Unsupervised Learning of Gaussian Mixture Models in the Presence of Dynamic Environments

Conference Paper *in* Lecture Notes in Electrical Engineering · July 2014

DOI: 10.1007/978-3-319-10380-8_37

2 authors:

Abdolrahman Khoshrou
Centrum Wiskunde & Informatica
27 PUBLICATIONS **106** CITATIONS

SEE PROFILE

A. Pedro Aguiar
University of Porto
373 PUBLICATIONS **7,725** CITATIONS

SEE PROFILE

Unsupervised Learning of Gaussian Mixture Models in the Presence of Dynamic Environments

A multiple-model adaptive algorithm *

Abdolrahman Khoshrou and A. Pedro Aguiar

Faculty of Engineering, University of Porto, Portugal
{a.khoshrou,pedro.aguiar}@fe.up.pt

Abstract. This paper tackles the on-line unsupervised learning problem of Gaussian mixture models in the presence of uncertain dynamic environments. In particular, we assume that the number of Gaussian components (clusters) is unknown and can change over time. We propose a multi-hypothesis adaptive algorithm that continuously updates the number of components and estimates the model parameters as the measurements (sample data) are being acquired. This is done by incrementally maximizing the likelihood probability associated to the estimated parameters and keeping/creating/removing in parallel a number of hypothesis models that are ranked according to the *minimum description length* (MDL), a well-known concept in information theory. The proposed algorithm has the additional feature that it relaxes "the sufficiently large data set" restriction by not requiring in fact any initial batch of data. Simulation results illustrate the performance of the proposed algorithm.

Keywords: On-line learning, Adaptation and learning, Gaussian mixture models

1 Introduction

Over the years, research on identifying and classifying unknown number of components in a dynamic environment has been an important topic in computer vision and pattern recognition communities. In particular, for data clustering, mixture models, where each component density of the mixture represents a given set of individuals/samples in the total population, has been applied in a widespread of applications. Mixture models are able to represent arbitrarily complex probability density functions (pdfs). This makes them an excellent choice for representing complex class-conditional pdfs (e.g. likelihood functions) in Bayesian supervised learning scenarios or prior probabilities for Bayesian parameter estimation [1]. For off-line clustering, and more precisely to compute

* This work was partially supported by project CONAV/FCT-PT [PTDC/EEACRO/113820/2009].

the parameters that define the mixture model given a finite data set, a widely used procedure is to apply the *expectation-maximization* (EM) algorithm that incrementally converges to a maximum likelihood estimate of the mixture model. Notice however that the basic EM algorithm is not able to deal with on-line data since it is an iterative algorithm that requires all the batch of data in each iteration. Another important restriction is the number of components of the mixture which is to be fixed (does not change) and has to be known a-priori.
To solve some of the above problems, several approaches have been developed. In [2], starting from a fixed number of clusters in a batch of data, a split-and-merge approach together a dissimilarity index concept is presented that adaptively updates the number of mixture models. Z. Zivkovic et al. [3] inspired by [4] proposed an on-line (recursive) algorithm that estimates the parameters of the mixture and simultaneously selects the number of components by starting with a high number of components in a small batch and searching for the *maximum a posteriori* (MAP) solution, and discarding the irrelevant components. A. Declercq and J. H. Piater et al. [5] presents a method to incrementally learning Gaussian mixture models (GMMs) based on a new fidelity criterion for splitting and merging mixture components.

In this paper we address the on-line unsupervised learning problem of GMMs in the presence of uncertain dynamic environments, i.e., we assume that the number of Gaussian components (clusters) is not only unknown but it also can change over time. Inspired by the work in [4], namely the use of the *minimum description length* (MDL) concept, we propose a multi-hypothesis adaptive algorithm that continuously updates the number of components and estimates the model parameters as the measurements (sample data) are being acquired. The proposed algorithm has the additional feature that it relaxes "the sufficiently large data set" restriction by not requiring in fact any initial batch of data. Simulation results illustrate the performance of the proposed algorithm where it shows that indeed it is able to continuously adapt to the dynamic changes of the number of clusters and estimate the parameters of the mixture model.

2 Problem Statement

Let $\{\mathbf{Y}_n,\ n = 0, 1, 2, \ldots\}$ be a discrete-time random process where for each particular time n, $\mathbf{Y}_n$ follows a K-component mixture of d-dimensional Gaussian with probability density function (pdf) given by

$$p(\mathbf{y}|\theta, w) := \sum_{k=1}^{K} w^{[k]}\, p^{[k]}(\mathbf{y}|\theta^{[k]}), \tag{1}$$

where $\mathbf{y}$ represents one particular outcome of $\mathbf{Y}_n$ and $w := \{w^{[1]}, \ldots, w^{[K]}\}$ is the mixing weight set that satisfies

$$\sum_{k=1}^{K} w^{[k]} = 1, \quad w^{[k]} > 0, \tag{2}$$

K denotes the number of components of the mixture, $\theta^{[k]} := \{\mu^{[k]},\ \Sigma^{[k]}\}$ is the mean and covariance matrix of the k^{th} component, with $\theta := \{\theta^{[1]}, \ldots, \theta^{[K]}\}$, and

$$p^{[k]}(\mathbf{y}|\theta^{[k]}) := \frac{1}{(2\pi)^{d/2}|\Sigma^{[k]}|^{1/2}} \exp\Big(-\frac{1}{2}(\mathbf{y}-\mu^{[k]})^T(\Sigma^{[k]})^{-1}(\mathbf{y}-\mu^{[k]})\Big). \quad (3)$$

Note that for simplicity of notation we have omitted in the parameters K, w, θ their explicit dependence on the time n.

We can now formulate the problem addressed in the paper: *Given a sequence of observations $Y_0, Y_1, \ldots$, find on-line, as the samples are arriving, a sequence of estimates for the parameters K, w, θ that is most likely to be in some sense close to the correct characterization of the random process $\{\mathbf{Y}_n,\ n = 0, 1, 2, \ldots\}$.*

3 Preliminaries and basic results

This section presents several background results, starting with the EM algorithm, that are needed to understand the proposed on-line unsupervised learning algorithm.

3.1 The Basic Expectation-Maximization (Off-line) Algorithm

For finite mixture models, given a set of n independent and identically distributed samples $Y = \{Y_1, \ldots, Y_n\}$, the log-likelihood corresponding to a K-component mixture where all the components are d-dimensional Gaussian is [1]

$$\ell := \log p(Y|\theta, w) = \log \prod_{i=1}^{n} p(Y_i|\theta, w) = \sum_{i=1}^{n} \log \sum_{k=1}^{K} w^{[k]} p(Y_i|\theta^{[k]}) \quad (4)$$

It is well-known that the *maximum likelihood* (ML) or *maximum a posteriori* (MAP) estimates can not be found analytically [1, Ch. 9]. An elegant and powerful method for finding ML or MAP solutions for models with latent variables is called the *expectation-maximization* or EM algorithm [6], [1, Ch. 9]. The EM is an easily implementable algorithm that iteratively increases the posterior density or likelihood function. In order to describe the EM, we need to introduce for each observation Y_i, a discrete unobserved indicator vector $Z_i = [Z_i^{[1]}, \ldots, Z_i^{[K]}]$. This vector specifies from which component the observation Y_i was drawn, i.e., if $Z_i^{[k]} = 1$ and $Z_i^{[p]} = 0$ for $k \neq p$, then this means that the sample Y_i was produced by the k^{th} component. Hence, the complete log-likelihood function (i.e. the one from which we could estimate θ, w if the *complete* data $X = \{Y, Z\}$ was observed [7]) can be written as a product

$$\log p(Y, Z|\theta, w) := \sum_{i=1}^{n} \sum_{k=1}^{K} Z_i^{[k]} \log \Big[w^{[k]} p(Y_i|\theta^{[k]})\Big] \quad (5)$$

The EM algorithm runs over the whole data set Y and until some convergence criterion is met, iteratively produces a sequence of estimates $\hat{\theta}_m, \hat{w}_m, m = 0, 1, 2, \ldots$ by alternatively applying two steps:

E-Step: Given Y and the current estimates $\hat{\theta}_m, \hat{w}_m$ and by considering the fact that $\log p(Y, Z|\theta, w)$ is linear with respect to the missing Z, the so-called Q-function computes the conditional expectation of the complete log-likelihood function as

$$Q(\theta, \hat{\theta}_m) := E\Big[\log p(Y, Z|\theta, w)\Big|Y, \hat{\theta}_m, \hat{w_m}\Big] = \log p(Y, \Gamma|\theta, w), \tag{6}$$

where $\Gamma \equiv E[Z|Y, \hat{\theta}_m, \hat{w}_m]$ is the a conditional expectation that each observation is generated by which component. Since the elements of Z are binary, as mentioned in [4], their conditional expectations are given by

$$\Gamma_i^{[k]} := E\left[Z_i^{[k]}|Y, \hat{\theta}_m, \hat{w}_m\right] = \Pr\left[Z_i^{[k]} = 1|Y_i, \hat{\theta}_m, \hat{w}_m\right] = \frac{\hat{w}_m^{[k]} p(Y_i|\hat{\theta}_m^{[k]})}{\sum_{k=1}^{K} \hat{w}_m^{[k]} p(Y_i|\hat{\theta}_m^{[k]})}, \tag{7}$$

where $\hat{w}_m^{[k]}$ corresponds to the a priori probability that $Z_i^{[k]} = 1$ in the m-th iteration of the basic EM algorithm over Y, while $\Gamma_i^{[k]}$ is the a posteriori probability that $Z_i^{[k]} = 1$, after observing Y_i.

M-Step: Maximizing Q by constructing a Lagrangian function to update the parameter estimation

$$\hat{\theta}_{m+1} = \arg\max_{\theta} Q(\theta, \hat{\theta}_m), \tag{8}$$

for the ML estimation. In the case of MAP criterion, instead of $Q(\theta, \hat{\theta}_m)$, we need to maximize $\{Q(\theta, \hat{\theta}_m) + \log p(\theta)\}$.

Since in many real world applications, the number of components is unknown and may change over time, and we may also have memory and time constraints, the above EM algorithm, for that type of applications, has to be modified to accommodate those issues and also to be applicable in an on-line context. Before we introduce the proposed algorithm, in the next section, first we briefly describe the criterion that is used in [4] in order to find the number of components in a batch of data. later, we explain how to use this criterion in real time applications.

3.2 The Minimum Description Length (MDL) Principle

The MDL principle is rooted in the fact that any regularity in a given set of data can be used to compress it. Thus, the more regularities there are, the more the data can be compressed. This principle can be also used for inductive inference to the model selection problem [8, 9].
Given a set of hypotheses $\mathcal{H} = \{\mathcal{H}_1, \mathcal{H}_2, \ldots\}$ and a data set Y, the goal is to find the hypothesis or combination of hypotheses in $\mathcal{H}$ that most compress Y.

For the particular case of a data set $Y = \{Y_1, \ldots, Y_n\}$, that has been generated according to Eq.(1)-Eq.(3), which has to be encoded and transmitted, the description length can be obtained as follow [4, 10]:

$$\mathcal{L}(\theta, w, Y) = \frac{N}{2} \sum_{k=1}^{K} \log(\frac{nw^{[k]}}{12}) + \frac{K}{2} \log \frac{n}{12} + \frac{K(N+1)}{2} - \ell \tag{9}$$

where N is a constant that grows quadratically with the dimension d of the data, K is the number of components, n is the total number of samples, $w^{[k]}$ is the mixing weight of the k^{th} component, and $-\ell$ can be viewed as the code-length of the data, given by Eq.(4).

3.3 Titterington's On-line Algorithm for a Multivariate Normal Mixture

As mentioned earlier, the original EM algorithm works in a batch manner. In contrast to the traditional version of the EM, on-line EM variants can flexibly update the parameters of $\mathbf{Y}_n$ as soon as a new sample is observed. In [11], the application of an on-line EM algorithm proposed by Titterington [12] for estimating the multivariate normal mixture in computer vision tasks is investigated. In the proposed on-line algorithm the *Titterington-type on-line parameter recursion* for multivariate normal mixtures are given by

$$\mu_{n+1}^{[k]} = \mu_n^{[k]} + \frac{1}{n} \frac{\Gamma_{n+1}^{[k]}}{w_n^{[k]}} (Y_{n+1} - \mu_n^{[k]}) \tag{10}$$

$$\Sigma_{n+1}^{[k]} = \Sigma_n^{[k]} + \frac{1}{n} \frac{\Gamma_{n+1}^{[k]}}{w_n^{[k]}} \left[(Y_{n+1} - \mu_n^{[k]})(Y_{n+1} - \mu_n^{[k]})^T - \Sigma_n^{[k]} \right] \tag{11}$$

$$w_{n+1}^{[k]} = w_n^{[k]} + \frac{1}{n} (\Gamma_{n+1}^{[k]} - w_n^{[k]}) \tag{12}$$

where Y_{n+1} is the new observation, $\Gamma_{n+1}^{[k]}$ is the a posteriori probability in Eq.(7), n is the time, $w_{n+1}^{[k]}$ is the mixing weight of k^{th} component at time $n+1$, $\mu_{n+1}^{[k]}$ is the updated mean of k^{th} component and $\Sigma_{n+1}^{[k]}$ is the updated covariance of k^{th} component. For more details and the derivation of the formulas see [11].

3.4 Gaussian Mixture Reduction

In this work, we chose a pairwise merging of components method that measures the dissimilarity between the post-merge mixture with respect to the pre-merge mixture based on an easily-computed upper bound of the *Kullback-Leibler* (KL) discrimination measure presented in [13].
Given a mixture of two Gaussian components m, p with the parameters θ and w, where $\theta^{[i]} \equiv \{\mu^{[i]}, \Sigma^{[i]}\}$, $i \in \{m, p\}$ and $w^{[m]} + w^{[p]} = 1$, we can obtain the parameters of merging of these two as follow

$$\mu^{[mp]} = w^{[m]} \mu^{[m]} + w^{[p]} \mu^{[p]} \tag{13}$$

$$\Sigma^{[mp]} = w^{[m]}\Sigma^{[m]} + w^{[p]}\Sigma^{[p]} + w^{[m]}w^{[p]}(\mu^{[m]} - \mu^{[p]})(\mu^{[m]} - \mu^{[p]})^T$$

The KL dissimilarity measure B, between two components m and p can be obtained according to (see [13] for details):

$$2B\left((\mu^{[m]}, \Sigma^{[m]}, w^{[m]}), (\mu^{[p]}, \Sigma^{[p]}, w^{[p]})\right) = tr({\Sigma^{[mp]}}^{-1}\breve{\Sigma}^{[mp]})$$
$$+(w^{[m]} + w^{[p]})\log\det(\Sigma^{[mp]}) - w^{[m]}\log\det(\Sigma^{[m]}) - w^{[p]}\log\det(\Sigma^{[p]}) \quad (14)$$

where

$$\breve{\Sigma}^{[mp]} = w^{[m]}\Sigma^{[m]} + w^{[p]}\Sigma^{[p]} - (w^{[m]} + w^{[p]})\Sigma^{[mp]} + \frac{w^{[m]}w^{[p]}}{w^{[m]} + w^{[p]}}(\mu^{[m]} - \mu^{[p]})(\mu^{[m]} - \mu^{[p]})^T$$

Algorithm 1 On-line EM-based Clustering

Input: Sample data: $Y_0, Y_1, \ldots,$
Mean and covariance of the initial component $\theta^{[0]}$:$(\mu^{[0]}, \Sigma^{[0]})$,
Maximum number of hypotheses: $\aleph_{max}$,
Dissimilarity measure threshold: B_{max}
Output: Number of the components: K,
Mean and covariance of the components: $\{\theta^{[1]}, \ldots, \theta^{[K]}\}$,
Mixing weights: $\{w^{[1]}, \ldots, w^{[K]}\}$
1.Update the current components in each hypothesis $\mathcal{H}_1, \ldots, \mathcal{H}_\aleph$**:**
Find the a posteriori probabilities Γ_{n+1} as Eq.(7)
Update the current components by Eq.(10)-Eq.(12)
Update the log-likelihood ℓ as in Eq.(16)
Update the description length $\mathcal{L}$ as in Eq.(9)
2.Add a new hypothesis: $\mathcal{H}_{\aleph+1}$
Create a new component according to Eq.(15)
Define the log-likelihood of this new hypothesis: $\ell_{\aleph+1} = \ell_1$
Obtain the description length of this new hypothesis: $\mathcal{L}_{\aleph+1}$ in Eq.(9)
3.Check if we can add another hypothesis: $\mathcal{H}_{\aleph+2}$
Find B for every pair of components in $\mathcal{H}_1$ according to Eq.(14)
if $min(B) < B_{max}$ **then**
  Merge two components according to Eq.(13)
  Set the log-likelihood of $\mathcal{H}_1$ as the log-likelihood of new hypothesis: $\ell_{\aleph+2} = \ell_1$
  Obtain the description length for this new hypothesis $\mathcal{L}_{\aleph+2}$ in Eq.(9)
end if
4. Refresh the model
Re-order incrementally the hypotheses according to their description length $\mathcal{L}$
Keep the first $\aleph_{max}$ hypotheses
Acquire the next sample and go to 1

4 The Proposed On-line Unsupervised Learning Algorithm

In this section we describe the proposed on-line unsupervised learning algorithm, which is composed of several models as it will become clear later. Algorithm 1 describes the pseudo-code for one model and its rational is as follows:

- Start with one single observation and build the first hypothesis $\mathcal{H}_1$ described by a single Gaussian distribution with mean $\mu^{[0]}$ at the point itself, and some predefined covariance $\Sigma^{[0]}$. Then, calculate the log-likelihood of this hypothesis $\ell_1 = -\log\sqrt{(2\pi)^d|\Sigma|}$ and find the corresponding description length $\mathcal{L}_1$ according to Eq.(9).

- The second acquired sample, updates the first hypothesis $\mathcal{H}_1$ according to Eq.(10)-Eq.(12), and builds the second hypothesis $\mathcal{H}_2$ which contains two components: the first updated component and a second component with mean $\mu^{[K+1]}$ at the point itself, with some predefined covariance $\Sigma^{[K+1]}$

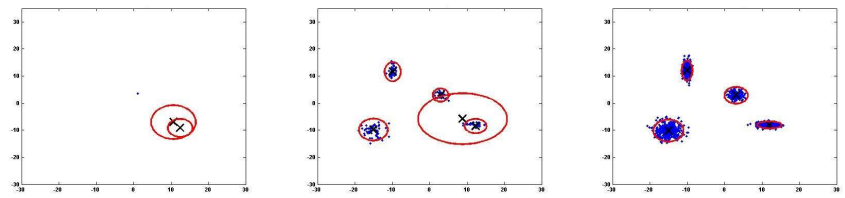
$$\mu^{[\mathrm{K+1}]} = Y_{n+1} \quad , \quad w^{[\mathrm{K+1}]} = \frac{1}{n+1} \tag{15}$$

  where K is the number of components at the time (being $K = 1$ for the case of the second sample).
- The third point will update the two current hypotheses and build another one by adding a new component to $\mathcal{H}_1$ and so on and so forth. For the sake of computational speed and memory, the number of hypotheses has to be bounded. Thus, after reaching the limit of maximum hypotheses $\aleph_{max}$, we rank the hypotheses in an increasing order according to their description length and keep only the first $\aleph_{max}$ hypotheses and discard the rest.
- As explained above, in each iteration we add a new hypothesis by assuming that the new arriving point is a new component, according to Eq.(15), beside the current Gaussian mixture in $\mathcal{H}_1$. Thus, it is likely that we face the very common problem of over fitting. To avoid that, in each iteration after updating the current components in all hypotheses, we check the possibility of adding another hypothesis by merging two most similar components in $\mathcal{H}_1$ (the hypothesis with minimum description length), according to the dissimilarity measure B_{max}. For example at time n, if there were 5 components in $\mathcal{H}_1$, by receiving a new point Y_{n+1}, first we would update the components in $\mathcal{H}_1$ as we do in all other hypotheses; then if there were two similar components according to a threshold in $\mathcal{H}_1$, we would merge them and add another hypothesis $\mathcal{H}_{\aleph+2}$ (see Algorithm 1) composed by the post-merge mixture. For this new hypothesis the log-likelihood is set to be the same as the log-likelihood of the pre-merge mixture in $\mathcal{H}_1$, since it is assumed that the two components were very similar to each other.
- The dissimilarity measure threshold B_{max} is an important quantity since a very small value would not be helpful in tackling the over-fitting problem and setting a very high threshold can cause under-fitting of the components. To address this problem, we propose to run different set of models in parallel, that is, several processes using Algorithm 1 but with different values of B_{max}. For each time n the model with MDL is selected as output.

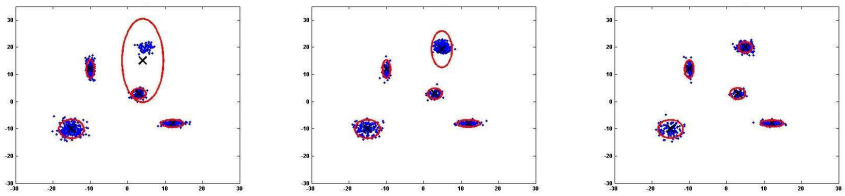
Another point that needs to be taken into consideration is the fact that the computation of the log-likelihood has to be done in a recursive on-line format. Thus, after updating θ_n and w_n, the log-likelihood ℓ from Eq.(4) is updated as

$$\ell_{n+1} = \ell_n + \log p(Y_{n+1}|\theta_n, w_n) \tag{16}$$

where $\theta_n = \{\theta_n^{[1]}, ..., \theta_n^{[K]}\}$ and $w_n = \{w_n^{[1]}, ..., w_n^{[K]}\}$.
For some practical reasons, in Eqs. (10), (11), (12), we changed the *learning rate* $\frac{1}{n}$ to a faster decaying envelope, i.e. we added a sufficiently large enough constant to n in order to reduce the problem of instability as proposed in [3].

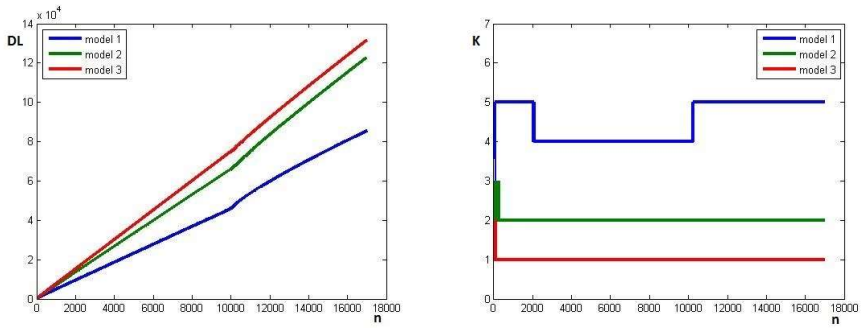


(a) n=3 (b) n=1100 (c) n=4000



(d) n=10300 (e) n=12000 (f) n=15000

Fig. 1: An example of the execution behaviour of the proposed algorithm.



(a) Description Length (DL) (b) Number of Components (K)

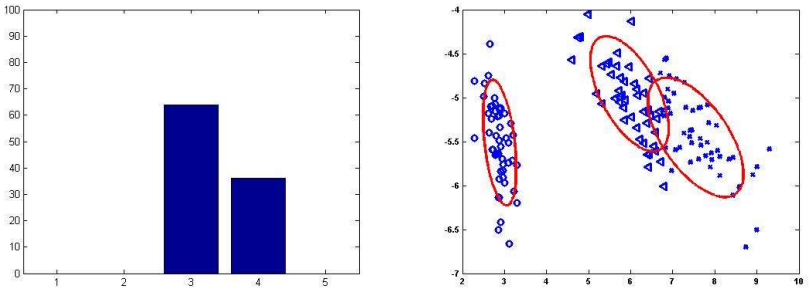
Fig. 2: Time evolution of the description length (DL) and K.

5 Simulation Results

This section illustrates the behaviour of the proposed algorithm for two types of experiments: a synthetic Gaussian mixture data set and the Iris data set.

5.1 A Gaussian Mixture Data Set

Fig.1 shows an example of 3 models running in parallel in order to find a mixture of well separated synthetic Gaussian components in real time starting with one



(a) Histogram results (100 trials) (b) Projected data in 2-D

Fig. 3: Iris Data Set Results

single observation. The maximum number of hypotheses was set to $\mathcal{N}_{max} = 10$ and the merging threshold B_{max} to 0.008, 0.08, and 0.8, respectively. This experiment can be split in two steps. For $n < 10000$, the observed data were randomly extracted, according to Eq.(1) with 4 components (K=4) and the mixing weights of the components from left to right are $w = [0.35, 0.25, 0.15, 0.25]$. It can be seen, after some transient situation, that the algorithm merged the two most similar components and were able to correctly determine the 4 components. Then, for $n \geq 10000$, we started to extract data from another component beside those previous ones. The algorithm was able to converge to the solution rapidly. Fig.2 shows the output of 3 different models running in parallel. Intuitively we can say that setting a higher threshold in models 2 and 3, led to early merging in the components and in turn smaller estimation for K. This problem which is known as under-fitting, can cause reduction in the log-likelihood and increase the description length in turn.

5.2 The Iris Data Set

We used the well-known 3-component 4-dimensional "Iris" data set [14]. This data set has only 150 samples, and therefore we had to randomize and repeat them 60 times. We set the maximum number of hypotheses $\mathcal{N}_{max} = 50$ in 10 different models with the merging threshold starting from $B_{max} = 0.002$. Fig.3(a) shows that in 64 out of 100 trials the 3 components were correctly identified. By visual inspection we could observe that the linearly separated component (iris setosa) could almost perfectly be identified. On the other hand, the properly identification of the other two non-linear separable components (iris versicolor and iris virginca) was more challenging since the order in which the data is presented can influence the recursive solution. The typical solution is shown in Fig.3(b) by projecting the 4-dimensional data to the first two principal components.

6 Conclusion

This paper proposed an on-line unsupervised learning of GMMs algorithm in the presence of uncertain dynamic environments. The algorithm relies on a multi-hypothesis adaptive scheme that continuously updates the number of components and estimates the model parameters as the measurements (sample data) are being acquired. The hypothesis models are ranked according to the MDL. In general, we could conclude that the algorithm has a good performance specially when the components are well separated. However, it is worth to mention that a critical issue is the initial selection of the covariance when a new component is created. This has to be done carefully because choosing a very small covariance can be experimentally problematic since in the process of calculating the a posteriori probability in Eq.(7), the result in Eq.(3) could be zero due to finite precision. On the other hand, choosing an extremely large covariance can lead to the "under-fitting" problem. This is something that deserves further investigation.

References

1. Bishop, C.M.: Pattern recognition and machine learning, Springer (2006)
2. Greggio, N., Bernardino, A., Victor, J.S.: A practical method for self adapting gaussian expectation maximization. In: ICINCO. (2010) 36–44
3. Zivkovic, Z., van der Heijden, F.: Recursive unsupervised learning of finite mixture models. (2004)
4. Figueiredo, M.A.T., Jain, A.K.: Unsupervised learning of finite mixture models. Volume 24. (2000) 381–396
5. Declercq, A., Piater, J.H.: Online learning of gaussian mixture models − a two level approach. (2008) 605–611
6. P., D.A., M., L.N., B., R.D.: Maximum likelihood from incomplete data via the EM algorithm. Volume 39. (1977) 1–38
7. McLachlan, G., Krishnan, T.: The EM Algorithm and Extensions. John Wiley & Sons, New York (1997)
8. Lanterman, A.D.: Schwarz wallace and rissanen: Intertwining themes in theories of model selection. (2000)
9. Grünwald, P.D.: The minimum description length principle. The MIT Press (2007)
10. Raudys, S.J., Jain, A.K.: Small sample size effects in statistical pattern recognition: Recommendations for practitioners. IEEE Transactions on Pattern Analysis and Machine Intelligence **13**(3) (1991) 252–264 cited By (since 1996)462.
11. Li, D., Xu, L., Goodman, E.: On-line EM variants for multivariate normal mixture model in background learning and moving foreground detection. (2012)
12. Titterington, D.M.: Recursive parameter estimation using incomplete data. Volume 46. (1984) 257–267
13. Runnalls, A.R.: A kullback-leibler approach to gaussian mixture reduction. Volume 43. (2007) 989–999
14. Iris data set. http://archive.ics.uci.edu/ml/datasets/Iris