Split and Merge EM Algorithm for Improving Gaussian Mixture Density Estimates

NAONORI UEDA AND RYOHEI NAKANO
NTT Communication Science Laboratories, Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0237, Japan

ZOUBIN GHAHRAMANI AND GEOFFREY E. HINTON
Gatsby Computational Neuroscience Unit, University College London, 17 Queen Square, London WC1N 3AR, UK

Received March 25, 1999; Revised September 28, 1999

Abstract. The EM algorithm for Gaussian mixture models often gets caught in local maxima of the likelihood which involve having too many Gaussians in one part of the space and too few in another, widely separated part of the space. We present a new EM algorithm which performs split and merge operations on the Gaussians to escape from these configurations. This algorithm uses two novel criteria for efficiently selecting the split and merge candidates. Experimental results on synthetic and real data show the effectiveness of using the split and merge operations to improve the likelihood of both the training data and the held-out test data.

1. Introduction

Gaussian mixtures have been extensively used in the field of statistical pattern recognition including neural networks [1–4]. The EM algorithm [5] has been well known as a convenient and efficient tool to iteratively compute the maximum likelihood estimates of Gaussian mixtures. There are, however, two serious problems in practice: singularities and local maxima. Although these problems have been pointed out by many researchers, the best way to solve them in practice is still an open question.

Ormoneit and Tresp [3] have recently proposed some sophisticated regularization methods to solve the singularity problem. Regarding the local maximum problem, two of the authors have proposed the deterministic annealing EM (DAEM) algorithm [6, 7], where a modified posterior probability parameterized by temperature is derived to avoid local maxima. In the case of Gaussian mixture density estimation, local maxima arise when there are too many Gaussians in one part of the space and too few in another. It is not possible to move a Gaussian from the overpopulated region to the underpopulated region without passing through positions that give lower likelihood. We therefore introduce a discrete move that simultaneously merges two Gaussians in an overpopulated region and splits a Gaussian in an underpopulated region without changing the number of Gaussians.

The idea of split and merge operations has been successfully applied to clustering or vector quantization (e.g., [8]). In this paper, we try to incorporate the split and merge operations into the EM algorithm for Gaussian mixture density estimates to overcome the local maxima problem. New criteria presented in this paper can efficiently select the split and merge candidates. Although the proposed method, unlike the DAEM algorithm, is limited to mixture models, we show experimentally that our split and merge EM algorithm obtains better solutions than the DAEM algorithm.

2. Gaussian Mixture Density Estimation via the EM Algorithm

The probability density function (pdf) of a finite Gaussian mixture is

\[ p(x; \Theta) = \sum_{m=1}^{M} \alpha_m g(x; \mu_m, \Sigma_m). \]  

(1)
where \( \alpha_m, m = 1, \ldots, M \) are mixing proportions and satisfy
\[
\alpha_m \geq 0 \quad \text{and} \quad \sum_{m=1}^{M} \alpha_m = 1. \tag{2}
\]
The \( g(x; \mu_m, \Sigma_m) \) is a \( d \)-dimensional normal density corresponding to the \( m \)th component given by:
\[
g(x; \mu_m, \Sigma_m) = (2\pi)^{-\frac{d}{2}} \det(\Sigma_m)^{-\frac{1}{2}} \times \exp \left\{ -\frac{1}{2} (x - \mu_m)^T \Sigma_m^{-1} (x - \mu_m) \right\}.
\]
Here \( \det(A) \) is the determinant of matrix \( A \) and \( T \) denotes the transpose operation. A set of unknown parameters, in this case, is \( \Theta = \{ (\alpha_m, \mu_m, \Sigma_m), m = 1, \ldots, M \} \).

Given a set of iid data \( \mathcal{X} = \{ x_1, \ldots, x_N \} \), the maximum likelihood estimate of the unknown parameters \( \Theta \) is efficiently obtained by the EM algorithm [5]. In the EM algorithm, the parameters \( \Theta \) are iteratively estimated by using two steps, E (for Expectation) and M (for Maximization). The E-step computes the expectation of the complete data log-likelihood using the posterior probability that \( x \) belongs to the \( m \)th component based on the current parameters \( \Theta^{(t)} \):
\[
Q(\Theta | \Theta^{(t)}) = \sum_{x \in \mathcal{X}} \sum_{m=1}^{M} P(m | x; \Theta^{(t)}) \log \alpha_m g(x; \mu_m, \Sigma_m), \tag{3}
\]
where
\[
P(m | x; \Theta^{(t)}) = \frac{\alpha_m^{(t)} g(x; \mu_m^{(t)}, \Sigma_m^{(t)})}{\sum_{l=1}^{M} \alpha_l^{(t)} g(x; \mu_l^{(t)}, \Sigma_l^{(t)})}. \tag{4}
\]
Next, the M-step maximizes this \( Q \) function with respect to \( \Theta \) to estimate the new parameter values \( \Theta^{(t+1)} \). More specifically, we rewrite (3) as
\[
Q(\Theta | \Theta^{(t)}) = \sum_{m=1}^{M} \sum_{x \in \mathcal{X}} P(m | x; \Theta^{(t)}) \log \alpha_m \\
+ \sum_{m=1}^{M} \sum_{x \in \mathcal{X}} P(m | x; \Theta^{(t)}) \log g(x; \mu_m, \Sigma_m). \tag{5}
\]
Then, by maximizing the first term of the right hand side of (5) with respect to \( \alpha_m \), subject to (2), we have
\[
\alpha_m^{(t+1)} = \frac{1}{N} \sum_{x \in \mathcal{X}} P(m | x; \Theta^{(t)}). \tag{6}
\]
Next, by maximizing the second term of the right hand side of (5) with respect to \( \mu_m \) and \( \Sigma_m \), respectively, we have
\[
\mu_m^{(t+1)} = \frac{\sum_{x \in \mathcal{X}} x P(m | x; \Theta^{(t)})}{\sum_{x \in \mathcal{X}} P(m | x; \Theta^{(t)})}, \tag{7}
\]
\[
\Sigma_m^{(t+1)} = \frac{\sum_{x \in \mathcal{X}} (x - \mu_m^{(t+1)})(x - \mu_m^{(t+1)})^T P(m | x; \Theta^{(t)})}{\sum_{x \in \mathcal{X}} P(m | x; \Theta^{(t)})}. \tag{8}
\]
To prevent the covariance from being singular, the following update rule based on the Bayesian regularization is available [3]:
\[
\Sigma_m^{(t+1)} = \sum_{x \in \mathcal{X}} (x - \mu_m^{(t+1)})(x - \mu_m^{(t+1)})^T P(m | x; \Theta^{(t)}) + \lambda I_d, \quad \sum_{x \in \mathcal{X}} P(m | x; \Theta^{(t)}) + 1 \tag{9}
\]
where \( I_d \) is the \( d \)-dimensional unit matrix and \( \lambda \) is a regularization constant determined by some validation data.

3. Split and Merge EM Algorithm

3.1. The Algorithm

Let \( \Theta^* \) denote the parameter values estimated by the usual EM algorithm. Then after the EM algorithm converged, (3) can be rewritten in the form of a direct sum:
\[
Q^* = Q^*_s + Q^*_j + Q^*_c + \sum_{m, m' j, k} Q^*_m, \tag{10}
\]
where
\[
Q^*_m = \sum_{x \in \mathcal{X}} P(m | x; \Theta^*) \log \alpha_m^* g(x; \mu_m^*, \Sigma_m^*). \tag{11}
\]
We then try to increase the first term of the right-hand side of (10) by merging the \( i \)th and \( j \)th Gaussians to produce the \( i' \)th Gaussian, and splitting the \( k \)th Gaussian into the \( j' \)th and \( k' \)th Gaussians. The split and merge operations are simultaneously performed so that the number of components is unchanged. That is, we note that our goal here is not to solve the model selection problem, but to solve the local maxima problem.