A Dynamic Merge-or-Split Learning Algorithm on Gaussian Mixture for Automated Model Selection*

Jinwen Ma and Qicai He

De a X e X f I f aX Sc e ce, Sch f Ma X re aXca
Sc e ce a d LMAM, Pe g U e X, Be g, 100871, Ch a
jwma@math.pku.edu.cn

Abstract. Ga a X e de g a e f X f daXa a a - . H e e , X e e ec X f be f Ga a X e X e, .e., X e X e de ca e ec X ; e a a d ffic X b e . I X a- e , e e a e d f d a c e ge- - X ea g (DMOSL) a g X Ga a X e er XaXXe be f Ga a ca be de d ed a X a ca X a d a c e ge- - X e aX a g e X aXed Ga a f X e EM g X . IX de XaXed b X e aX e e e X aXXe DMOSL a g X ca a X- aXca deXe e X e be f Ga a a a e daXa eX a d a ead X a g d e X aX f X e a a eXe X e g a X e. M e e , X e DMOSL a g X a ed X X e c a fica X f I daXa.

1 Introduction

Many problems in data analysis, especially in clustering analysis and classification, can be solved through Gaussian mixture model [1]. Actually, several statistical methods have been proposed for Gaussian mixture modelling (e.g., the EM algorithm [2] and k-means algorithm [3]). But it is usually assumed that the number k of Gaussians in the mixture is given in advance. However, in many cases, this key information is not available and the selection of an appropriate number of Gaussians must be made with the parameter estimation, which is a rather difficult task [4].

The traditional approach to this task is to choose a best k^* via some selection criterion, such as the Akaike's information criterion [5] or its extensions. However, these methods incur a large computational cost since we need to repeat the entire parameter estimation process independently at a number of different values of k. Moreover, all these criteria have their limitations and often lead to a wrong result.

Recently, a new kind of automated model selection approach has been developed using the idea that an appropriate number of Gaussians can be automatically allocated during the parameter learning, with the mixing proportions of

* T a Xed b X e Na Xa Sc e ce F daX f Ch a f P ecX 60471054

M. Gallagher, J. Hogan, and F. Maire (Eds.): IDEAL 2005, LNCS 3578, pp. 203–210, 2005.

the extra Gaussians attenuating to zero. From the Bayesian Ying-Yang (BYY) harmony learning theory, the gradient-type harmony learning algorithms [6]-[7]

2.1 Ga ian Mi e M del

We begin to introduce the Gaussian mixture model as follows:

$$P(x|\Theta) = \sum_{i=1}^{k} \alpha_i P(x|m_i, \Sigma_i), \qquad \alpha_i \geq 0, \quad \sum_{i=1}^{K} \alpha_i = 1, \tag{1}$$

where

$$P(x|m_i, \Sigma_i) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma_i|^{\frac{1}{2}}} e^{-\frac{1}{2}(\ -m_i)^T \Sigma_i^{-1}(\ -m_i)} \tag{2}$$

and where k is the number of Gaussians or components in the mixture, x denotes a sample vector and d is the dimensionality of x. The parameter vector Θ consists of the mixing proportions α_i, the mean vectors m_i, and the covariance matrices $\Sigma_i = (\sigma_{pq}^{(i)})_{d\times d}$ which are assumed positive definite.

For a sample data set $\mathcal{S} = \{x_i\}_{i=1}^{N}$ from the Gaussian mixture, we define the posteriori probability of a sample x_i over the j-Gaussian or component as follows.

$$P(j|x_i; \Theta) = \frac{\alpha_j P(x_i|m_j, \Sigma_j)}{P(x_i|\Theta)} = \frac{\alpha_j P(x_i|m_j, \Sigma_j)}{\sum_{i=1}^{k} \alpha_i P(x_i|m_i, \Sigma_i)}. \tag{3}$$

According to these posteriori probabilities, we can divide the sample points into k clusters corresponding to the k Gaussians in the mixture by

$$G[j] = \{x_i : P(j|x_i; \Theta) = \max_{i=1,\cdots,k} P(i|x_i; \Theta)\}. \tag{4}$$

2.2 The Me ge and S li Cr i e ia

We further introduce the merge and split criteria on the estimated Gaussians after the EM algorithm has converged. Actually, via the estimated parameters, we can obtain the clusters $G[j]$. For the merge or split operation, we first check whether the sample points in two or more neighboring clusters are subject to a Gaussian distribution. If they are, we think the corresponding estimated Gaussians should be merged. Furthermore, we check whether the sample points in each remaining $G[j]$ (excluding these ones to be merged) are subject to a Gaussian distribution. If they are not, the estimated Gaussian should be split.

Specifically, we give the merge and split criteria as follows.

Me ge Cr i e i n: For the $i-$th and $j-$th estimated Gaussians, we introduce the following merge degree:

$$J_{me\ ge}(i, j; \Theta) = \frac{P_i(\Theta)^T P_j(\Theta)}{\|P_i(\Theta)\| \|P_j(\Theta)\|} \tag{5}$$

where $P_l(\Theta)$ is an N-dimensional vector consisting of posterior probabilities of the sample points over the $l-$th Gaussian, and $\|\cdot\|$ denotes the Euclidean vector

norm. Clearly, when the two estimated Gaussians should be merged together, $P_i(\Theta)$ and $P_j(\Theta)$ should be similar at a ceratin degree so that $J_{me\ ge}(i,j;\Theta)$ will be high. According to this merge degree and a threshold value $\alpha > 0$, we have the merge criterion: if $J_{me\ ge}(i,j;\Theta) \geq \delta$, these two Gaussians will be merged together, otherwise, they will not.

In the simulation experiments, we found that $J_{me\ ge}$ has a relationship with N. So, by experience, we set $\delta = 0.004N^{1/2}$ in the following experiments. Moreover, we also found in the simulation experiments that if the two estimated Gaussians should not be merged, $J_{me\ ge}$ becomes very small (in general, $J_{me\ ge}(i,j;\Theta) < 10^{-3}$). Therefore, the merge degree is reasonable.

S li C i e i n: We use the Srivastav method [10] to check the normality for the sample points in each remaining cluster. In fact, via the singular value decomposition, the Srivastav method turns the test of multivariate normality into the test for a number of independent normal variables. For the test of univariate normality, we implement the Kolmogorov-Smirnov test. For the $j-$th estimated Gaussian (remaining from the merge criterion), according to the Srivastav method, if the sample points in $G[j]$ are not subject to a normal distribution, it will be split into two Gaussians; otherwise, there will be no need for the split on this estimated Gaussian.

2.3 The P ced e f he DMOSL Alg i hm

With the above preparations, we can now present the procedure of the DMOSL algorithm. Structurally, the DMOSL algorithm consists of a number of phases. At the beginning phase, we set k as the best possible estimation of the number of actual Gaussians in the sample data. With this initial k, the EM algorithm is conducted to get the estimated Gaussians. Then, the DMOSL algorithm turns into the second phase. In this or the sequential phase, according to the obtained Gaussians or clusters, we check whether the merge or split operation is needed. If a merge or split operation is needed, we can use the mathematical method proposed in [10] to put the two estimated Gaussians into one or split one estimated Gaussian into two, with the parameters being modified. Starting from the obtained and modified parameters in the new Gaussian mixture setting, the EM algorithm is further conducted to the new estimated Gaussians for the following phase. In this way, the model selection will be made dynamically and automatically during the learning phases via the merge and split operations. Finally, the DMOSL algorithm will be halted when there is no need for the merge or split operation on the estimated Gaussians.

For the fast convergence, we also add a component eliminating mechanism to the DMOSL algorithm on the mixing proportions obtained from the EM algorithm. That is, if $\alpha_j < 0$.

S e 2 At phase t, we perform the EM algorithm starting from the parameters obtained from the last phase after the merge and split operations if $t > 0$ or from the initial parameters if $t = 0$. After the EM algorithm has converged, we get Θ^{ι} at the $t-$th phase. According to Θ^{ι}, we can get the estimated Gaussians and the corresponding clusters $G[j]$. If there is no need for the merge or split operation on the estimated Gaussians, the DMOSL algorithm is halted. Otherwise, we go to the next step.

S e 3 Merge operation: we compute $J_{me\ ge}(i, j; \Theta^{\iota})$ for $i, j = 1, \cdots, k$ and $i \neq j$. and sort them in a descend order. If these exists any $J_{me\ ge}(i, j; \Theta^{\iota})$ that is no less than δ, i.e., $J_{me\ ge}(i, j; \Theta^{\iota}) \geq \delta$, we merge these two Gaussians into a new Gaussian i'. The parameters of this new Gaussian are computed as follows.

$$\alpha_{i'} = \alpha_i + \alpha_j; \tag{6}$$

$$m_{i'} = (\alpha_i m_i + \alpha_j m_j)/\alpha_{i'}; \tag{7}$$

$$\Sigma_{i'} = (\alpha_i \Sigma_i + \alpha_j \Sigma_j + \alpha_i m_i m_i^T + \alpha_j m_j m_j^T - \alpha_{i'} m_{i'} m_{i'}^T)/\alpha_{i'}. \tag{8}$$

It can be found in the experiments that sometimes $\Sigma_{i'}$ may not be positive. In this special case, we can use the covariance matrix of the sample data in $G[i]$ and $G[j]$ instead. If one estimated Gaussian can be merged into two or more estimated Gaussians, we merge the two estimated Gaussians with the highest merge degree. When a merge operation is implemented, k is automatically decreased by one, i.e., $k = k - 1$.

S e 4 Split operation: after the merge operation, there are certain estimated Gaussians remained. For each remaining estimated Gaussian, we check whether it should be split according to the split criterion. If it should be, say the i-th Gaussian, we split it into two Gaussians i' and j' as follows.

From the covariance matrix Σ_j, we have its singular value decomposition $\Sigma_j = USV^T$, where S is a diagonal matrix with nonnegative diagonal elements in a descent order, U and V are two (standard) orthogonal matrices. Then, we further set $A = U\sqrt{S}$ (refer to [10] for the derivation), and get the first column A_1 of A. Finally, we have the parameters for the two split Gaussians as follows.

$$\alpha_{i'} = \alpha_i * \gamma, \alpha_{j'} = \alpha_i * (1 - \gamma); \tag{9}$$

$$m_{i'} = m_i - (\alpha_{j'}/\alpha_{i'})^{1/2} \mu A_1; \tag{10}$$

$$m_{j'} = m_i + (\alpha_{i'}/\alpha_{j'})^{1/2} \mu A_1; \tag{11}$$

$$\Sigma_{i'} = (\alpha_{j'}/\alpha_{i'})\Sigma_i + ((\beta - \beta\mu^2 - 1)(\alpha_i/\alpha_{i'}) + 1)A_1 A_1^T; \tag{12}$$

$$\Sigma_{j'} = (\alpha_{i'}/\alpha_{j'})\Sigma_i + ((\beta\mu^2 - \beta - \mu^2)(\alpha_i/\alpha_{j'}) + 1)A_1 A_1^T, \tag{13}$$

where γ, μ, β are all equal to 0.5.

When a split operation is implemented, k is automatically increased by one, i.e., $k = k + 1$.

S e 5 We let $t = t + 1$ and return to Step 2.

We finally give some remarks on the DMOSL algorithm. (1). The split criterion or operation is based on the test of the normality on the sample points in

the resulted clusters $G[j]$. Actually, only when the number of the sample points from each actual Gaussian is large enough and the actual Gaussians are separated in a certain degree, this normality test can be reasonable and lead to a correct result. Hence, the DMOSL algorithm can be only suitable for the sample data set in which the actual Gaussians have a large number of sample points and are separated in a certain degree. (2). The split criterion is based on the statistical test and the merge criterion is based on the merge degrees between two estimated Gaussians through a threshold value selected by experience. Theoretically, there exists a small probability of the error on the DMOSL algorithm. (3). In Step 4, for consideration of robustness, we can add a checking step on the two split Gaussians to make sure whether this split operation is really necessary. If the two split Gaussians i' and j' on the data set $G[i]$ should be merged under the merge criterion on these two Gaussians only, we abandon the split operation. Otherwise, we keep the split operation. However, it is found in the experiments that this checking step is hardly active.

3 Experimental Results

In this section, several simulation experiments are carried out to demonstrate the DMOSL algorithm for automated model selection as well as parameter estimation on seven data sets from Gaussian mixtures. Moreover, we apply the DMOSL algorithm to the classification of Iris data.

3.1 Sim la i n E , e imen

We conducted several experiments on seven sets of samples drawn from a mixture of four or three bivariate Gaussians densities (i.e., $n = 2$). As shown in Fig. 1, each data set of samples is generated at different degree of overlap among the clusters (Gaussians) and with equal or unequal mixing proportions of the clusters in the mixture.

Using k^* to denote the number of actual Gaussians in the sample data or the original mixture, we implemented the DMOSL algorithm on these seven data sets with different initial values such as $k < k^*$, $k = k^*$ and $k > k^*$. The other parameters were initialized randomly within certain intervals.

Typically, we give the experimental results of the DMOSL algorithm on the sample data set (d) in which $k^* = 4$. For $k = k^* = 4$, the DMOSL algorithm was halted immediately with no merge or split operation and the result is shown in Fig. 2. In this and the following figures, T represents the number of merge and split operations in each phase, k is the initial number of Gaussians, k' is the changing number of estimated Gaussians in each phase of the algorithm. For $k = 1$, the DMOSL algorithm first split one Gaussian into two Gaussians and then split two into four, see Fig.3 for $k' = 2$. On the other hand, when $k = 8$, the 8 estimated Guassians merged into 4 Gaussian only in one phase, and we show the results in Fig.4 for $k' = 8$. From these figures, we can observe that, through the dynamic merge or split operations, the DMOSL algorithm can make model selection automatically on the sample data set and at the same time lead to a good estimation of the parameters in the original mixture.

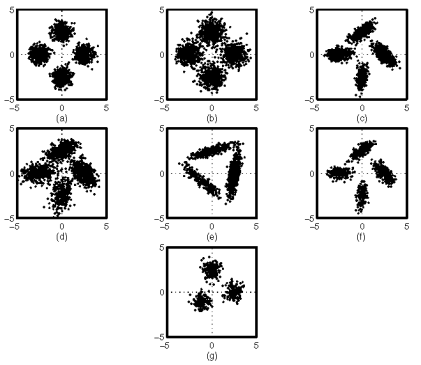

Fig. 1. Se e eX f a e daXa ed X e e e e X

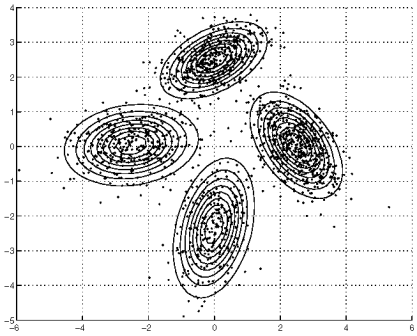

Fig. 2. $k^* = 4, k = 4, k' = 4, T = 0$ (X ed)

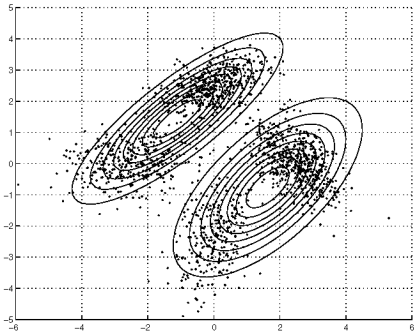

Fig. 3. $k^* = 4, k = 1, k' = 2, T = 2$ (2 X e aX)

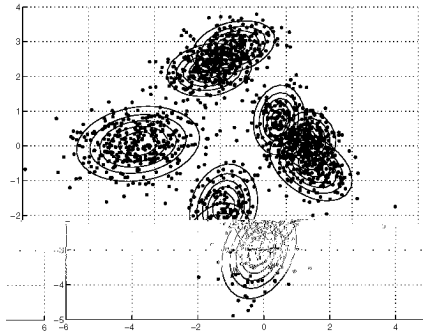

Fig. 4. $k^* = 4, k = 8, k' = 8, T = 4$ (4 e ge e aX)

The further experiments of the DMOSL algorithm on the other sample sets had been also made successfully for the automated model selection and parameter estimation in the similar cases. Since the DMOSL algorithm can escape the local solution with the merge or split operation, it outperforms the conventional EM algorithm. It also outperforms the split-and-merge EM algorithms given in [9]-[10] since it has the ability of automated model selection. As compared with the automated model selection algorithms in [6]-[8],[11], the DMOSL algorithm has no limitation for the initial value of k and converges more quickly in the general case.

3.2 E , e imen n Cla i ca i n f I i Da a

We further apply the DMOSL algorithm to the classification of the Iris data[1] which is a typical real dataset for testing the classification algorithm. The Iris data set consists of 150 4-dimension data from three classes: Iris Versicolor, Iris

[1] ReX e ed f XX :// . c .ed/ ea /MLRe X .X

Virginca and Iris Setosa. Each class contains 50 samples. We implemented the DMOSL algorithm on the Iris data with $k = 1 - 8$. When $k = 1 - 4$, the DMOSL algorithm can detect the three classes correctly, with the classification accuracy over 96.65%. However, when $k = 5 - 8$, the DMOSL algorithm always leads to 4 or 5 Gaussians in which three major Gaussians can be located the actual classes approximately, while one or two abundant small Gaussians cannot be eliminated. The reason may be that the number of samples in the Iris data is not large enough and each class cannot match a Gauussian well so that some small Gaussians cannot be eliminated when k is much larger than $k^* = 3$.

4 Conclusions

We have investigated the automated model selection and the parameter estimation on Gaussian mixture modelling via a dynamic merge-or-split learning (DMOSL) algorithm. The DMOSL algorithm is constructed with a merge or split operation on the estimated Gaussians from the EM algorithm. It is demonstrated by the simulation experiments that the DMOSL algorithm can automatically determine the number of actual Gaussians in the sample data, also with a good estimation of the parameters in the original mixture. The DMOSL algorithm can be also successfully applied to the classification of Iris data.

References

1. G. Mc ac a , D. Pee , *Finite Mixture Models*, Ne Y : W e , 2000.
2. R. A. Re de a d H. F. Wa e , "M Xe de Xe , a e d a d Xe EM a g Xg ," *SIAM Review,* 26(2): 195-239, 1984.
3. A. K. Ja a d R. C. D be , *Algorithm for Clustering Data*, E g e d C ff , N. J.: P e Xce Ha , 1988.
4. J. A. Ha Xga , "D X b X b e c Xe g," *Classification and clustering*, J. Va R Ed ., : 45-72, Ne : Acade c P e , 1977.
5. H. A a e, "A Ne L aXXe SXaX Xca M de Ide XficaX ," *IEEE Trans. on Automatic Control,* AC-19: 716-723, 1974.
6. J. Ma, T. Wa g a d L. X, "A g ad e XBYY a ea g e Ga a Xe X a aXed de e ecX ," *Neurocomputing*, 56: 481-487, 2004.
7. J. Ma, B. Ga , Y. Wa g, a d Q. Ch g, "T f Xe g ad e XBYY ea g e f Ga a Xe X a aXed de e ecX ," *Lecture Notes in Computer Science*, 3177: 690-695, 2004.
8. J. Ma, T. Wa g, "E X e a ed a aXed de e ecX Ga a Xe," *International Journal of Pattern Recognition and Artificial Intelligence*, 18(8): 1501-1512, 2004.
9. N. Ueda, R. Na a , Y. Z. G ah a a , a d G. E. H X , "SMEM a g Xg f Xe de ," *Neural Computation*, 12(10): 2109-2128, 2000.
10. Z. Zha g, C. Che , J. S , a d K. L. Cha , "EM A g Xg f ea g Ga a Xe de Xa d e ge e aX ," *Pattern Recognition*, 36(9): 1973-1983, 2003.
11. N. V a a d A. L a , "A G eed EM A g Xg f Ga a M Xe Lea g," *Neural Processing Letters*, 15: 77-87, 2002.
12. M. S. S a Xa a, *Methods of Multivariate Statistics*, Ne Y : W e -I Xe c e ce, 2002.