Entropy-Based Variational Scheme for Fast Bayes Learning of Gaussian Mixtures*

Antonio Peñalver[1], Francisco Escolano[2], and Boyan Bonev[2]

[1] Miguel Hernández University, Elche, Spain
[2] University of Alicante, Spain

Abstract. In this paper, we propose a fast entropy-based variational scheme for learning Gaussian mixtures. The key element of the proposal is to exploit the incremental learning approach to perform model selection through efficient iteration over the Variational Bayes (VB) optimization step in a way that the number of splits is minimized. In order to minimize the number of splits we only select for spliting the worse kernel in terms of evaluating its entropy. Recent Gaussian mixture learning proposals suggest the use of that mechanism if a bypass entropy estimator is available. Here we will exploit the recently proposed Leonenko estimator. Our experimental results, both in 2D and in higher dimension show the effectiveness of the approach which reduces an order of magnitude the computational cost of the state-of-the-art incremental component learners.

1 Introduction

Mixture models, in particular those that use Gaussian kernels, have been widely used in areas involving statistical modeling of data like pattern recognition, computer vision, image analysis or complex probability density functions (pdfs) approximation. In statistical pattern recognition, mixture models provide a formal approach for clustering [1][2]. Mixtures model the data as being generated by one of a set of kernels. The estimation of the parameters of each kernel leads to a clustering of the data set. Whereas traditional clustering methods are based on heuristics (e.g. k-means algorithm) or hierarchical agglomerative techniques [3], mixture models allow us to address the problem of validating the parameters of a given model in a formal way. Mixture models are also suitable for representing complex class-conditional pdfs in Bayesian supervised learning scenarios [4][5] or Bayesian parameter estimation [6]. The task of estimating the parameters of a given mixture can be achieved with different approaches: maximum likelihood, maximum a posteriori (MAP) or Bayesian inference [7].

The same is true for the Bayesian Maximum a Posteriori (MAP) estimation approach that tries to find the parameters that correspond to the location of the MAP density function, and it is used when this density cannot be obtained directly [8]. Bayesian inference models the *a posteriori* parameter probability distribution, so it is assumed that the parameters are not uniquely described and they are modeled by probability density functions (pdfs) [7]. Thus, an additional set of hyperparameters is required in order to model the distribution of parameters. Then, the *a posteriori* probability of the data

* Funded by the Project TIN2008- 04416/TIN of the Spanish Government.

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 100–108, 2010.

set is obtained by integration over the probability distribution of the parameters. The task of defining proper distribution functions for parameters can be computationally heavy and may result in intractable integrals. There are some approaches that try to solve those drawbacks: Laplacian method [9], Markov Chain Monte Carlo (MCMC) [10], and Variational methods [11][12]. Laplacian methods employ an approximation based on the Taylor expansion for the expression of the integrals [9]. However, in high dimensional contexts this approach is computationally expensive and may provide poor approximation results. MCMC methods require both an appropriate distribution selection and sampling techniques in order to draw suitable data samples. Besides, due to their stochastic nature, MCMC algorithms may require a long time to converge [10]. Variational algorithms are guaranteed to provide a lower bound of the approximation error [11]. In most approaches, parameter initialization is selected randomly, defined over a given range of values, but it could lead to overfitting and poor generalization [13]. Although the results show good performance in clustering, blind signal detection or color image segmentation, the computational complexity of the Variational EM algorithm is higher than the classic EM with the maximum likelihood criterion. However, variational methods are more suitable than EM-MDL based methods [14][15][8][16] for model-order selection. Thus in this paper, we propose an fast extension of the Variational Bayes (BV) method proposed in [17] for inferring Gaussian mixtures and solving the model-order selection problem.

2 Variational Bayes for Mixtures

Given N i.i.d. samples $\mathcal{X} = \{\boldsymbol{x}^1, \ldots, \boldsymbol{x}^N\}$ of a d-dimensional random variable X, their associated hidden variables $Z = \{\boldsymbol{z}^1, \ldots, \boldsymbol{z}^N\}$ and the parameters Θ of the model, the Bayesian posterior is given by [18]:

$$p(Z,\Theta|X) = \frac{p(\Theta)\prod_{n=1}^{N} p(\boldsymbol{x}^n, \boldsymbol{z}^n|\Theta)}{\int p(\Theta)\prod_{n=1}^{N} p(\boldsymbol{x}^n, \boldsymbol{z}^n|\Theta)d\Theta} \ . \tag{1}$$

Since the integration w.r.t. Θ is analytically intractable, the posterior is approximated by a factorized distribution $q(Z,\Theta) = q(Z)q(\Theta)$ and the optimal approximation is the one that minimizes the variational free energy:

$$\mathcal{L}(q) = \int q(Z,\Theta)\log\frac{q(Z,\Theta)}{p(Z,\Theta|X)}d\Theta - \log\int p(\Theta)\prod_{n=1}^{N} p(\boldsymbol{x}^n|\theta)d\Theta \ , \tag{2}$$

where the first term is the Kullback-Leibler divergence between the approximation and the true posterior. As the second term is independent of the approximation, the Variational Bayes (VB) approach is reduced to minimize the latter divergence. Such minimization is addressed in a EM-like process alternating the updating of $q(\Theta)$ and the updating of $q(Z)$ [19]:

$$q(\Theta) \propto p(\Theta)\exp\left\{\sum_{n=1}^{N}\langle\log p(\boldsymbol{x}^n, \boldsymbol{z}^n|\Theta)\rangle_{q(Z)}\right\} \tag{3}$$

$$q(Z) \propto \exp\left\{\sum_{n=1}^{N}\langle\log p(\boldsymbol{x}^n, \boldsymbol{z}^n|\Theta)\rangle_{q(\Theta)}\right\} \tag{4}$$

When the posterior is modeled by a mixture we have that

$$p(X|\Omega) = \sum_{k=1}^{K} \pi_k p(X|\Omega_k), \tag{5}$$

where $0 \leq \pi_k \leq 1$, $k = 1, \ldots, K$, $\sum_{k=1}^{K} \pi_k = 1$, K is the number of kernels, $\pi_1, \ldots, \pi_K$ are the a priori probabilities of each kernel, and Ω_k are the parameters that describe the kernel. In Gaussian mixtures, $\Omega_k = \{\mu_k, \Sigma_k\}$, that is, the mean vector and the covariance matrix. Consequently we have

$$p(X, Z|\Theta) = \prod_{n=1}^{N} \prod_{k=1}^{K} z_k^n \pi_k p(\boldsymbol{x}^n|\Omega_k). \tag{6}$$

where $z^i = [z_1^n, \ldots, z_K^n]$ is a binary vector and $z_m^n = 1$ and $z_p^n = 0$, if $p \neq m$, denote that $\boldsymbol{x}^n$ has been generated by the kernel m. Then, considering the complete mixture let $\mu = \{\mu_k\}$, $\Sigma = \{\Sigma_k\}$, $\pi = \{\pi_k\}$ and K the parameters of the model, that is, $\Theta = \{\mu, \Sigma, \pi, K\}$. Including in the parameter set the number of mixtures implies dealing with the problem of model order selection (obtain the optimal K). In [17], model order selection is implicitly solved within the Bayesian approach. In the latter framework, it is assumed that a number of $K - s$ components fit the data well in their region of influence (*fixed components*) and then model order selection is posed in terms of optimizing the parameters of the remaing s (*free components*). Let $\alpha = \{\pi_k\}_{k=1}^{s}$ the coefficients of the free components and $\beta = \{\pi_k\}_{k=s+1}^{K}$ the coefficients of the fixed components. Obviously, the sum of coefficients in α and β must be the unit. In addition, under the i.i.d. sampling assumption, the prior distribution of Z given α and β can be modeled by a product of multinomials:

$$p(Z|\alpha, \beta) = \prod_{n=1}^{N} \prod_{k=1}^{s} {\pi_k}^{z_k^n} \prod_{k=s+1}^{K} {\pi_k}^{z_k^n} . \tag{7}$$

Moreover, assuming conjugate Dirichlet priors over the set of mixing coefficients, we have that

$$p(\beta|\alpha) = \left(1 - \sum_{k=1}^{s} \pi_k\right)^{-K+s} \frac{\Gamma\left(\sum_{k=s+1}^{K} \gamma_k\right)}{\prod_{k=s+1}^{K} \Gamma(\gamma_k)} \cdot \prod_{k=s+1}^{K} \left(\frac{\pi_k}{1 - \sum_{k=1}^{s} \pi_k}\right)^{\gamma_k - 1} . \tag{8}$$

Then, considering fixed coefficients Θ is redefined as $\Theta = \{\mu, \Sigma, \beta\}$ and we have the following factorization:

$$q(Z, \Theta) = q(Z)q(\mu)q(\Sigma)q(\beta) . \tag{9}$$

Then, in [17], the optimization of the variational free energy yields:

$$q(Z)=\prod_{n=1}^{N}\prod_{k=1}^{s}{r_{k^n}}^{z_k^n}\prod_{k=s+1}^{K}{\rho_{k^n}}^{z_k^n} \tag{10}$$

$$q(\mu)=\prod_{k=1}^{K}\mathcal{N}(\mu_k|m_k,\Sigma_k) \tag{11}$$

$$q(\Sigma)=\prod_{k=1}^{K}\mathcal{W}(\Sigma_k|\nu_k,V_k) \tag{12}$$

$$q(\beta)=\left(1-\sum_{k=1}^{s}\pi_k\right)^{-K+s}\frac{\Gamma\left(\sum_{k=s+1}^{K}\tilde{\gamma}_k\right)}{\prod_{k=s+1}^{K}\Gamma(\tilde{\gamma}_k)}\cdot\prod_{k=s+1}^{K}\left(\frac{\pi_k}{1-\sum_{k=1}^{s}\pi_k}\right)^{\tilde{\gamma}_k-1} \tag{13}$$

where $\mathcal{N}(.)$ and $\mathcal{W}(.)$ are respectively the Gaussian and Wishart densities, and the rest of parameters are obtained as specified in [17]. Furthermore, after the maximization of the free energy w.r.t. $q(.)$, it proceeds to update the coefficients in α. This interwinted process is repeated until convergence. However, how is model selection solved within this approach?

3 Model Order Selection in VB: The EBVS Approach

An incremental model order selection algorithm starts from a small number of components (one or two) and proceeds to split them until convergence. For instance, in [16] a unique initial kernel is used. However in [17] the VBgmm method [20] is used for training an initial $K=2$ model. Then, in the so called VBgmmSplit, they proceed by sorting the obtained kernels and then trying to split them recursively. Each splitting consists of: (i) removing the original component, and (ii) replacing it by two kernels with the same covariance matrix as the original but with means placed in opposite directions along the maximum variabiability direction. Such direction is given by the principal axis (eigenvector ϕ) of the inverse of the original covariance matrix and the amount of displacement, and the amount of displacement is $\pm\sqrt{\lambda}\phi$, being λ the corresponding eigenvalues. If the original mixing coefficient is π the new coefficients are $\pi/2$. A more complex and robust split method is proposed in [21] and used efficiently in [16]. Independently of the split strategy, the critical point of VBgmmSplit is the *amount of splits needed until convergence*. At each iteration of the latter algorithm the K current exisiting kernels are splited. Consider the case of any split is detected as proper (non-zero π after running the VB update described in the previous section, where each new kernel is considered as *free*). Then, the number of components increases and then a new set of splitting tests starts in the next iteration. This means that if the algorithm stops (all splits failed) with K kernels, the number of splits has been $1+2+\ldots+K=K(K+1)/2$. Although the computational cost of a split is not critical, what is critical is the increasing amount of kernels considered for the VB optimization. Thus, it is important to control the number of splits because it has an important impact in the complexity of the VB optimization step. This is our main contribution in this paper, and we dubbed it the *Entropy-based Variational Scheme* (EBVS).

3.1 The EBVS Split Scheme

Instead of considering all the current kernels K at each iteration, we split only one kernel per iteration. In order to do so, we implement a selection criterion based on measuring the entropy of the kernels. According to [16], as the Gaussian distribution maximizes entropy among all the distributions with the same covariance, the lower the entropy of a kernel, the more suitable it is for being split. The main problem of this approach is the fact that entropy must be estimated and this may be a very difficult task if data dimensionality d is high (curse of dimensionality) if a bypass entropy estimator (no need to estimate the probability density function) is not used. For instance, in [16], the Entropic Graphs based estimator [22] is extrapolated from Rényi entropy to the Shannon one. However, if ones uses the recently proposed Leonenko's estimator [23] (see above) then there is no need of extrapolation, and asymptotic consistence is ensured. This is the entropy estimator used in this paper. Then, at each iteration of the algorithm we select the *worse*, in terms of low entropy, to be split. If the split is successful we will have $K + 1$ kernels to feed the VB optimization in the next iteration. Otherwise, there is no need to add a new kernel and the process converges to K kernels. The key question here is that the overall process is linear (one split per iteration) with the number of kernels instead of being quadratic.

3.2 Entropy Estimation

A simple way to understand the k-NN entropy estimation proposed by Leonenko [23] is to look at the Shannon entropy formula $H(X) = -\int f(x) \log f(x) dx$, as an average of $\log f(x)$, being $f(x)$ an existing pdf. The estimation of $\widehat{\log f(x)}$ would allow the estimation of $\hat{H}(X) = -N^{-1} \sum_{i=1}^{N} \widehat{\log f(x)}$. For this purpose the probability distribution $P_k(\epsilon)$ of the distance between a sample x_i and its k-NN is considered. If a ball of diameter ϵ is centered at x_i and there is a point within distance $\epsilon/2$, then there are $k-1$ other points closer to x_i and $N-k-1$ points farther from it. The probability of this to happen is $P_k(\epsilon)d\epsilon = k\binom{N-1}{k}\frac{dp_i(\epsilon)}{d\epsilon}p_i^{k-1}(1-p_i)^{N-k-1}$ being p_i the mass of the ϵ-ball and $p_i(\epsilon) = \int_{||\xi - x_i|| < \epsilon/2} f(\xi) d\xi$.

The expectation of of $\log p_i(\epsilon)$ is $E(\log p_i) = \int_0^\infty P_k(\epsilon) \log p_i(\epsilon) d\epsilon$ that is $= k\binom{N-1}{k} \int_0^1 p^{k-1}(1-p)^{N-k-1} \log p \cdot dp = \psi(k) - \psi(N)$, where $\psi(\cdot)$ is the well-known digamma function. If assumed that $f(x)$ is constant in the entire ϵ-ball, then the approximation $p_i(\epsilon) \approx \frac{V_d}{2^d}\epsilon^d \mu(x_i)$ can be formulated. Here d is the dimension and V_d is the volume of the unit ball $\mathcal{B}(0,1)$, defined as $V_d = \frac{\pi^{\frac{d}{2}}}{\Gamma(\frac{d}{2}+1)}$. From the previous approximation and using the expectation of $\log p_i(\epsilon)$, we have the approximation $\log f(\epsilon) \approx \psi(k) - \psi(N) - dE(\log \epsilon) - \log \frac{V_d}{2^d}$, and finally,

$$\hat{H}(X) = -\psi(k) + \psi(N) + \log \frac{V_d}{2^d} + \frac{d}{N} \sum_{i=1}^{N} \log \epsilon_i \tag{14}$$

is the estimation of $H(X)$, where $\epsilon_i = 2||x_i - x_j||$ is twice the distance between the sample x_i and its k-NN x_j. It is suggested that the error for Gaussian and uniform distributions is $\sim k/N$ or $\sim k/N \log(N/k)$.

4 Experiments

We present serveral experiments in order to show the performance of our method. We have tested the algorithm on both synthetic and real data.

4.1 Simple Densities

In this first experiment we have generated $2,500$ samples from five bidimensional Gaussians with different prior probabilities, averages and covariance matrices. However the distributions do not overlap and are well separated. Fig. 1 shows the estimations of the mixtures and the final Bayesian classification of the samples. The parameters of the mixture of this experiment are:

$$\Sigma_1 = \begin{bmatrix} 0.20\ 0.00 \\ 0.00\ 0.30 \end{bmatrix}, \Sigma_2 = \begin{bmatrix} 0.60\ 0.15 \\ 0.15\ 0.60 \end{bmatrix},$$

$$\Sigma_3 = \begin{bmatrix} 0.40\ 0.00 \\ 0.00\ 0.25 \end{bmatrix}, \Sigma_4 = \begin{bmatrix} 0.60\ 0.00 \\ 0.00\ 0.30 \end{bmatrix},$$

$$\Sigma_5 = \begin{bmatrix} 0.20\ 0.00 \\ 0.00\ 0.30 \end{bmatrix},$$

$$\begin{gathered} \pi_k = 0.2, \\ \mu_1 = [-1, -1]^T, \mu_2 = [6, 3]^T, \mu_3 = [3, 6]^T, \\ \mu_4 = [2, 2]^T, \mu_5 = [0, 0]^T. \end{gathered} \tag{15}$$

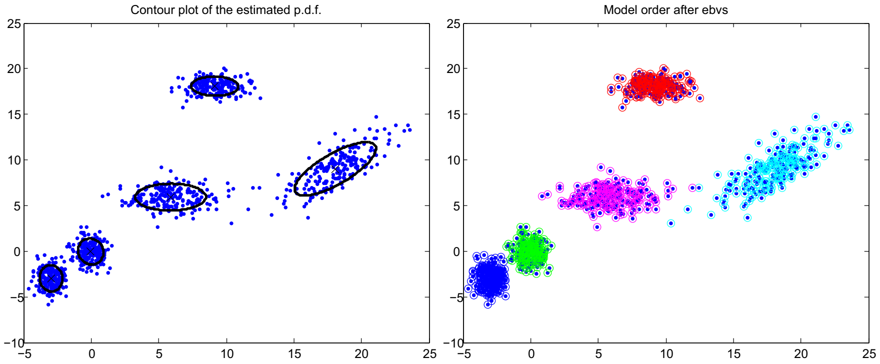

Fig. 1. First experiment: Easy density estimation

4.2 Overlapping Densities

This experiment presents the problem of having overlaping densities. The method show a sucessful density estimation and classification. We generated $1,000$ samples from four

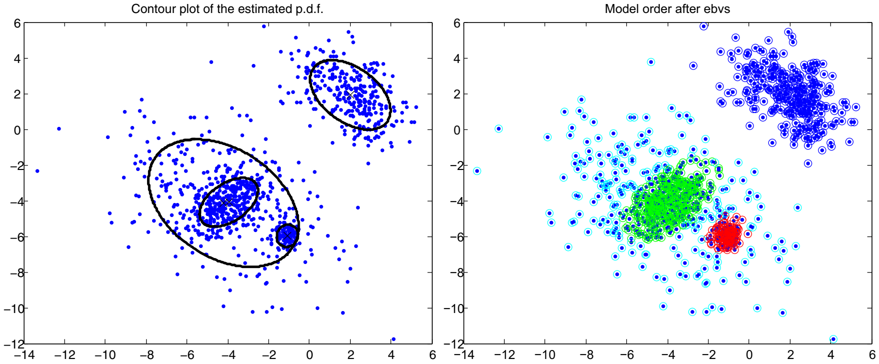

Fig. 2. Overlaped components experiment

bidimensional Gaussians with different prior probabilities, averages and covariance matrices. Fig. 2 shows the estimations of the mixtures and the final Bayesian classification of the samples. The mixture parameteres are the following:

$$\Sigma_1 = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}, \Sigma_2 = \begin{bmatrix} 6 & -2 \\ -2 & 6 \end{bmatrix},$$

$$\Sigma_3 = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}, \Sigma_4 = \begin{bmatrix} 0.125 & 0 \\ 0 & 0.125 \end{bmatrix},$$

$$\begin{gathered} \pi_1 = \pi_2 = \pi_3 = 0.3, \\ \pi_4 = 0.1, \\ \mu_1 = \mu_2 = [-4, -4]^T, \mu_3 = [2, 2]^T, \mu_4 = [-1, -6]^T. \end{gathered} \tag{16}$$

This experiment is used in [16] to argument that their proposed incremental improves the results obtained in [8] (in both cases a MDL criterion is used for model selection). Then our method produces also a good result in this case.

4.3 Symmetric Densities

This experiment presents the problem of having symmetric densities. One Gaussian is placed in the center and four symmetric Gaussians, with symmetric covariances, are placed around the center. We generated $1,000$ samples from this mixture. Fig. 3 shows the successful estimations of the mixtures and the final Bayesian classification of the samples. The mixture parameteres are the following:

$$\Sigma_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \Sigma_2 = \Sigma_4 = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}, \Sigma_3 = \Sigma_5 = \begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix},$$

$$\begin{gathered} \pi_k = 0.2, \\ \mu_1 = [0, 0]^T, \mu_2 = [3, -3]^T, \mu_3 = [3, 3]^T, \mu_4 = [-3, 3]^T, \mu_5 = [-3, -3]^T. \end{gathered} \tag{17}$$

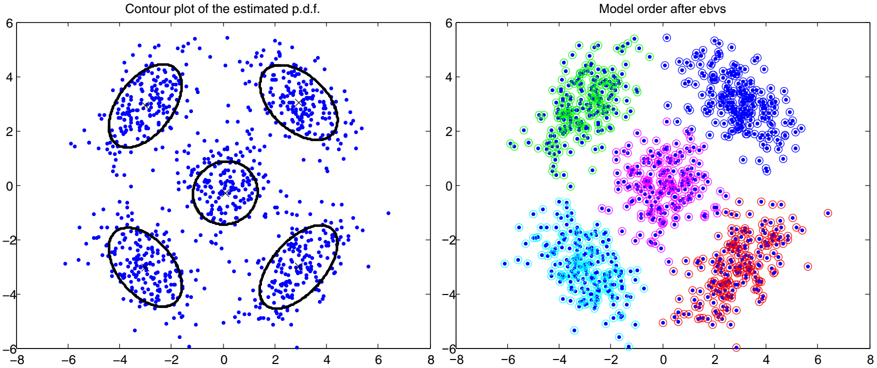

Fig. 3. Experiment with symmetric densities

4.4 Real Data

We have experimented with real data in the context of unsupervised pattern classification. The data set we tested has a relatively high number of dimensions. The well-known *Wine* data set contains three classes of 178 samples of 13 dimensions. This data set comes from chemical analysis of wines grown in different cultivars from the same region, in Italy. The dimensions correspond to 13 constituents found in each one of the three types (classes) of wines. The data set is preprocessed in order to have zero mean and unit variance in each one of the dimensions. The classification performance we obtain on this data set is 86%.

Altough experiments in higher dimensions can be performed, when the number of samples is not high enough, the risk of unbounded maxima of the likelihood function is higher, due to singular covariance matrices. The entropy estimation method, however, performs very well with thousands of dimensions.

5 Conclusions and Future Work

In this paper we have proposed a significant improvement, in terms of computational complexity, of the VBgmmSplit method. Such an improvement relies on the reduction of the quadratic number of splits per iteration to a linear one and this is key for reducing the complexity of the VB optimization method. Splits are reduced by selecting only the worse (lowest entropy) kernel to split and entropy estimation is addressed through a recently proposed bypass method. Our future work includes the extension to other kind of mixtures as well as the incorporation of a more robust/reliable split strategy.

References

1. Jain, A., Dubes, R., Mao, J.: Statistical pattern recognition: A review. IEEE Trans. Pattern Anal. Mach. Intell. 22(1), 4–38 (2000)
2. Titterington, D., Smith, A., Makov, U.: Statistical Analysis of Finite Mixture Distributions. John Wiley and Sons, Chichester (2002)

3. Jain, A., Dubes, R.: Algorithms for Clustering Data. Prentice Hall, Englewood Cliffs (1988)
4. Hastie, T., Tibshirani, R.: Discriminant analysis by gaussian mixtures. Journal of The Royal Statistical Society(B) 58(1), 155–176 (1996)
5. Hinton, G., Dayan, P., Revow, M.: Modeling the manifolds of images of handwriting digits. IEEE Transactions On Neural Networks 8(1), 65–74 (1997)
6. Dalal, S., Hall, W.: Approximating priors by mixtures of natural conjugate priors. Journal of The Royal Statistical Society(B) 45(1) (1983)
7. Box, G., Tiao, G.: Bayesian Inference in Statistical Models. Addison-Wesley, Reading (1992)
8. Figueiredo, M., Jain, A.: Unsupervised learning of finite mixture models. IEEE Trans. Pattern Anal. Mach. Intell. 24(3), 381–399 (2002)
9. Husmeier, D.: The bayesian evidence scheme for regularizing probability-density estimating neural networks. Neural Computation 12(11), 2685–2717 (2000)
10. MacKay, D.: Introduction to Monte Carlo Methods. In: Jordan, M.I. (ed.) Learning in Graphical Models. MIT Press, MA (1999)
11. Ghahramani, Z., Beal, M.: Variational inference for bayesian mixture of factor analysers. In: Adv. Neur. Inf. Proc. Sys., MIT Press, Cambridge (1999)
12. Nasios, N., Bors, A.: Variational learning for gaussian mixtures. IEEE Trans. on Systems, Man, and Cybernetics - Part B: Cybernetics 36(4), 849–862 (2006)
13. Nasios, N., Bors, A.: Blind source separation using variational expectation-maximization algorithm. In: Petkov, N., Westenberg, M.A. (eds.) CAIP 2003. LNCS, vol. 2756, pp. 442–450. Springer, Heidelberg (2003)
14. Figueiredo, M., Leitao, J., Jain, A.: On fitting mixture models. In: Hancock, E.R., Pelillo, M. (eds.) EMMCVPR 1999. LNCS, vol. 1654, pp. 54–69. Springer, Heidelberg (1999)
15. Figueiredo, M.A.T., Jain, A.K.: Unsupervised selection and estimation of finite mixture models. In: Proc. Int. Conf. Pattern Recognition, pp. 87–90. IEEE, Los Alamitos (2000)
16. Penalver, A., Escolano, F., Sáez, J.: Learning gaussian mixture models with entropy-based criteria. IEEE Transactions on Neural Networks 20(11), 1756–1772 (2009)
17. Constantinopoulos, C., Likas, A.: Unsupervised learning of gaussian mixtures based on variational component splitting. IEEE Transactions on Neural Networks 18(3), 745–755 (2007)
18. Watanabe, K., Akaho, S., Omachi, S.: Variational bayesian mixture model on a subspace of exponential family distributions. IEEE Transactions on Neural Networks 20(11), 1783–1796 (2009)
19. Attias, H.: Inferring parameters and structure of latent variable models by variational bayes. In: Proc. of Uncertainty Artif. Intell., pp. 21–30 (1999)
20. Corduneau, A., Bishop, C.: Variational bayesian model selection for mixture distributions. In: Artificial Intelligence and Statistics, pp. 27–34. Morgan Kaufmann, San Francisco (2001)
21. Richardson, S., Green, P.: On bayesian analysis of mixtures with unknown number of components (with discussion). Journal of the Royal Statistical Society B 59(1), 731–792 (1997)
22. Hero, A., Michel, O.: Estimation of rényi information divergence via pruned minimal spanning trees. In: Workshop on Higher Order Statistics, Caessaria, Israel. IEEE, Los Alamitos (1999)
23. Leonenko, N., Pronzato, L.: A class of rényi information estimators for multi-dimensional densities. The Annals of Statistics 36(5), 2153–2182 (2008)