{2}y_t^{\mathrm{T}}\Sigma_i^{-1}y_t - \frac{1}{2}m_i^{\mathrm{T}}\Sigma_i^{-1}m_i - y_t^{\mathrm{T}}\Sigma_i^{-1}m_i\right)\cdot \Pr(x_t = i|y_t, \Phi^{(j-1)})$$

where

$$\Pr(x_t = i|y_t, \Phi^{(j-1)}) = \frac{\pi_i^{(j-1)}\cdot \mathcal{N}\left(y_t; m_i^{(j-1)}, \Sigma_i^{(j-1)}\right)}{\sum_{l=1}^{k}\pi_l^{(j-1)}\cdot \mathcal{N}\left(y_t; m_l^{(j-1)}, \Sigma_l^{(j-1)}\right)}.$$

In this case, the $Q$ function is maximized for

$$\pi_i^{(j)} = \frac{\sum_{t=1}^{T}\mathbb{E}\left[\delta_i(x_t)|y_{1:T}, \Phi^{(j-1)}\right]}{T} = \frac{\sum_{t=1}^{T}\Pr(x_t = i|y_t, \Phi^{(j-1)})}{T}, \quad (1)$$

$$m_i^{(j)} = \frac{\sum_{t=1}^{T}\mathbb{E}\left[y_t\delta_i(x_t)|y_{1:T}, \Phi^{(j-1)}\right]}{\sum_{t=1}^{T}\mathbb{E}\left[\delta_i(x_t)|y_{1:T}, \Phi^{(j-1)}\right]} = \frac{\sum_{t=1}^{T} y_t \Pr(x_t = i|y_t, \Phi^{(j-1)})}{T\pi_i^{(j)}}, \quad (2)$$

$$\Sigma_i^{(j)} = \frac{\sum_{t=1}^{T}\mathbb{E}\left[y_t y_t^{\mathrm{T}}\delta_i(x_t)|y_{1:T}, \Phi^{(j-1)}\right]}{\sum_{t=1}^{T}\mathbb{E}\left[\delta_i(x_t)|y_{1:T}, \Phi^{(j-1)}\right]} - m_i^{(j)}m_i^{(j)\mathrm{T}} = \frac{\sum_{t=1}^{T} y_t y_t^{\mathrm{T}} \Pr(x_t = i|y_t, \Phi^{(j-1)})}{T\pi_i^{(j)}} - m_i^{(j)}m_i^{(j)\mathrm{T}}. \quad (3)$$

### 2.2.2 Recursive EM

In a recursive framework, we want to be able to update the parameter estimate $\Phi^{(t)}$ at the time index $t$ based on the new observation $y_t$. Most of the recursive EM algorithms proposed in the literature rely on updates of the form [12, 17, 18]

$$\Phi^{(t)} = (1-\gamma_t)\Phi^{(t-1)} + \gamma_t I(\Phi^{(t-1)})\nabla \log g(y_t|\Phi)|_{\Phi^{(t-1)}} \quad (4)$$

where $I(\Phi^{(t-1)})$ is the *complete* Fisher information matrix and $\{\gamma_t\}$ is a non-decreasing stepsize sequence. The main issue with this approach is that if some components of the parameter $\Phi$ are restricted to a manifold then the update (4) does not guarantee they will remain in this manifold. This is, for example, the case for the Gaussian mixture models where $(\pi_1, \ldots, \pi_k)$ have to lie on the simplex and $(\Sigma_1, \ldots, \Sigma_k)$ have to be positive definite. To handle these problems, a standard approach requires the use of reprojection algorithms or the use of an alternative parameterization. This is not elegant and can perform poorly in practice.

The general algorithm we are proposing here is inspired from [1] (see also for a related approach [16]). It bypasses elegantly these problems by directly working with the sufficient statistics appearing in the standard batch EM algorithm. If we denote by $S(x, y)$ the sufficient statistics appearing in the EM algorithm, i.e. for multivariate Gaussian mixtures

$$S(x,y) = (\delta_1(x), \ldots, \delta_k(x), y\delta_1(x), \ldots, y\delta_k(x), yy^{\mathrm{T}}\delta_1(x), \ldots, yy^{\mathrm{T}}\delta_k(x)), \quad (5)$$

then we simply use the following modified recursive Expectation update:

$$S^{(t)} = (1-\gamma_t)S^{(t-1)} + \gamma_t\mathbb{E}[S(x_t, y_t)|y_t, \Phi^{(t-1)}]. \quad (6)$$

Then given $S^{(t)}$, we use the standard M-step of the EM algorithm to obtain $\Phi^{(t)}$. The algorithm is thus a minor modification of the standard EM algorithm and does require neither reprojection nor reparameterization.

In practice we use $\gamma_t = t^{-\alpha}$ for $0.5 < \alpha \leq 1$. This algorithm can be rewritten as a stochastic approximation algorithm minimizing the Kullback–Leibler distance over $\theta$ between the distribution of the observations and the parametric family $\{g(y|\Phi); \Phi \in \mathbf{\Phi}\}$. A proof of convergence relying on stochastic approximation is sketched in [16] (see [5] for an introduction).

*Example 1* (continued) In the multivariate Gaussian case, the sufficient statistics are given by (5) and thus the update (6) proceeds as follows: we have

$$S^{(t)} = \left(a_{1:k}^{(t)}, b_{1:k}^{(t)}, c_{1:k}^{(t)}\right)$$

where

$$a_i^{(t)} = (1-\gamma_t)a_i^{(t-1)} + \gamma_t \Pr(x_t = i|y_t, \Phi^{(t-1)}), \quad (7)$$

$$b_i^{(t)} = (1-\gamma_t)b_i^{(t-1)} + \gamma_t y_t \Pr(x_t = i|y_t, \Phi^{(t-1)}), \quad (8)$$

$$c_i^{(t)} = (1-\gamma_t)c_i^{(t-1)} + \gamma_t y_t y_t^{\mathrm{T}} \Pr(x_t = i|y_t, \Phi^{(t-1)}). \quad (9)$$

and we use

$$\pi_i^{(t)} = \frac{a_i^{(t)}}{\sum_{j=1}^{k} a_j^{(t)}}, \quad (10)$$

$$m_i^{(t)} = \frac{b_i^{(t)}}{a_i^{(t)}}, \tag{11}$$

$$\Sigma_i^{(t)} = \frac{c_i^{(t)}}{a_i^{(t)}} - m_i^{(t)} m_i^{(t)\mathrm{T}}. \tag{12}$$

## 3 Dirichlet process mixtures and EM algorithms

### 3.1 Dirichlet process mixtures

The previous examples assume that the number of components of the mixture is fixed and known. However, in practice the number of components is often unknown and needs to be estimated from the data. To achieve this, we rely on DPM models which are a very popular class of models in the literature (see for example [4, 10]). In this model, we have

$$g(y|\Phi) = \sum_{i=1}^{\infty} \pi_i f_i(y|\theta_i)$$

where the infinite sequence of weights $\{\pi_i\}$ is defined as follows: we have $\pi_1 = v_1$ and for $i > 1$

$$\pi_i = v_i \prod_{j=1}^{i-1} (1 - v_j) \tag{13}$$

where $\{v_i\}$ is an infinite sequence of i.i.d. random variables distributed according to a beta distribution of parameters $(1, \alpha)$. Here $\alpha$ is an hyperparameter such that the higher $\alpha$ the higher the number of significant components. Equation 13 corresponds to the so-called stick-breaking prior representation of the Dirichlet process [10]. In the rest of the paper, we will always consider $\alpha \geq 1$; this corresponds to selecting a bounded prior density for $v_i$. The selection of the parameter $\alpha$ has a crucial influence on the inference results and we present in the next section a principled approach to select it automatically from the data.

For sake of implementation, we will consider here a truncated Dirichlet prior where we select a large value $N$ and we set $v_N = 1$. It is shown in [10] than even for very large datasets this truncation has virtually no effect if $N$ is taken reasonably large, i.e. $N = 100$. In this case the parameter $\Phi$ of interest is given by $\Phi = (v_{1:N-1}, \theta_{1:N})$.

### 3.2 Batch and recursive EM algorithms for truncated DPMs

Inference in DPMs models is usually performed using MCMC methods [10] or VB approaches [4]. For very large datasets, these approaches remain too computationally intensive. We show here how we can simply apply the batch and recursive EM algorithms in this framework when the parameter $\alpha$ is given and then discuss a procedure to estimate it automatically from the data.

#### 3.2.1 Batch EM

Given the observations $y_{1:T}$ we want to estimate $\Phi = (v_{1:N-1}, \theta_{1:N})$. In this case, we have

$$\begin{aligned}
Q(\Phi,\Phi^{(j-1)}) \equiv & \sum_{n=1}^{T} (\log f_1(y_n|\theta_1) + \log v_1) \Pr(x_n=1|y_n,\Phi^{(j-1)}) \\
& + \sum_{k=2}^{N} \sum_{n=1}^{T} \left( \log f_k(y_n|\theta_k) + \log v_k + \sum_{l=1}^{k-1} \log(1-v_l) \right) \\
& \times \Pr(x_n=k|y_n,\Phi^{(j-1)}) \\
& + \sum_{k=1}^{N-1} (\alpha-1) \log(1-v_k) \\
= & \sum_{k=1}^{N} \left( \sum_{n=1}^{T} \log f_k(y_n|\theta_k) \Pr(x_n=k|y_n,\Phi^{(j-1)}) \right) \\
& + \sum_{k=1}^{N-1} \log v_k \left[ \sum_{n=1}^{T} \Pr(x_n=k|y_n,\Phi^{(j-1)}) \right] \\
& + \sum_{k=1}^{N-1} \log(1-v_k) \\
& \times \left[ \alpha - 1 + \sum_{l=k+1}^{N} \sum_{n=1}^{T} \Pr(x_n=l|y_n,\Phi^{(j-1)}) \right]
\end{aligned}$$

where $\equiv$ means "equal" up to an additive constant independent of the first argument of $Q$.

By maximizing this expression in $v_i$ we obtain

$$v_i^{(j)} = \frac{\sum_{n=1}^{T} \Pr(x_n = i|y_n, \Phi^{(j-1)})}{\sum_{n=1}^{T} \Pr(x_n = i|y_n, \Phi^{(j-1)}) + \alpha - 1 + \sum_{l=i+1}^{N} \sum_{n=1}^{T} \Pr(x_n = l|y_n, \Phi^{(j-1)})} \tag{14}$$

as we have set $\alpha \geq 1$. From the expression of $v_{1:N-1}^{(j)}$, we can compute the weights $\pi_{1:N}^{(j)}$ using (13). The expressions for the other parameter updates $\theta_{1:N}$ are similar to the standard finite mixture case.

#### 3.2.2 Recursive EM

The recursive EM algorithm can also be implemented straightforwardly in the DPM case. To update the parameters $\theta_{1:N}$, we use the same update equations as for the recursive EM algorithm described previously. To update $\pi^{(t)}_{1:N}$, we simply compute $a^{(t)}_{1:N-1}$ recursively in time using (7). Based on $a^{(t)}_{1:N-1}$, we obtain the estimates $v^{(t)}_{1:N-1}$ using

$$v_i^{(t)} = \frac{a_i^{(t)}}{a_i^{(t)} + \alpha - 1 + \sum_{l=i+1}^{N} a_l^{(t)}}$$

From these estimates we obtain the estimates of $\pi^{(t)}_{1:N}$ using (13).

#### 3.2.3 Learning the parameter $\alpha$

We show here that it is also possible to come up with modified EM algorithms which updates the parameter $\alpha$ by maximizing an approximation of the marginal likelihood of $\alpha$. In the batch case, this proceeds as follows:

Ideally, we would like to also implement an EM type procedure to maximize the marginal likelihood of the observations with respect to $\alpha$. Assume $\Phi$ is known for the time being, then an EM algorithm would proceed as follows:

$$\alpha^{(j)} = \arg\max_{\alpha} Q(\alpha, \alpha^{(j-1)})$$

where

$$Q(\alpha, \alpha^{(j-1)}) = \mathbb{E}[\log f(x_{1:T}, y_{1:T}|\alpha, \Phi)|y_{1:T}, \alpha^{(j-1)}, \Phi].$$

After a few calculations, we obtain

$$Q(\alpha, \alpha^{(j-1)}) = (N-1)\log\alpha + \sum_{k=1}^{N-1} \mathbb{E}[\log \mathcal{B}(C_k(x_{1:T}) + 1, C_{>k}(x_{1:T}) + \alpha)|y_{1:T}, \alpha^{(j-1)}, \Phi] + C \tag{15}$$

where $C$ is a constant independent of $\alpha$, $C_i(x_{1:T})$ is the number of latent variables equal to $i$,$C_{>i}(x_{1:T})$ the number of latent variables strictly superior to $i$ and $\mathcal{B}(u, v)$ is the Beta function defined for $u$,$v > 0$ by

$$\mathcal{B}(u, v) = \int_0^1 t^{u-1}(1-t)^{v-1}\mathrm{d}t.$$

Unfortunately, computing $Q(\alpha, \alpha^{(j-1)})$ has complexity $\mathcal{O}(T^N)$. Using the convexity of the log-Beta function [8], it is, however, possible to establish that

$$\sum_{k=1}^{N-1} \mathbb{E}[\log \mathcal{B}(C_k(x_{1:T}) + 1, C_{>k}(x_{1:T}) + \alpha)|y_{1:T}, \alpha^{(j-1)}, \Phi] \geq \sum_{k=1}^{N-1} \log \mathcal{B}(\overline{C}_k + 1, \overline{C}_{>k} + \alpha).$$

where

$$\overline{C}_k = \mathbb{E}[C_k(x_{1:T})|y_{1:T}, \alpha^{(j-1)}, \Phi],$$
$$\overline{C}_{>k} = \mathbb{E}[C_{>k}(x_{1:T})|y_{1:T}, \alpha^{(j-1)}, \Phi].$$

We propose to approximate $Q(\alpha,\alpha^{(j-1)})$ by

$$Q(\alpha, \alpha^{(j-1)}) \approx (N-1)\log\alpha + \sum_{k=1}^{N-1} \log \mathcal{B}(\overline{C}_k + 1, \overline{C}_{>k} + \alpha).$$

Finally, to maximize this approximate $Q(\alpha, \alpha^{(j-1)})$, we used a fixed-point iteration in the spirit of [15] where we update

$$\alpha \leftarrow \frac{N-1}{\sum_{k=1}^{N-1} \Psi(\overline{C}_k + \overline{C}_{>k} + \alpha) - \Psi(\overline{C}_{>k} + \alpha)}$$

where $\Psi(x) = \frac{\mathrm{d}\Gamma(x)}{\mathrm{d}x}$ is the digamma function, that is the derivative of the $\log\Gamma(x)$ function.

In the batch case, we alternative update steps for $\alpha$ and standard update steps for $\Phi$. In the on-line case, we only update $\alpha$ after each pass on the data where $\Phi$ is updated.

## 4 Simulation results

In order to assess the proposed model and algorithms, it will be necessary to examine the following:

1. estimation accuracy and computational complexity
2. ability of the proposed EM for DPM model against finite mixture models
3. competitiveness against other inference methods for DP, e.g., VB
4. estimation capability of scale parameter.

The proposed batch EM algorithm for DPM performs well in the experiments reported below. It would also be worth conducting performance comparison of batch and recursive EM algorithms since the latter is an interesting alternative to the former. Specifically, the following experiments are conducted using synthetic data:

- Section 4.1 compares batch and recursive EM algorithms in terms of estimation accuracy and computational complexity. In order to make the comparison clear, the underlying mixture model is finite.
- Section 4.2.1 examines the performance of the proposed batch EM for DPM model instead of finite mixture model. In order not to blur the comparison, we first fix $\alpha$. Estimation of $\alpha$ is reported in Sect. 4.3 below.
- Section 4.2.2 compares the proposed recursive EM algorithms for DPM model with standard recursive EM algorithm for finite mixture model.
- Section 4.2.3 compares the proposed EM algorithm for DP with VB, another iterative maximization based algorithm.

After these comparisons, we use the EM algorithm with DPM to perform image segmentation. Finally, we demonstrate the performance of the EM algorithm with DPM in an image classification context.

## 4.1 Batch versus recursive EM for finite mixture models

We consider a mixture of three two-dimensional Gaussian distributions with parameters $\pi_1 = \pi_2 = 0.3$, $\pi_3 = 0.4$,

$$m_1 = \begin{pmatrix} 3 \\ 3 \end{pmatrix}, \quad m_2 = \begin{pmatrix} -3 \\ 3 \end{pmatrix}, \quad m_3 = \begin{pmatrix} 0 \\ -3 \end{pmatrix},$$
$$\Sigma_1 = \begin{pmatrix} 1 & -0.5 \\ -0.5 & 1 \end{pmatrix}, \quad \Sigma_2 = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}, \quad \Sigma_3 = \begin{pmatrix} 1 & 0 \\ 0 & 0.5 \end{pmatrix}.$$

This is a rather simple problem where the components only mildly overlap. We simulated 10,000 data points from this mixture and compare the results on the batch and recursive EM algorithms by running 1,000 realizations of both algorithms. For each realization, the initialization is random but similar for the batch and recursive EM algorithms. We run the recursive algorithm only once while we perform 30 iterations of the batch algorithms. In Fig. 1, we display the histogram in red of the log-likelihood of the parameter estimates for these 100 realizations associated with the batch algorithm for 1, 10, 20, and 30 iterations and the histogram in blue obtained using one single pass through the data for the recursive algorithm.

We see that essentially half the realizations of the recursive EM algorithm converge toward the same mode as the batch EM algorithm. The computational complexity of a pass of the batch EM is in $\mathcal{O}(Tkd^2 + kd^3)$ whereas it is in $\mathcal{O}(Tkd^3)$ for the recursive EM algorithm.[1] As the recursive EM converges much faster then this suggests that an efficient approach to perform parameter estimation for very large datasets consists of using a small run of the recursive EM algorithm and picking the parameter estimate associated with the highest likelihood instead of iterating a standard batch EM algorithm.

## 4.2 $\Sigma$ versus DPM

In many scenarios, we do not know the number of components of the mixture. In this context, we can try to either fit a standard finite mixture model with a large number $k$ of components, say $k = 100$. Alternatively, we can fit a (truncated) DPM model with say $N = 100$. The DPM model penalizes implicitly models with a large number of components, whereas the standard finite mixture model does not include such a penalty term.

### 4.2.1 Batch EM algorithm

We consider the same parameters as in the previous experiment but only 1,000 datapoints. In order to assess the performance of the proposed EM for DPM, in the first place, we perform experiments with $\alpha$ fixed first. We fit the DPM model with an empirical value $\alpha = 2$ ($\alpha = 1$ yields very similar results) which the authors had obtained through preliminary experiments and run the batch EM algorithm. Since the experiments to be reported below are successful, the next issue would be to estimate $\alpha$ automatically. In Sect. 4.3 below, $\alpha$ is estimated using the procedure discussed in Sect. 3.2.3.

After 100 iterations the parameters associated with those components with the largest mixture parameters (relabeled 1,2,3 for convenience) are given by

$$\widehat{\pi}_1 = 0.3, \quad \widehat{\pi}_2 = 0.29, \quad \widehat{\pi}_3 = 0.41,$$
$$\widehat{m}_1 = \begin{pmatrix} -3.03 \\ 2.95 \end{pmatrix}, \quad \widehat{m}_2 = \begin{pmatrix} 2.91 \\ 3.08 \end{pmatrix}, \quad \widehat{m}_3 = \begin{pmatrix} -3.74 \\ 2.96 \end{pmatrix},$$
$$\widehat{\Sigma}_1 = \begin{pmatrix} 1.32 & 0.42 \\ 0.42 & 0.83 \end{pmatrix}, \quad \widehat{\Sigma}_2 = \begin{pmatrix} 1.34 & -0.36 \\ -0.36 & 0.72 \end{pmatrix},$$
$$\widehat{\Sigma}_3 = \begin{pmatrix} 0.93 & 0 \\ 0 & 0.49 \end{pmatrix}.$$

The parameters are satisfactorily estimated. We display in Fig. 2 the estimates of $\pi_{1:100}$ and in Fig. 3 the estimates of the components of $m_{1:100}$ as a function of the iteration number.

We next consider a more difficult problem where $\pi_1 = \pi_2 = 0.3$, $\pi_3 = 0.4$,

$$m_1 = \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \quad m_2 = \begin{pmatrix} -2 \\ 2 \end{pmatrix}, \quad m_3 = \begin{pmatrix} 0 \\ -1 \end{pmatrix},$$
$$\Sigma_1 = \begin{pmatrix} 1 & -0.5 \\ -0.5 & 1 \end{pmatrix}, \quad \Sigma_2 = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}, \quad \Sigma_3 = \begin{pmatrix} 1 & 0 \\ 0 & 0.5 \end{pmatrix}.$$

Compared with the previous example, the components overlap significantly. Figure 4 demonstrates the estimated $\pi_{1:100}$. The algorithm still appears functional. This example will also be used in the following argument:

### 4.2.2 Recursive EM algorithm

We simulated 10,000 data points from the second example in the previous section with more overlap than the first one. We ran 100 realizations of both the recursive EM for finite mixture model and for DPM model. For each realization, the initialization is random but similar for the two recursive

[1] We can get also implementation of order $\mathcal{O}(Tkd^2 + kd^3)$ for the recursive EM, by using Sherman-Morrison-Woodbury formula for updating of the inverse matrix of $\Sigma_t$ and Sylvester's determinant theorem for updating of the log determinant of $\Sigma_t$.

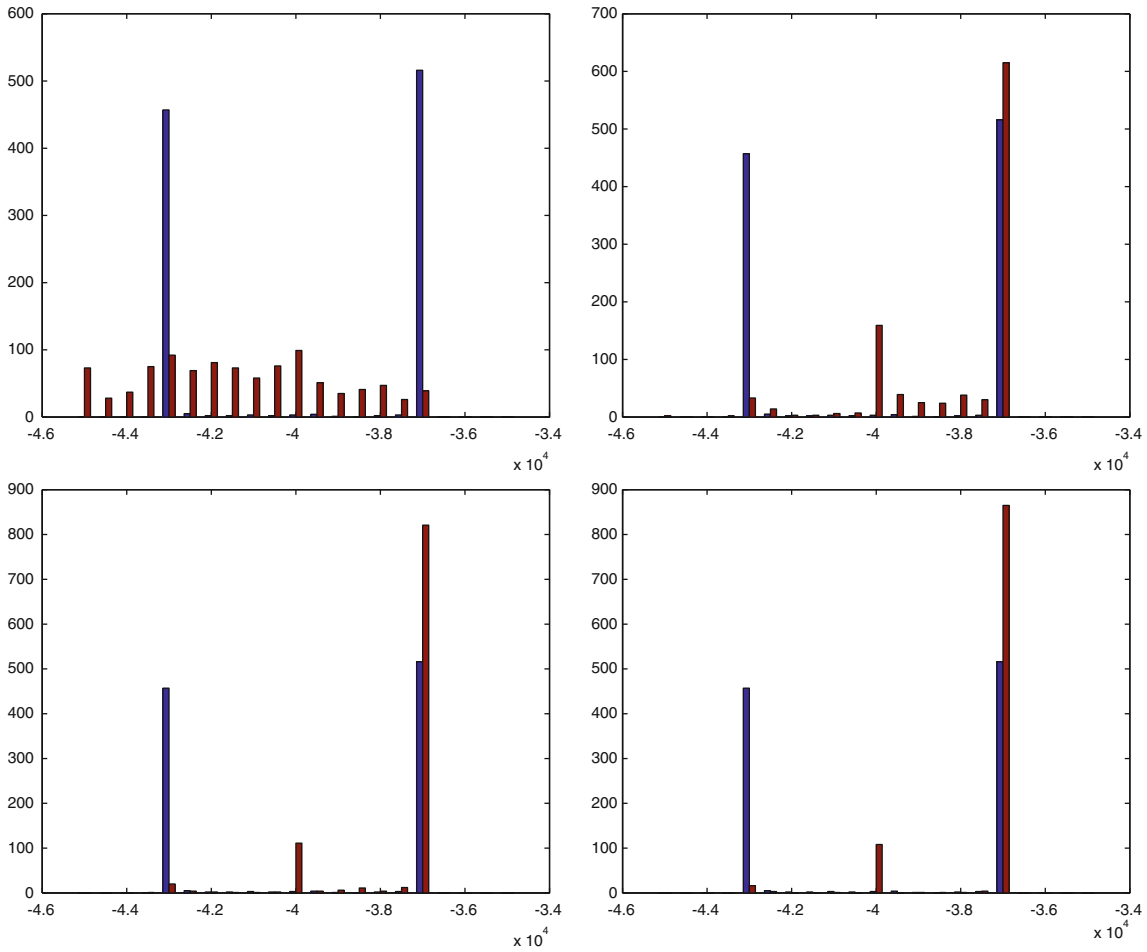
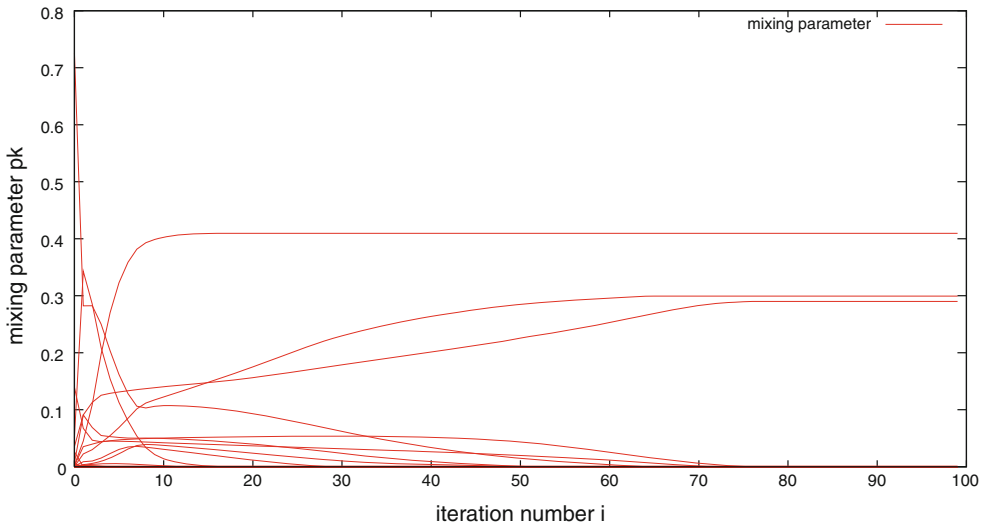



**Fig. 1** Histograms of log-likelihood of the estimates for 100 realizations: one single pass on-line EM (*blue*) versus batch EM (*red*) after 1 (*up left*), 10 (*up right*), 20 (*bottom left*) and 30 (*bottom right*) iterations

**Fig. 2** Estimates of $\pi_{1:100}$




EM algorithms. Out of these 100 runs, 35 runs of the recursive EM for DPM gave only three $\pi_i$ such that $\pi_i > 0.05$ among $\pi_{1:100}$, whereas only one run of the recursive EM for the standard finite model provided such a result. The recursive EM algorithm associated with the standard finite mixture model fails spectacularly in many

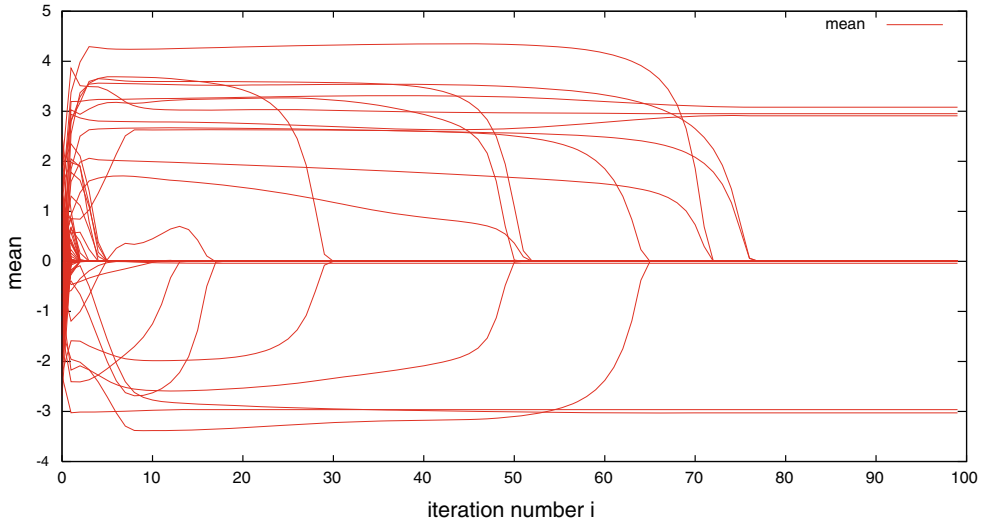



**Fig. 3** Estimates of the components of $m_{1:100}$

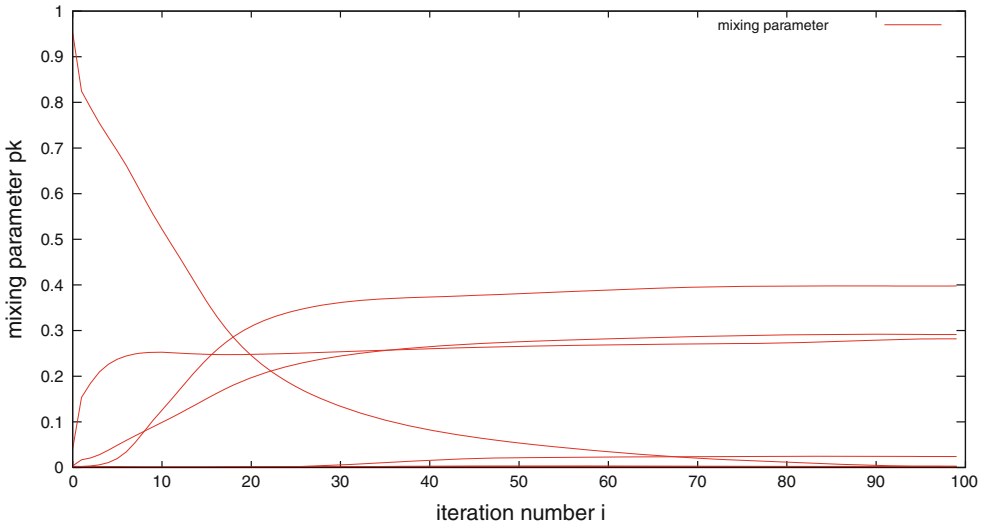



**Fig. 4** Estimates of $\pi_{1:100}$ for the mixture with more overlap

cases by fitting far too many components or a single one. The recursive EM for DPM provides far more reasonable and reliable results in all the experiments we conducted.

#### *4.2.3 EM versus variational inference for DPM*

In a Bayesian framework, we assign a prior distribution to the parameters of interest and inference relies on the associated posterior distribution. Unfortunately, the posterior distribution does not admit a closed-form expression and needs to be approximated. A standard approach consists of using iterative sampling algorithms such as MCMC methods but these techniques are computationally expensive [2]. An alternative method to approximate the posterior is variational Bayes (VB) [3, 20]. VB provides an analytical approximation to the posterior which is obtained by maximizing a lower bound of the marginal likelihood. VB has become very popular over the past few years and has been recently applied to DPM [4].

We compare here the batch EM algorithm to VB approximations for standard finite mixture of Gaussians and DPM. We apply all the algorithms to synthetic data obtained from a mixture of seven two-dimensional Gaussian distributions with parameters $\pi_1 = \pi_2 = \pi_3 = \pi_4 = \pi_5 = \pi_6 = 0.14$, $\pi_7 = 0.16$ and

$$\mu_1 = \begin{pmatrix} -5.0 \\ 0.0 \end{pmatrix}, \quad \mu_2 = \begin{pmatrix} -5.0 \\ 5.0 \end{pmatrix}, \quad \mu_3 = \begin{pmatrix} 0.0 \\ 5.0 \end{pmatrix},$$

$$\mu_4 = \begin{pmatrix} 5.0 \\ 5.0 \end{pmatrix}, \quad \mu_5 = \begin{pmatrix} 5.0 \\ 0.0 \end{pmatrix}, \quad \mu_6 = \begin{pmatrix} 5.0 \\ -5.0 \end{pmatrix}, \quad \mu_7 = \begin{pmatrix} 3.0 \\ 7.0 \end{pmatrix}$$

$$\Sigma_1 = \Sigma_2 = \begin{pmatrix} 1.0 & 0.0 \\ 0.0 & 3.0 \end{pmatrix}, \quad \Sigma_3 = \Sigma_4 = \begin{pmatrix} 3.0 & 0.0 \\ 0.0 & 1.0 \end{pmatrix}$$

$$\Sigma_5 = \Sigma_6 = \Sigma_7 = \begin{pmatrix} 1.5 & 0.5 \\ 0.5 & 3.0 \end{pmatrix}.$$

We simulated 100 data points from this mixture to estimate parameters. Initialization of the parameters and hyperparameters was the same in both algorithms, except for the prior on $\alpha$. In VB, we assumed the gamma prior for $\alpha$ with mean 2.0. We ran the batch EM algorithm with $\alpha$ set equal to 2.0. We also ran the batch and on-line EM algorithms which estimate $\alpha$ using the procedure discussed in Sect. 3.2.3. We truncate the DPM model at $N = 100$ for both the VB and EM algorithms. For the joint prior distribution on the mean and covariance parameters, we used an inverse-Wishart normal.

We assess the performance of the EM and VB algorithms by comparing the Kullback–Leibler distance between the true distribution and the observations $p(y|\Phi_{\text{true}})$ and the predictive distribution $p(y|y_{1:T})$ estimated through the algorithms. We have

$$\text{KL} = \int p(y|\Phi_{\text{true}}) \log \frac{p(y|\Phi_{\text{true}})}{p(y|y_{1:T})} \text{d}y \approx \frac{1}{P} \sum_{k=1}^{P} \frac{p(y_k^*|\Phi_{\text{true}})}{p(y_k^*|y_{1:T})}$$

where $\{y_k^*\}_{k=1,\ldots,P}$ are sampled from $p(y|\Phi_{\text{true}})$. We use $P = 500$ in our experiments. For the EM algorithm, we use

$$p(y_k^*|y_{1:T}) \approx p(y_k^*|\hat{\Phi})$$

where $\hat{\Phi}$ is the MAP parameter estimate. For VB, we use

$$p(y_k^*|y_{1:T}) \approx \int p(y_k^*|\Phi) q(\Phi|y_{1:T}) \text{d}\Phi,$$

where $q(\Phi|y_{1:T})$ is the variational posterior. The results are presented in Fig. 5 below. In this example, EM outperforms significantly VB.

Figure 5 shows the Kullback–Leibler distance for predictive density estimates associated with

- batch EM for DPM model with $\alpha$ fixed at 2.0 (denoted by DPMAP a = 2.0)
- batch EM for DPM model with $\alpha$ estimated (denoted by DPMAP a = Auto)
- recursive EM for DPM model with $\alpha$ estimated (denoted by DPMAP recursive)
- VB for DPM model with $\alpha$ fixed at 2.0 (denoted by DPVB a = 2.0)
- VB for DPM model with $\alpha$ learned (denoted by DPVB a = Auto).

The proposed batch and recursive EM for DPM model outperformed VB at least in this experiment. For a reference purpose, Fig. 5 also gives box plots of the Kullback–Leibler distance associated with

- batch EM for finite mixture model with the number of components fixed at 5, 10, and 20, respectively (denoted by MAP5, MAP10, and MAP20)
- VB for finite mixture model with the number of components fixed at 5, 10, and 20, respectively (denoted by VB5, VB10, and VB20).

From the experiments performed in this section, we have the following observations:

1. The proposed batch EM algorithm for DPM model captured the true parameters well as demonstrated in Sect. 4.2.1.
2. The proposed recursive EM algorithm for DPM model outperformed the standard recursive EM algorithm for finite mixture model as shown in Sect. 4.2.2.

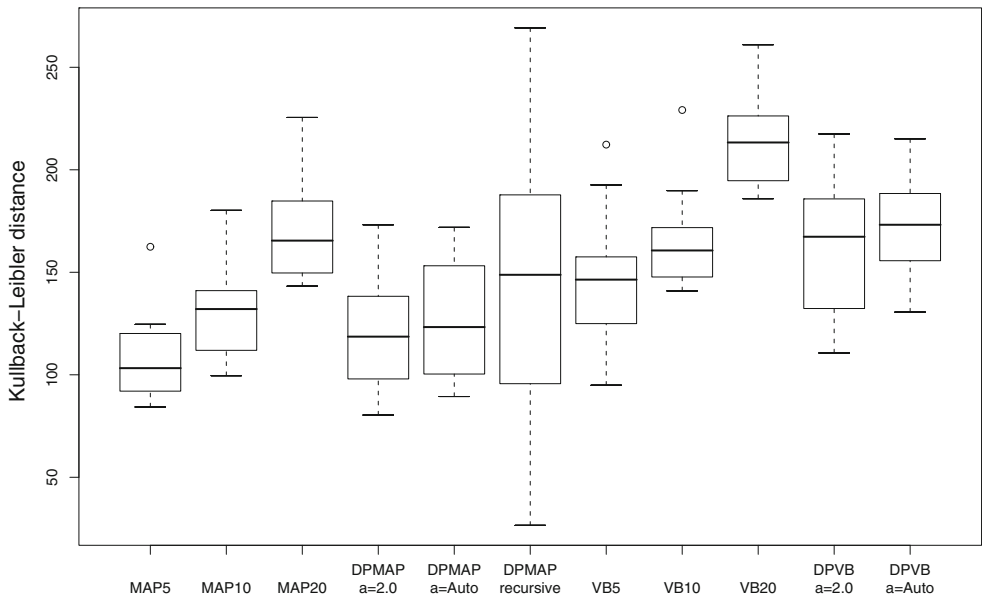



**Fig. 5** Kullback–Leibler distance for predictive density estimates using EM and VB. *MAP* denotes MAP EM, while *VB* denotes variational inference without DPM; *MAPX* denotes the fact that the number of components is fixed to X. *DPMAP* and *DPVB* indicate MAP and VB with DPM. $\alpha = 2.0$ means that $\alpha$ is fixed at 2.0, whereas $\alpha =$ Auto means that $\alpha$ is estimated from the data

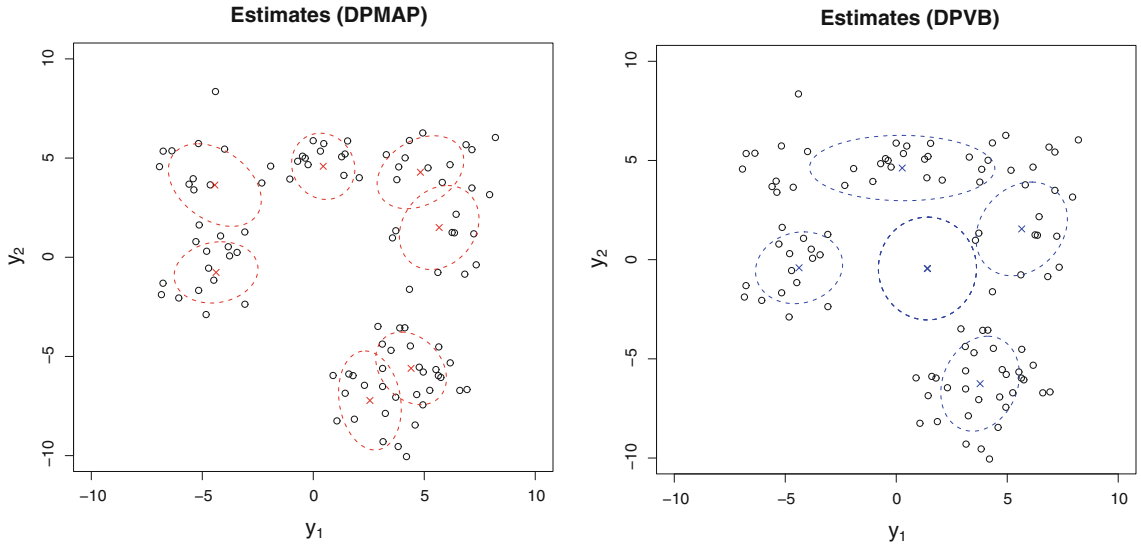
3. The proposed batch and recursive EM algorithms for DPM model outperformed VB algorithm as demonstrated in Sect. 4.2.3.
4. The batch and the recursive EM algorithms with $\alpha$ estimation are also competitive.
5. The proposed algorithm gave reasonable results on image segmentation and image classification problems.

The two figures in Fig. 6 display the estimated mixture components with DPM and VB. Crosses and dotted ellipses, respectively, indicate the mode of the mean and covariance of each component. We only display components with mixing weights $\pi_i \geq 0.0001$.

Note that when the components are truncated at $N$, there are $N$ parameter vectors, $(\pi_i, \theta_i),\ i = 1, \ldots, N$ estimated by the proposed EM algorithm. The thresholding was simply introduced to ease the presentation of the results. This same is also applied to other experiments. Note also that the VB estimate gives rise to a spurious component near the center of the dataset. Another two figures in Fig. 7 show the mixing weights estimated by DPM and VB.

### 4.3 Image segmentation

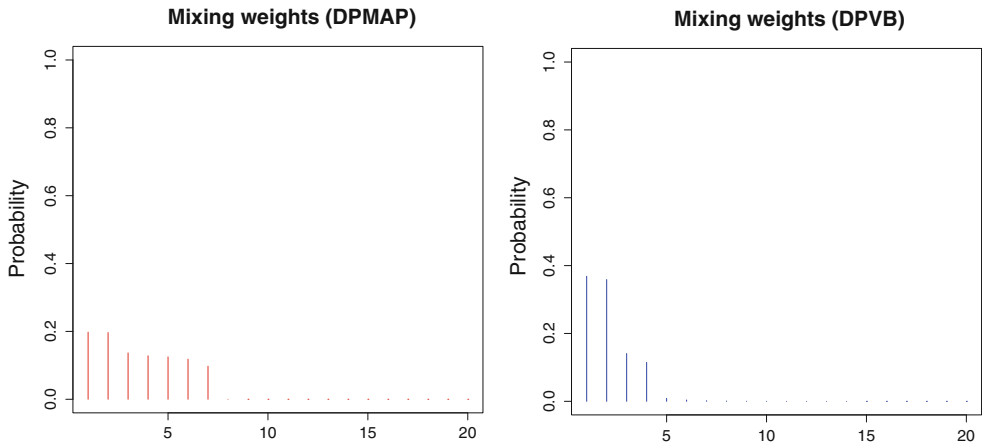
We present an application of our algorithm to image segmentation [13]. Each pixel corresponds to a point in $\mathbb{R}^3$; each component corresponds to a specific color R, G or B. We build a Gaussian mixture model for the distribution of the pixels. We then estimate the number of significant




Estimated components using EM for DPM

Estimated components using VB

**Fig. 6** Estimated components using VB




Mixing weights using EM for DPM model

Mixing weights using VB for DPM model

**Fig. 7** Mixing weights using VB for DPM model

**Fig. 8** Original image

**Fig. 9** Segmented image using DPM model

components (i.e. whose estimated mixing weights $\pi_i$ is such that $\pi_i > 0.01$) and the parameters of these components using the DPM model and the EM algorithm. Each pixel is then attributed to the component having the highest posterior probability.

We apply this algorithm to an image taken from Berkeley Segmentation Dataset (BSDS No. 253036). The size of the image is 481 × 321; hence this corresponds to 154,401 observations. After having applied the batch EM algorithm to this dataset, we identified 16 significant components. We display in the original image in Fig. 8 and its segmented version in Fig. 9.

While image segmentation algorithm often involves detailed high-level operations, the method used in the present experiment is a simple RGB component clustering, and yet it appears to have captured at least several important segmented components. Observe that the tree and the elephants are segmented in a relatively crisp manner. The algorithm also appears to have captured the segmented sky as well as the clouds. The textures associated with the foreground grasses are also discernible. More specifically, all the elephants belonged to the same class with mixing parameter 0.04 where average value of (R, G, B) is (72, 76, 68). This class also contained the tree in the center. The sky consisted of four components:

- mixing parameter = 0.34, (R, G, B) = (173, 197, 230)
- mixing parameter = 0.17, (R, G, B) = (212, 229, 242)
- mixing parameter = 0.11, (R, G, B) = (177, 190, 216)
- mixing parameter = 0.06, (R, G, B) = (236, 253, 254).

The foreground grasses consisted of two components:

- mixing parameter = 0.08, (R, G, B) = (95, 102, 57)
- mixing parameter = 0.07, (R, G, B) = (115, 113, 63).

### 4.4 Image classification

Our aim is to classify some images based on PCA-SIFT features [11]. We selected five categories from the Caltech Database [9]: airplanes, cars, faces, leaves, and motorbikes. Figure 10 shows some of the images from the dataset.

For each category $C \in \{1, 2, 3, 4, 5\}$, we approximate the distribution of the features using a Gaussian DPM model using 100 training data, i.e. we estimate a vector of parameter $\Phi_C$ using the EM algorithm. For each new image, we then extract a vector of features and compute its likelihood for each of the possible Gaussian DPM model. We select the category of the new image as the one corresponding to the highest likelihood.

For each category, we use 50 test images. Given an image, the PCA-SIFT detects 50–300 features points and a 36-dimensional vector is associated with each feature point. To reduce the computational cost and the number of parameters, we use only diagonal covariance matrices with $N = 100$ and $\alpha = 2.0$ for the mixture model. The result was compared with a standard bag-of-features model using a naive Bayes classifier where the feature vectors are discretized using a $K$-means algorithm [6]. For example, in Fig. 11, $K = 100$ means that the number of discretized features is 100.

The results are displayed in Fig. 11.

Figure 11 indicates that, at least in this example, the proposed algorithm achieves the highest accuracy. Figure 12 illustrates some of the classification results. The red circles represent PCA-SIFT features. The rightmost picture, which was incorrectly categorized by the proposed algorithm, appears to carry several feature points around the building in the back.

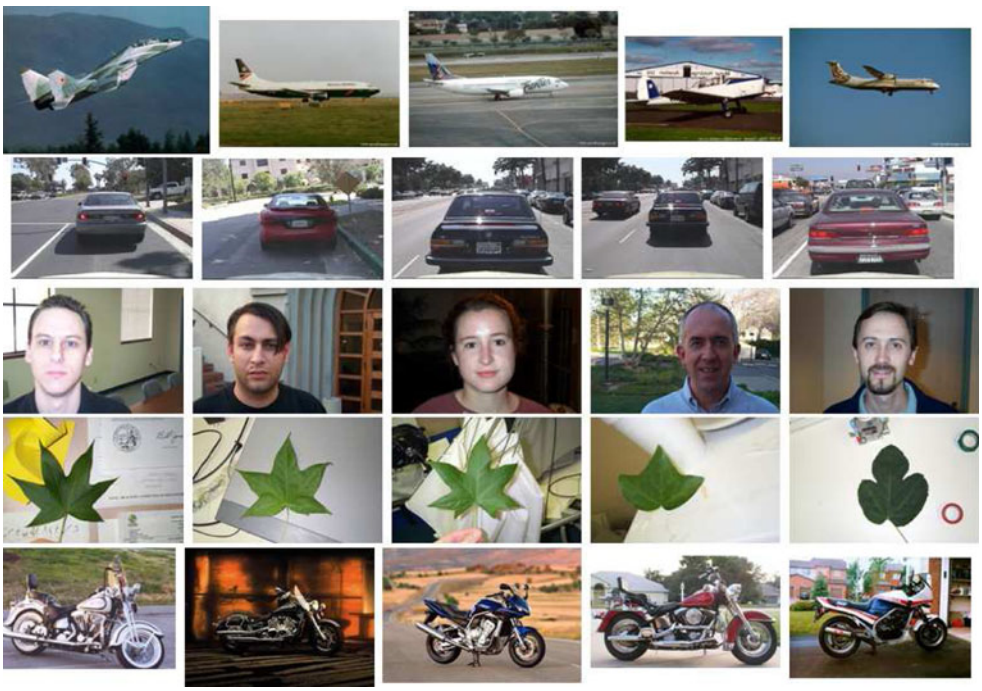


**Fig. 10** A few images from the Caltech database

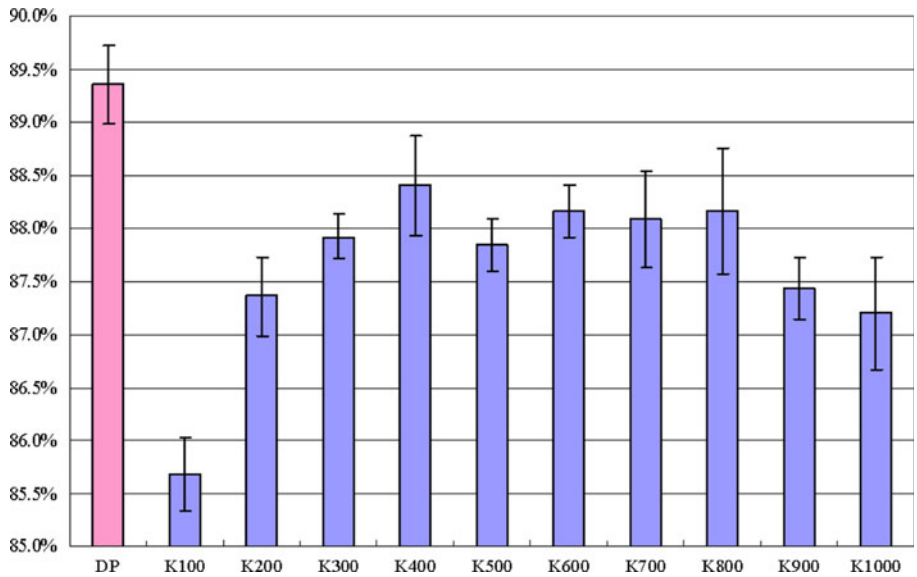



**Fig. 11** Classification accuracy of the two algorithms. DP corresponds to the proposed EM algorithm for Gaussian. $K - x$ indicates naive Bayes with $x$ discretized features


**Fig. 12** Test images: the first four are correctly classified as airplanes, the fifth one is misclassified

## 5 Discussion

DPM models have become a very popular class of statistical models to perform inference in mixture models when the number of components is unknown. Standard approaches to fit DPM rely either on MCMC or VB methods.

Note that MCMC computation requires typically several thousand iterations [2]. Our experience on VB tells us that

a typical VB needs 15–30 iterations before convergence. Since the proposed recursive EM is a single pass scheme, it is less expensive than MCMC and VB.

In order to consider the complexity of VB, note that the iteration formula for the basic parameters of VB are similar to those of batch EM given in (1)–(3) (Sect. 2). Therefore, the complexity of the covariance matrix statistics computation is of order $O(Tkd^2)$. For the computation of the test distribution $q(z)$, there are $k$ *log determinants* as well as $k$ *inverse matrix* computations. This gives rise to $O(kd^3)$. Adding those two, we have $O(Tkd^2 + kd^3)$ for the complexity of VB which is the same as the complexity of the batch EM described in Sect. 4.1.

We have proposed here original EM-type algorithms to fit DPM models. These batch and recursive EM algorithms are computationally efficient and perform well in the set of experiments we have conducted. We believe that these methods complement the current set of tools available to fit DPM and are an attractive alternative.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

## References

1. Andrieu C, Doucet A (2001) Recursive Monte Carlo algorithms for parameter estimation in general state-space models. In: Proceedings of the 11th workshop IEEE statistical signal processing, pp 14–17
2. Andrieu C, Doucet A, De Freitas N, Jordan MI (2003) An introduction to MCMC for machine learning. Mach Learn 50:5–43
3. Attias H (2000) A variational Bayesian framework for graphical models. In: Advances in neural information processing systems, vol 12. MIT Press, Cambridge
4. Blei D, Jordan MI (2005) Variational inference for Dirichlet process mixtures. Bayesian analysis, 1, pp 121–144
5. Bottou L (2004) Stochastic learning, advanced lectures on machine learning. Lecture notes in artificial intelligence, vol 3176. Springer, Berlin, pp 146–168
6. Csurka G, Dance CR, Fan L, Willamowski J, Bray C (2004) Visual categorization with bags of keypoints. In: Proceedings of the ECCV international workshop on statistical learning in computer vision, pp 1–22
7. Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the EM algorithm. J R Stat Soc B 39:1–38
8. Dragomir SS, Agarwal RP, Barnett NS (2000) Inequalities for beta and gamma functions via some classical and new integral inequalities. J Inequal Appl 5:103–165
9. Fergus R, Perona P, Zisserman A (2003) Object class recognition by unsupervised scale-invariant learning. In: Proceedings of the IEEE conference on CVPR, vol 2, pp 264–271
10. Ishwaran H, James LF (2001) Gibbs sampling methods for stick-breaking priors. J Am Stat Assoc 96:161–173
11. Ke Y, Sukthankar R (2004) PCA-SIFT: a more distinctive representation for local image descriptors. In: Proceedings of the CVPR, vol 2, pp 506–513
12. Liu Z, Almhana J, Choulakian V, McGorman R (2005) Online EM algorithm for mixture with application to internet traffic modeling. Comput Stat Data Anal 50:1052–1071
13. Martin D, Fowlkes C, Tal D, Malik J (2001) A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proceedings of the ICCV
14. McLachlan G, Peel D (2000) Finite mixture models. Wiley series in probability and statistics. John Wiley & Sons, Hoboken, NJ
15. Minka T (2000) Estimating a Dirichlet distribution. Technical report, MIT
16. Sato M, Ishii S (2000) On-line EM algorithm for the normalized Gaussian network. Neural Comput 12:407–432
17. Titterington DM (1984) Recursive parameter estimation using incomplete data. J R Stat Soc B 46:257–267
18. Titterington DM, Smith AFM, Makov UE (1985) Statistical analysis of finite mixture distributions. Wiley, New York
19. Ueda N (2002) Bayesian learning: application of variational Bayesian learning. Trans IEICE 85(8):633–638
20. Zivkovic Z, van der Heijden F (2004) Recursive unsupervised learning of finite mixture models. IEEE Trans Pattern Anal Mach Intell 26:651–656