An Online EM Algorithm Using Component Reduction

Nobuo Suematsu, Kumiko Maebashi, and Akira Hayashi
Graduate School of Information Sciences,
Hiroshima City University, 731-3194, Japan
suematsu@im.hiroshima-cu.ac.jp, bochin@robotics.im.hiroshima-cu.ac.jp, akira@im.hiroshima-cu.ac.jp

Abstract. The EM algorithm has been widely used in many learning or statistical tasks. However, since it requires multiple database scans, applying the EM algorithm to data streams is not straightforward. In this paper we propose an online EM algorithm which can deal with data streams. The algorithm utilizes a component reduction technique which reduces the number of components in a mixture model. A notable advantage of our algorithm over existing online variants of the EM algorithm lies in its simplicity. Our algorithm almost preserves the theoretical clearness of the EM algorithm.

1 Introduction

The EM algorithm[1] is a general method to estimate the parameters of a stochastic model from data which can be viewed as being incomplete. The fitting of mixture models by maximum likelihood is a typical example of a problem which is simplified considerably by the EM algorithm. Mixture modeling using the EM algorithm has been widely used in many learning or statistical tasks such as density estimation, clustering, classification, and data summarization.

However, since the EM algorithm is a batch algorithm in which whole training data is scanned at every iteration to improve its estimation, applying the EM algorithm to data streams is not realistic. In order to cope with data streams, a learning algorithm should work in an online fashion. In addition, memory space required by the algorithm should be constant with respect to number of data points processed so far.

In this paper, we present an online variant of the EM algorithm. The algorithm requires a component reduction technique which is used to incorporate new data into the maintained mixture model. We use the extension of the EM algorithm developed by Vasconcelos and Lippman[2] as a component reduction technique. Through the paper, their algorithm will be referred to as HEM (hierarchical EM) for short. Our algorithm is an online algorithm, that is, it can discard data once processed. In addition, our algorithm works within a confined memory space whose size is determined by choosing some design parameters.

A notable advantage of our algorithm over existing online variants of the EM algorithm (e.g. [3],[4]) lies in its simplicity. Our algorithm does not rely on other clustering or data summarization techniques. So our algorithm preserves the clearness and simplicity of the EM algorithm and we can expect that it will be able to cope with wide variety of applications fairly well with little tuning.
2 The Standard EM Algorithm

In this section we describe mixture model estimation using the standard EM algorithm (e.g. [5]). Let us consider to approximate the data distribution by a mixture model,

\[ f(x|M) = \sum_{i=1}^{C} \pi_i p_i(x|\theta_i), \]  

(1)

where \( p_i(x|\theta_i) \) are densities with parameter vector \( \theta_i \), \( \pi_i \) are nonnegative quantities which sum to one, that is, \( 0 \leq \pi_i \leq 1 \) \( (i = 1, \ldots, C) \) and \( \sum_{i=1}^{C} \pi_i = 1 \). In addition, \( M = \{ \pi_1, \ldots, \pi_C, \theta_1, \ldots, \theta_C \} \) is the set of all parameters in the mixture model.

Given a set of data points, \( X = \{ x_1, \ldots, x_N \} \), we seek to find a set of parameters \( M \) which has a large likelihood for \( X \). The EM algorithm gives us an efficient approach for this problem. When we apply the EM algorithm, it is assumed that each data point \( x_n \) has been drawn from one of the components of the mixture model. The component from which \( x_n \) is arisen is represented by variables \( z_{ni} \) \( (i = 1, \ldots, C) \). \( z_{ni} \) takes value 1 if \( x_n \) is drawn from \( i \)-th component and 0 otherwise.

The EM algorithm treats \( z_{ni} \) as hidden variables. Let \( Z = \{ z_{ni} | n = 1, \ldots, N, i = 1, \ldots, C \} \), then the log likelihood of \( M \) for the complete data \( X \) and \( Z \), is given by

\[ L_c(M) = \sum_{n=1}^{N} \sum_{i=1}^{C} h_{ni} \{ \log \pi_i + \log p_i(x_n|\theta_i) \}. \]

In the procedure called E-step, the conditional expectation of \( L_c(M) \) is taken given the observed data \( X \) using the current estimate \( M \) denoted by \( M^{(t)} \). This yields

\[ Q(M; M^{(t)}) = E[L_c(M)|X, M^{(t)}] = \sum_{n=1}^{N} \sum_{i=1}^{C} h_{ni} \{ \log \pi_i + \log p_i(x_n|\theta_i) \}, \]

(2)

where \( h_{ni} = E[z_{ni}|x_n, M^{(t)}] \). Then in the procedure called M-step, \( Q(M; M^{(t)}) \) is maximized with respect to \( M \) and a new estimate \( M^{(t+1)} \) is obtained. Under many popular choices for the component model, \( M^{(t+1)} \) has closed form.

Starting with an initial guess \( M^{(0)} \), the EM algorithm generates successive estimates, \( M^{(1)}, M^{(2)}, \ldots \) by iterating E-step and M-step. The iteration is terminated when the sequence of estimates is converged.

3 Online EM Algorithm

We propose an online EM-algorithm which uses a component reduction technique. Given a database, our algorithm maintains a mixture model which approximates the density function of the data. For simplicity, in the rest of the paper, we restrict our discussion to the case when data points are \( d \)-dimensional vectors and components of mixture models are Gaussian.

The pseudo code of the algorithm is shown in Figure 1. In our algorithm, every \( b \) data points obtained from a data stream is treated as a block. The initial \( C \)-component
mixture model is constructed from the first block using the standard EM-algorithm (line 2). The mixture model is updated every time next data block is received. In the update process, an intermediate mixture model $\mathcal{M}^{C+}$ is constructed by merging the current mixture model $\mathcal{M}^C$ and the block of $b$ data points $\mathcal{X}^b$ (line 6), where the number of components $C_+$ of $\mathcal{M}^{C+}$ is larger than $C$. Then, a new $C$-component mixture model $\mathcal{M}^C$ is obtained by reducing the number of components in the intermediate mixture model $\mathcal{M}^{C+}$ (line 8). The method used for this reduction of the number of components is crucial for our algorithm. We will describe a method for component reduction in the following subsection.

We have some alternatives in the way to merge a mixture model $\mathcal{M}^C$, and a set of data points, $\mathcal{X}^b$, at line 6 in the pseudo code of Figure 1. In the paper we investigate two approaches to the task which are described in Subsection 3.2.

### 3.1 Component Reduction

We use HEM to reduce the number of components in a mixture model. HEM has been proposed by Vasconcelos and Lippman[2] as the learning algorithm for hierarchical mixture models.

A hierarchical mixture model is a tree structured model whose each level represents a mixture model. In a hierarchical mixture model, mixture models at higher levels have fewer components.

Nodes at level $l$ one to one correspond to the components of the mixture model at the level. In addition, each node (or component) at level $l$ is represented by a mixture model composed of its child nodes at level $l+1$. Thus, the hierarchical mixture model is tree structured. To represent the tree structure, binary variables $z^l_{ij}$ are introduced which are defined so that

$$z^l_{ij} = \begin{cases} 
1 & \text{if component } j \text{ at level } l \text{ is the parent of component } i \text{ at level } l+1, \\
0 & \text{otherwise}.
\end{cases}$$
Let $C_l$ be the number of nodes at level $l$ and $\mathcal{M}^{C_l}$ be the mixture model at level $l$. HEM is developed as an extension of the EM algorithm which learns mixture model $\mathcal{M}^{C_l}$ from $\mathcal{M}^{C_{l+1}}$. HEM can be derived by considering a virtual sample generated from the mixture model $\mathcal{M}^{C_{l+1}}$ and applying EM-algorithm in which $z_{ij}^l$ are treated as latent variables. For Gaussian mixture models, they derived the formula for $h_{ij}^l = E[z_{ij}^l|M^{C_l}, \mathcal{M}^{C_{l+1}}]$ which is used in the E-step, and update rules for $M_l$ which is used in the M-step.

Since $C_l < C_{l+1}$, we can use HEM as a component reduction technique. A pseudo code of a component reduction method based on HEM is shown in Figure 2.

### 3.2 Merging of a Mixture Model and Data

A straightforward way to merge a $C$-component mixture model and $b$ data points is to use the empirical density function of the data points. We call this method direct approach.

Instead of using the empirical density of data points, $X^b$, we can use a temporal mixture model learned by the standard EM-algorithm from $X^b$. We refer this method as indirect approach. In this approach, we can choose the number of components of the temporal mixture model arbitrarily between 1 and $b$ in principle. In this paper, we investigate only when $C$-component temporal mixture models are used in this approach.

### 4 Experimental Results

As described in Section 3, our algorithm arises two variations, one of which uses the direct approach and the other uses the indirect approach to merge the current estimate and data. In this section, the variations which uses the direct and the indirect approaches are denoted by dOEM and iOEM respectively. We applied these two variations, dOEM and iOEM, and the standard EM to the data stream drawn from the 2-dimensional 3-component Gaussian mixture model in which each component’s covariance matrix is the identity matrix and components’ means are located on a circle of radius 2 isotropically. In addition, mixing rates are $1/3$.

All algorithms learned 3-component Gaussian mixture models, i.e. $C = 3$. To give a simple comparison, we introduce the following error measure for learned mixture models error $=$ $\sum_{i=1}^{3} \| \mu_i' - \mu_i \| + \sum_{i=1}^{3} \sqrt{(a_i - 1)^2 + (b_i - 1)^2}$, where $\mu_i'$ are learned means and $a_i, b_i$ are halves of major and minor axes of learned covariance matrices. The component labels were chosen so that the error measure was minimized.

The obtained learning curves are shown in Figure 4. The accuracies achieved by dOEM were relatively low. dOEM was evaluated when block size $b = 200, 500, 1000$, and it achieved best accuracies when $b = 500$. The learning curve for $b = 500$ is solely shown for dOEM. Note that, in our implementation eigenvalues of covariance matrices were lower bounded by $10^{-3}$, and empirical densities were composed of normal distributions whose covariances are $10^{-3}I$.

For iOEM, the accuracies improves as block size increases. When $b = 1000$, the errors for iOEM is almost identical to those for the standard EM.
Fig. 3. Learning curves.

5 Conclusion

In this paper, we proposed an online EM algorithm which uses a component reduction technique. As the component reduction technique we used the extension of the EM algorithm proposed by Vasconcelos and Lippman[2]. Since our algorithm does not rely on other type of clustering or data summarization techniques, it preserves the theoretical clearness and the simplicity of the standard EM algorithm. Our algorithm learns a mixture model by constructing an intermediate mixture model which obtained by merging the current estimate and the next block of data points then reducing the number of components of the intermediate mixture model.

The two variants of our algorithm were empirically investigated. The empirical results indicate that one of these variants can be comparable to the standard EM, in spite of the fact that it can discard data once processed and can work within a confined memory space.

References