A GREEDY-EM ALGORITHM FOR GAUSSIAN MIXTURE LEARNING

Nikos Vlassis, Aristidis Likas

20-2000

Preprint no. 20-00/2000

Department of Computer Science
University of Ioannina
451 10 Ioannina, Greece

A greedy-EM algorithm for Gaussian mixture learning

Nikos Vlassis
RWCP Auton. Learn. Funct. SNN
University of Amsterdam
Kruislaan 403, 1098 SJ Amsterdam
The Netherlands
vlassis@science.uva.nl

Aristidis Likas
Department of Computer Science
University of Ioannina
45110 Ioannina
Greece
arly@cs.uoi.gr

Abstract

Learning a Gaussian mixture with a local algorithm like EM can be difficult because (i) the true number of mixing components is usually unknown, (ii) there is no generally accepted method for parameter initialization, and (iii) the algorithm can get stuck in one of the many local maxima of the likelihood function. In this paper we propose a greedy algorithm for learning a Gaussian mixture which tries to overcome these limitations. In particular, starting with a single component and adding components sequentially until a maximum number k, the algorithm is capable of achieving solutions superior to EM with k components in terms of the likelihood of a test set. The algorithm is based on recent theoretical results on iterative mixture density estimation, and uses a combination of global and local search each time a new component is added to the mixture.

1 Introduction

Finite mixture distributions [1] provide a simple framework for modeling population heterogeneity. If $\phi(\mathbf{x};\boldsymbol{\theta}_j)$ is the j-th component model parametrized on $\boldsymbol{\theta}_j$, then a mixture density for a random vector $\mathbf{x}$ assuming k components is

$$f_k(\mathbf{x}) = \sum_{j=1}^{k} \pi_j \phi(\mathbf{x};\boldsymbol{\theta}_j) \qquad (1)$$

where π_j are the mixing weights satisfying $\pi_1 + \ldots + \pi_k = 1$, $\pi_j \geq 0$. Mixtures have proven useful tools for data analysis and recent examples are mixtures of factor analyzers [2] and principal component analyzers [3].

Learning the mixture, namely, estimating the weights π_j and the parameters $\boldsymbol{\theta}_j$ of each component, is often carried out through likelihood maximization using the Expectation-Maximization (EM) algorithm [4]. The popularity of EM is due to its simple implementation together with the guaranteed monotone increase of the likelihood of the training set during optimization. However, known limitations of EM are (i) it assumes that the number k of mixing components is known, (ii) there is no widely accepted 'good' method for initializing the parameters, and (iii) the algorithm is of local nature and thus can get stuck in local maxima of the likelihood function.

Theoretical evidence [5, 6] indicates that it is possible to learn a mixture density by maximum likelihood in a greedy fashion, namely, by iteratively adding components to the mixture up to a desired number of components k. As shown in [6], if component insertion is carried out in an optimal way, such an iteratively computed mixture is almost as good as any k-component mixture in the form (1).

The practical consequence of this result is that learning a k-component mixture can be replaced by an iterative two-component mixture learning which is a substantially simpler task, making the learning algorithm less sensitive to the above limitations of EM. Nevertheless, in the work of [6] the important issue of optimal allocation of a new component is not adequately addressed, and this constitutes an open problem.

To this end, we propose in this paper a greedy algorithm for learning a general multivariate Gaussian mixture. We start with one component. Assuming at some point of the algorithm k components, regular EM steps are carried out until convergence, and then a new component is added to the mixture according to [6]. To locate the optimal position of the new component we propose the use of (i) a global search among all input points, followed by (ii) a local search based on partial EM steps for fine tuning of the parameters of the new component. Simulation results indicate that the proposed algorithm (running until k components have been added) seems to outperform EM (with k components) in terms of the likelihood of a test set.

2 Gaussian mixtures and the EM algorithm

A multivariate Gaussian mixture is given by the weighted sum (1), where the j-th component $\phi(\mathbf{x};\boldsymbol{\theta}_j)$ is the d-dimensional Gaussian density

$$\phi(\mathbf{x};\boldsymbol{\theta}_j) = (2\pi)^{-d/2}|\mathbf{S}_j|^{-1/2}\exp\left[-0.5(\mathbf{x}-\mathbf{m}_j)^T\mathbf{S}_j^{-1}(\mathbf{x}-\mathbf{m}_j)\right] \tag{2}$$

parametrized on the mean $\mathbf{m}_j$ and the covariance matrix $\mathbf{S}_j$, collectively denoted by the parameter vector $\boldsymbol{\theta}_j$. For the learning problem we assume a training set $\{\mathbf{x}_1,\dots,\mathbf{x}_n\}$ of independent and identically distributed points sampled from the mixture, and the task is to estimate the parameters $\{\pi_j,\mathbf{m}_j,\mathbf{S}_j\}$ of the k components that maximize the log-likelihood

$$\mathcal{L}_k = \sum_{i=1}^{n}\log f_k(\mathbf{x}_i). \tag{3}$$

In order to prevent maxima of the log-likelihood going to infinity, we must place appropriate constraints on the singular values of the covariance matrices of the mixing components. In particular, the eccentricity of each component, i.e., the ratio of the largest to the smallest singular value of its covariance matrix, must be upper bounded. Then, log-likelihood maximization can be carried out by the EM algorithm using the following iterative update equations for each component j, $j=1,\dots,k$ [7]

$$P(j|\mathbf{x}_i) = \frac{\pi_j\phi(\mathbf{x}_i;\boldsymbol{\theta}_j)}{f_k(\mathbf{x}_i)}, \tag{4}$$

$$\pi_j := \frac{1}{n}\sum_{i=1}^{n}P(j|\mathbf{x}_i), \tag{5}$$

$$\mathbf{m}_j := \frac{\sum_{i=1}^{n}P(j|\mathbf{x}_i)\mathbf{x}_i}{\sum_{i=1}^{n}P(j|\mathbf{x}_i)}, \tag{6}$$

$$\mathbf{S}_j := \frac{\sum_{i=1}^{n}P(j|\mathbf{x}_i)(\mathbf{x}_i-\mathbf{m}_j)(\mathbf{x}_i-\mathbf{m}_j)^T}{\sum_{i=1}^{n}P(j|\mathbf{x}_i)}. \tag{7}$$

It can be shown that in each EM step the log-likelihood cannot decrease [8]. Further details about the EM algorithm can be found, e.g., in [9].

3 Greedy mixture learning

The proposed algorithm is based on theoretical evidence [6] that, under assumptions, maximum likelihood learning of a mixture can be done in a greedy manner by iteratively adding components to the mixture. In particular, assume that a new component $\phi(\mathbf{x};\boldsymbol{\theta})$ is added to a k-component mixture $f_k(\mathbf{x})$ giving the mixture

$$f_{k+1}(\mathbf{x}) = (1-a)f_k(\mathbf{x}) + a\phi(\mathbf{x};\boldsymbol{\theta}), \tag{8}$$

with a in $(0,1)$. Then, as long as for each k, given $f_k(\mathbf{x})$, the weight a and the parameter vector $\boldsymbol{\theta}$ of $\phi(\mathbf{x};\boldsymbol{\theta})$ are optimally chosen so that the new log-likelihood

$$\mathcal{L}_{k+1} = \sum_{i=1}^{n} \log f_{k+1}(\mathbf{x}_i) = \sum_{i=1}^{n} \log[(1-a)f_k(\mathbf{x}_i) + a\phi(\mathbf{x}_i;\boldsymbol{\theta})] \tag{9}$$

is maximized, then the resulting mixture has almost at least as high a likelihood as is achieved by any mixture density with the same number of components. A notable property of the above two-component maximization problem is that the parameters of $f_k(\mathbf{x})$ remain fixed during maximization of $\mathcal{L}_{k+1}$.

The importance of this result is that maximum likelihood learning of a general Gaussian mixture can be replaced by the iterative learning of the two-component mixtures $f_{k+1}(\mathbf{x})$, where the first component is the old mixture $f_k(\mathbf{x})$ and the second one is the Gaussian component $\phi(\mathbf{x};\boldsymbol{\theta})$ with $\boldsymbol{\theta} = [\mathbf{m}, \mathbf{S}]$ its mean and covariance matrix. This is advantageous from a practical point of view since a two-component mixture is easier to learn than a general mixture, however, appropriate search techniques need to be developed in order to optimally specify the parameters a, $\mathbf{m}$, and $\mathbf{S}$ that maximize $\mathcal{L}_{k+1}$. An effective technique to deal with this problem is presented next.

3.1 Local search

Since we have to learn a two-component mixture, an EM algorithm can be employed to perform local search for the maxima of $\mathcal{L}_{k+1}$ with respect to a, $\mathbf{m}$, and $\mathbf{S}$. Moreover, since the parameters of $f_k(\mathbf{x})$ remain fixed during component allocation, partial EM steps can be used in which (4)–(7) update only the mixing weight a, the mean $\mathbf{m}$, and the covariance matrix $\mathbf{S}$ of the newly inserted component, i.e.,

$$P(k+1|\mathbf{x}_i) = \frac{a\phi(\mathbf{x}_i;\mathbf{m},\mathbf{S})}{(1-a)f_k(\mathbf{x}_i) + a\phi(\mathbf{x}_i;\mathbf{m},\mathbf{S})}, \tag{10}$$

$$a := \frac{1}{n}\sum_{i=1}^{n} P(k+1|\mathbf{x}_i), \tag{11}$$

$$\mathbf{m} := \frac{\sum_{i=1}^{n} P(k+1|\mathbf{x}_i)\mathbf{x}_i}{\sum_{i=1}^{n} P(k+1|\mathbf{x}_i)}, \tag{12}$$

$$\mathbf{S} := \frac{\sum_{i=1}^{n} P(k+1|\mathbf{x}_i)(\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^T}{\sum_{i=1}^{n} P(k+1|\mathbf{x}_i)}. \tag{13}$$

Since only the parameters of the new component are updated, partial EM steps constitute a simple and fast method for locally searching for the maxima of $\mathcal{L}_{k+1}$, without needing to resort to other computationally demanding nonlinear optimization methods. However, the above EM-based scheme is still a local algorithm which is very sensitive to the initial values of the parameters a, $\mathbf{m}$, and $\mathbf{S}$. In the following we propose a global search strategy for proper parameter initialization.

3.2 Global search

The partial EM steps require the initialization of the parameters $\mathbf{m}$ and $\mathbf{S}$ of the new component, and also the weight a. In order to facilitate the global search over the parameter space, we can substitute the log-likelihood function (9) with a Taylor approximation about a point $a = a_o$, and then use the resulting estimate to search for the optimal $\mathbf{m}$ and $\mathbf{S}$. More specifically, we expand $\mathcal{L}_{k+1}$ by second order Taylor about $a_o = 0.5$ and then maximize the resulting quadratic function with respect to a. It is not difficult to see that this procedure gives the approximation

$$\hat{\mathcal{L}}_{k+1} = \mathcal{L}_{k+1}(a_o) - \frac{[\mathcal{L}'_{k+1}(a_o)]^2}{2\mathcal{L}''_{k+1}(a_o)} \tag{14}$$

with $\mathcal{L}'_{k+1}$ and $\mathcal{L}''_{k+1}$ the first and second derivatives of $\mathcal{L}_{k+1}$ with respect to a. If we define

$$\delta(\mathbf{x},\boldsymbol{\theta}) = \frac{f_k(\mathbf{x}) - \phi(\mathbf{x};\boldsymbol{\theta})}{f_k(\mathbf{x}) + \phi(\mathbf{x};\boldsymbol{\theta})} \tag{15}$$

then a local maximum of $\mathcal{L}_{k+1}$ near $a_o = 0.5$ is easily shown to be

$$\hat{\mathcal{L}}_{k+1} = \sum_{i=1}^{n} \log \frac{f_k(\mathbf{x}_i) + \phi(\mathbf{x}_i;\boldsymbol{\theta})}{2} + \frac{1}{2}\frac{\left[\sum_{i=1}^{n}\delta(\mathbf{x}_i,\boldsymbol{\theta})\right]^2}{\sum_{i=1}^{n}\delta^2(\mathbf{x}_i,\boldsymbol{\theta})} \tag{16}$$

and is obtained for a equal to

$$\hat{a} = \frac{1}{2} - \frac{1}{2}\frac{\sum_{i=1}^{n} \delta(\mathbf{x}_i, \boldsymbol{\theta})}{\sum_{i=1}^{n} \delta^2(\mathbf{x}_i, \boldsymbol{\theta})}. \tag{17}$$

If this value falls outside the range $(0, 1)$ then we can initialize the partial EM with the approximation $\hat{a} = 0.5$ for $k = 1$ and $\hat{a} = 2/(k+1)$ for $k \geq 2$, according to [6].

The above procedure makes the likelihood function in (9) independent of the mixing weight a, and the next step is to find a good initialization of the center $\mathbf{m}$ and covariance matrix $\mathbf{S}$ of the new component. In general, this involves a global search over the space of all $[\mathbf{m}, \mathbf{S}]$ parameters which is clearly infeasible since $\mathbf{S}$ is a general covariance matrix involving many parameters.

However, we observe from (16) that $\hat{\mathcal{L}}_{k+1}$ depends only on $\phi(\mathbf{x}_i; \mathbf{m}, \mathbf{S})$ (remember that $f_k(\mathbf{x})$ remains fixed during optimization) which, for constant covariance matrix $\mathbf{S} = \sigma^2\mathbf{I}$, is just a function of the Euclidean distance of $\mathbf{m}$ to the input point $\mathbf{x}_i$. Thus, if we restrict our global search for the new $\mathbf{m}$ only over the input points $\mathbf{x}_j$, then evaluation of $\hat{\mathcal{L}}_{k+1}$ for all $\mathbf{m}$ requires the computation of all pairwise Euclidean distances $||\mathbf{x}_i - \mathbf{x}_j||$ between inputs which can be carried out only once at the beginning of the algorithm.

Thus, in the initialization of the learning algorithm we also store a matrix $\mathbf{K}$ with elements

$$k_{ij} = (2\pi\sigma^2)^{-d/2} \exp(-0.5||\mathbf{x}_i - \mathbf{x}_j||^2/\sigma^2) \tag{18}$$

for an appropriate σ, and then use the k_{ij} in each component allocation step in the rest of the algorithm for computing the required $\phi(\mathbf{x}_i; \mathbf{x}_j, \mathbf{S})$ values in (16). We can also define many kernel matrices, one for each value of σ, and then compute the maximum of $\hat{\mathcal{L}}_{k+1}$ over all σ. A similar approach that uses a matrix $\mathbf{K}$ for searching for global solutions over the parameter space has been proposed in [10].

The choice of σ must depend on the dimension d of the data and the size n of the training set, and we propose a σ proportional to the value that minimizes the mean integrated squared error of a nonparametric multivariate density estimator [11]

$$\sigma = \beta \left[\frac{4}{(d+2)n}\right]^{1/(d+4)} \tag{19}$$

with β a fixed number.

Using the above approach the time complexity of each evaluation of $\hat{\mathcal{L}}_{k+1}$ is $O(n)$, giving total complexity for the global search $O(n^2)$. The use of the kernel matrix $\mathbf{K}$ makes this computation feasible during optimization.

3.3 The greedy-EM algorithm

Summarizing the above ideas we have the following 'greedy-EM' algorithm for learning a multivariate Gaussian mixture.

1. Initialize using one component with $\mathbf{m} = E[\mathbf{x}]$ and $\mathbf{S}$ =Cov($\mathbf{x}$). Compute σ in (19) by setting β to half the value of the maximum singular value of the covariance matrix of $\mathbf{x}_i$. Compute the kernel matrix from (18).

2. Perform EM steps until convergence: $\mathcal{L}_k^t/\mathcal{L}_k^{t-1} - 1 <$ 1e-6. If an appropriate stopping condition holds then terminate.

3. Search over all $\mathbf{x}_j$ for candidate locations for the new component. Set $\mathbf{m}$ to the $\mathbf{x}_j$ that maximizes (16) using the precomputed kernel values k_{ij} in place of $\phi(\mathbf{x}_i; \mathbf{x}_j, \sigma^2\mathbf{I})$.

4. Initialize the partial EM with the estimated value of $\mathbf{m}$, with $\mathbf{S} = \sigma^2\mathbf{I}$, and $\hat{a}$ from (17).

5. Apply partial EM steps (10)–(13) until convergence as in step 2.

6. If $\mathcal{L}_{k+1} \leq \mathcal{L}_k$ then terminate, otherwise allocate the new component and go to 2.

Since EM cannot lead to decrease of the log-likelihood and the partial EM solutions are accepted only if $\mathcal{L}_{k+1} > \mathcal{L}_k$, the algorithm ensures the monotone increase of the log-likelihood of the training set.

The stopping condition in step 2 is typically the maximum allowed number of components k. If the task is the estimation of the true number of components of the mixture, then we can run the algorithm for a large final value of k and then select the optimal $\hat{k}$ based on some model selection criterion, e.g., cross-validation using a set of test points, a coding scheme based on MDL [6], etc.

4 Experiments

We compared the performance of the proposed greedy-EM algorithm to the regular EM algorithm by carrying out a set of experiments using both synthetic and real data.

In the synthetic data case we created 50 random mixtures of varying complexity (see below), and sampled from each of them a training set of 400 points and a test set of 200 points. We tested the methods for dimensions $d = 2$ and $d = 5$ because higher dimensions would suffer from the limited size of the training set (curse of dimensionality). We applied both the greedy-EM and the EM algorithm to these mixtures and computed the average, over the 50 runs, of the log-likelihood of the test set for each of them. The quality measure we used was the difference between the two averages (the standard deviations were small and comparable so we skipped them).

In all experiments the greedy-EM was initialized with σ and β as described in Section 3.3. For the initialization of the EM algorithm, the mixing weights were uniformly sampled in $[1/(2k), 1]$ and then normalized to sum to one, the component centers were randomly placed over the range of the training points, and the initial covariances were spherical, each with variance according to [12]

$$\sigma_i^2 = \frac{1}{2d} \min_{j \neq i} ||\mathbf{m}_j - \mathbf{m}_i||^2. \tag{20}$$

The covariance matrix of each component was transformed after each EM step to its Cholesky decomposition leading to a speedup of the algorithm, while its eccentricity was upper bounded by 1e4 in order to avoid singular solutions from the EM steps.

The complexity of the mixtures in the 50 experiments was controlled by the number of components k taking the values $k = 4, 6, 8, 10$, the maximum allowed eccentricity $e_{\max}$ (see definition in Section 2) which was kept constant to the value $e_{\max} = 15$, and the degree of separation c of the components of the mixture taking the values $c = 1, 2, 3, 4$. For setting the eccentricity of each component, the singular values of its covariance matrix were sampled uniformly in $[1, e_{\max}]$, while the degree of separation c imposed a constraint on the positions of the component centers according to [12]

$$||\mathbf{m}_i - \mathbf{m}_j|| \geq c\sqrt{\max\{\text{trace}(\mathbf{S}_i), \text{trace}(\mathbf{S}_j)\}}. \tag{21}$$

Each mixture was generated so as to satisfy this bound as tightly as possible.

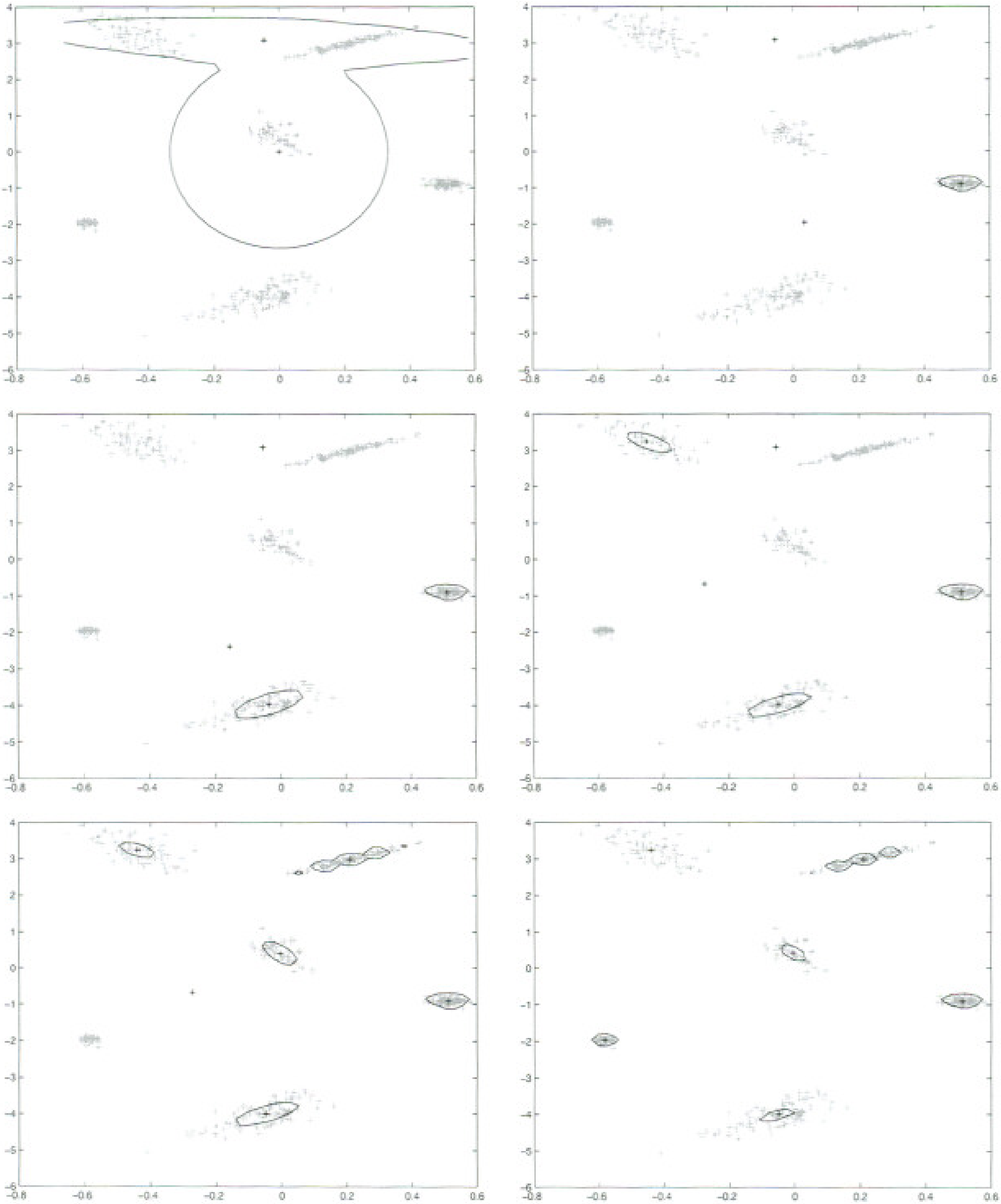

Figure 1: A six-component 4-separated bivariate mixture and the component allocation steps.

$d = 2$	$c = 1$	$c = 2$	$c = 3$	$c = 4$
$k = 4$	0.1006	0.3155	0.4113	0.4944
$k = 6$	0.1182	0.2851	0.5518	0.6657
$k = 8$	0.1060	0.3182	0.4694	0.6733
$k = 10$	0.0666	0.3828	0.5432	0.7726

$d = 5$	$c = 1$	$c = 2$	$c = 3$	$c = 4$
$k = 4$	0.1149	0.4390	0.5439	0.4376
$k = 6$	0.0888	0.2730	0.6869	0.3097
$k = 8$	0.1134	0.2482	0.2202	0.6083
$k = 10$	0.1605	0.3413	0.5267	0.7754

Figure 2: Results of the synthetic experiment for $d = 2$ and $d = 5$ and various values of k (number of components) and c (degree of separation).

A typical six-component mixture with $d = 2$, $c = 4$, and $e_{\max} = 15$, together with the results of the greedy-EM algorithm is shown in Fig. 1. In each diagram we plot the newly allocated component after partial EM has converged, while the contours indicate the value of the mixture density. Note how partial EM and regular EM steps are combined to locate the components centers. The results of the complete synthetic experiment are summarized in Fig. 2. We see that the greedy-EM algorithm achieves solutions superior to EM on the average, and this becomes clearer as the degree of separation c of the components increases.

In the second part of the experiments we applied the greedy-EM and the EM algorithm on a real image segmentation data set available from the UCI repository [13]. This data set contains 210 training patterns and 2100 test patterns of 19 features. Although this is a supervised data set, in the conducted experiments we ignored the class labels of the patterns. To reduce the dimensionality we applied Principal Component Analysis and retained only the first five principal dimensions which explained more than 95% of the total variance of the data. In the reduced five-dimensional data set, we applied the greedy-EM once and the EM algorithm 50 times (with different initializations), for values of k from two to 20.

In Fig. 3 we plot the log-likelihood of the test set as a function of k. The boxes correspond to the value of the greedy-EM, the crosses to the maximum (over the 50 trials) value of the EM, and the circles (with error bars) to the average ($\pm\sigma$) values of the EM. We see that the greedy-EM is constantly superior to the average EM solutions and even reaches or surpasses the best possible EM solutions for some k. The value $k = 13$ was reported by the greedy-EM algorithm as the optimal number of components using the test

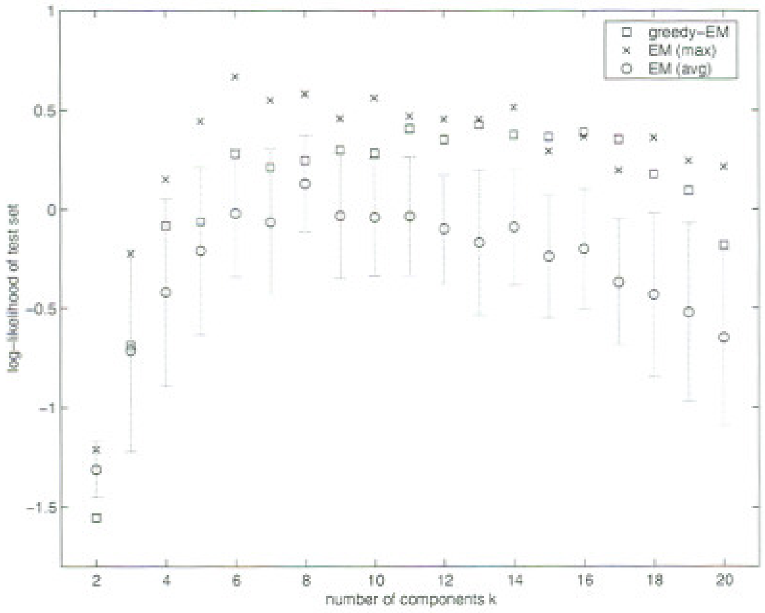

Figure 3: Log-likelihood of the test set for the UCI data after PCA. The greedy-EM algorithm achieves superior solutions to EM on the average.

set. This experiment provides a second indication of the superiority of the greedy-EM algorithm over EM.

A Matlab implementation of the algorithm is available at `http://www.science.uva.nl/~vlassis/software`

5 Discussion

In the previous section we showed experimental results from the application of both the greedy-EM and the EM algorithms to synthetic and real data. One point that should be stressed, however, is that the performance of EM depends very much on the initialization of its parameters, and for this reason the comparison of the greedy-EM to EM can be considered biased. However, we decided to compare with EM since the latter is still state of the art in mixture problems and also because several other algorithms report results compared to those of EM [12, 14].

There are a few published algorithms in the literature that bear similarities to the proposed greedy-EM algorithm and belong to the category of techniques related to the

Vertex Direction Method (VDM) [5, 15, 16]. More specifically, an algorithm that combines the EM algorithm with dynamic component allocation has been proposed in [15] but only for univariate mixtures. Also, like in our approach, a second order Taylor expansion about $a_o = 0$ of a two-component mixture has been proposed in [16]. However, due to the discontinuities of $\mathcal{L}_{k+1}$ near $a_o = 0$, suboptimal solutions could result as we noticed in our implementations. In the latter work it is also not clearly specified how to search for the maxima of $\hat{\mathcal{L}}_{k+1}$, especially in the case of multivariate mixtures.

A Bayesian approach to the problem of learning a mixture with an unknown number of components has been proposed in [17]. There, a reversible jump Markov Chain Monte Carlo method is employed for switching between parameter subspaces of different dimensionality (the number of components of the mixture). The dimension-changing 'jumps' are in the form of splitting, combining, and deleting components. The main argument against the use of this method is its computational complexity (see discussion in [17]).

A related approach which also uses split and merge operations between components has been proposed in [14]. In this approach the number of components remains unchanged, while a form of backtracking is required to ensure that the log-likelihood increases in each split-merge step. The split test statistic employed in this approach is the Kullback divergence between a component density and the empirical density in the vicinity of the component.

In the past we have also proposed an incremental scheme for learning univariate Gaussian mixtures, in which a component of the mixture is split according to a statistical test involving the kurtosis of the component [18]. However, the proposed greedy-EM algorithm seems to be more robust and avoids possible problems of the kurtosis related to outliers.

Ongoing research focuses on two directions. First we are investigating the possibility of applying this technique in learning latent mixture models [2, 3]. A second important issue is the acceleration of the algorithm, especially the EM optimization part and the global search which is currently $O(n^2)$. Some variant of EM [19] could be used for the former, while for the latter we expect that the use of specialized structures for storing high-dimensional data like, e.g., k-d trees, combined with a more sophisticated global search scheme may offer significant speedup, especially for high-dimensional problems with a large number of points.

Acknowledgments

N. Vlassis is supported by the Real World Computing program. A. Likas is partially supported by the Greek General Secretariat for Research and Technology through the R&D program PENED99/318.

References

[1] D. M. Titterington, A. F. Smith, and U. E. Makov, *Statistical Analysis of Finite Mixture Distributions*, Wiley, 1985.

[2] Z. Ghahramani and G.E. Hinton, "The EM algorithm for mixtures of factor analyzers," Tech. Rep., University of Toronto, 1997, CRG-TR-96-1.

[3] M. E. Tipping and C. M. Bishop, "Mixtures of probabilistic principal component analysers," *Neural Computation*, vol. 11, no. 2, pp. 443–482, 1999.

[4] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Statist. Soc. B*, vol. 39, pp. 1–38, 1977.

[5] B. G. Lindsay, "The geometry of mixture likelihoods: a general theory," *Ann. Statist.*, vol. 11, no. 1, pp. 86–94, 1983.

[6] J. Q. Li and A. R. Barron, "Mixture density estimation," in *Proc. NIPS*, 1999.

[7] R. A. Redner and H. F. Walker, "Mixture densities, maximum likelihood and the EM algorithm," *SIAM Review*, vol. 26, no. 2, pp. 195–239, Apr. 1984.

[8] B. D. Ripley, *Pattern Recognition and Neural Networks*, Cambridge University Press, Cambridge, U.K., 1996.

[9] G. J. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*, Wiley, New York, 1997.

[10] A. J. Smola, O. L. Mangasarian, and B. Schölkopf, "Sparse kernel feature analysis," Tech. Rep., Data Mining Institute, University of Wisconsin, Madison, 1999, 99-04.

[11] M. P. Wand, "Fast computation of multivariate kernel estimators," *J. Comp. and Graph. Statistics*, vol. 3, no. 4, pp. 433–445, 1994.

[12] S. Dasgupta, "Learning mixtures of Gaussians," in *Proc. IEEE Symp. on Foundations of Computer Science*, 1999.

[13] C. L. Blake and C. J. Merz, "UCI repository of machine learning databases," University of California, Irvine, Dept. of Information and Computer Sciences, 1998.

[14] N. Ueda, R. Nakano, Z. Ghahramani, and G. E. Hinton, "SMEM algorithm for mixture models," *Neural Computation*, vol. 12, pp. 2109–2128, 2000.

[15] R. DerSimonian, "Maximum likelihood estimation of a mixing distribution," *J. Roy. Statist. Soc. C*, vol. 35, pp. 302–309, 1986.

[16] D. Böhning, "A review of reliable maximum likelihood algorithms for semiparametric mixture models," *J. Statist. Plann. Inference*, vol. 47, pp. 5–28, 1995.

[17] S. Richardson and P. J. Green, "On Bayesian analysis of mixtures with an unknown number of components," *J. Roy. Statist. Soc. B*, vol. 59, no. 4, pp. 731–792, 1997.

[18] N. Vlassis and A. Likas, "A kurtosis-based dynamic approach to Gaussian mixture modeling," *IEEE Trans. on Systems, Man, and Cybernetics, Part A*, vol. 29, no. 4, pp. 393–399, July 1999.

[19] R. M. Neal and G. E. Hinton, "A view of the EM algorithm that justifies incremental, sparse, and other variants," in *Learning in graphical models*, M. I. Jordan, Ed., pp. 355–368. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998.