A Componentwise Simulated Annealing EM Algorithm for Mixtures

Affan Pervez and Dongheui Lee

Technische Universität München

{affan.pervez, dhlee}@tum.de

Abstract. This paper addresses the problem of fitting finite Gaussian Mixture Model (GMM) with unknown number of components to the univariate and multivariate data. The typical method for fitting a GMM is Expectation Maximization (EM) in which many challenges are involved i.e. how to initialize the GMM, how to restrict the covariance matrix of a component from becoming singular and setting the number of components in advance. This paper presents a simulated annealing EM algorithm along with a systematic initialization procedure by using the principals of stochastic exploration. The experiments have demonstrated the robustness of our approach on different datasets.

1 Introduction

Finite mixture models provide a probabilistic tool for modeling arbitrarily complex probability density functions and have been used in a variety of different applications [9], [12,13]. The usual approach for fitting a GMM is EM, which provides the maximum likelihood (ML) parameter estimate of the model. K-means can be used for initializing EM. During EM the estimate may converge to the boundary of the parameter space (singular covariance matrix), causing likelihood to approach infinity. When this occurs, EM should be aborted and restarted with different initialization of the parameters. *Deterministic Annealing EM* (DAEM) algorithm [21] has been proposed for avoiding the estimate to converge at the boundary of the parameter estimate but it can get trapped at saddle points [21].

Broadly speaking there are three main approaches for estimating the number of components [7]: EM based methods [6], [16], Variational Bayesian methods [23] and stochastic methods by using Markov Chain Monte Carlo (MCMC) sampling [2], [19]. Variational Bayesian methods avoid overfitting but only provide an approximate solution while Stochastic methods are computationally very expensive. Moreover there are two types of EM based methods in which the number of components need not to be fixed in advance: Firstly *divisive* where the estimate starts from a single component which split into multiple components as the algorithm proceeds [3,4], [25] and secondly *agglomerative* where the estimate starts from a large number of components which are decreased as the algorithm proceeds [7,8]. A variety of ways have been proposed for spliting or merging GMM components [3], [14], [22], [24,25].

2 Overview of EM algorithm

A GMM with k components is parameterized by $\boldsymbol{\Theta}_{(k)} = \{\pi_m, \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m\}_{m=1}^{k}$ where $\pi_1, \ldots, \pi_k$ are mixing coefficients / priors with constraints $0 < \pi_m < 1$ and $\sum_{m=1}^{k} \pi_m = 1$, $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k$ are means and $\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_k$ are covariance matrices. The log-likelihood of a dataset $\mathbf{y}_{\mathsf{obs}} = \{\mathbf{y}^{(1)}, \cdots, \mathbf{y}^{(n)}\}$ for a GMM is defined as:

$$\mathcal{L}(\boldsymbol{\Theta}_{(k)}, \mathbf{y}_{\mathsf{obs}}) = \sum_{i=1}^{n} log \sum_{m=1}^{k} \pi_m \mathcal{N}(\mathbf{y}^{(i)}; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m) \tag{1}$$

The parameters are updated by cycling through the E-step and the M-step [6]. In the E-step, the expected value of complete data log-likelihood is calculated using the old parameters estimate, which is used for maximization in the M-step. It is a common practice to add a small regularization term $\lambda\mathbf{I}$ in all covariance matrics of the GMM at each update cycle. The EM is terminated when the increase in log-likelihood becomes very small i.e. $\frac{\mathcal{L}^{t+1}}{\mathcal{L}^{t}} - 1 < \epsilon$.

3 Stochastic exploration initialization approach

Many nature inspired algorithms exist for solving discrete optimization problems [11]. It is a well known fact that humans by nature are social animals. A person who is alone will tend to find other people and will stay in a community, if possible. Not only humans but animals also exhibit such behavior [10]. Given this, now consider a scenario where an individual is alone at a location. Seeking someone is important due to the above mentioned reasons. If there is no heuristic about the surrounding then with equal probability it will start to look in any direction. Suppose that there are many individuals like this at a place, what will happen when one comes in interaction with someone else? Most probably its search pace will decrease, since living in a group provides more stability. Now they also have a heuristic about the search direction.

Surprisingly it is not difficult to import the proposed approach for initializing a GMM. Now each observation will act like an individual so we will place a Gaussian at each observation. In the begining the covariance matrix is diagonal, with small equal positive value δ in the diagonal. The prior value of each component is $\frac{1}{n}$ at this stage where n is the total number of observations. After this the next step is exploration phase. The exploration is done by slightly expanding each component and then testing that whether it has sufficient overlap with one of its neighbouring Gaussians. The rate of expansion of each Gaussian is inversely propotional to its prior value. For expanding a Gaussian, we add exploration terms in the standard deviation (σ) of each eigenvector of its covariance matrix. The magnitude of each exploration term is proportional to its corresponding σ.

Once two Gaussians have sufficient overlap we can merge them. For detecting overlap between two components we define the coefficient $\mathbf{C}$. When two Gaussians are multiplied then the resulting density is again a Gaussian but multiplied with this coefficient [17] i.e. $\mathcal{N}(\boldsymbol{\mu}_Q, \boldsymbol{\Sigma}_Q) \times \mathcal{N}(\boldsymbol{\mu}_R, \boldsymbol{\Sigma}_R) = \mathbf{C}_{QR}\mathcal{N}(\boldsymbol{\mu}_S, \boldsymbol{\Sigma}_S)$ where $\mathbf{C}_{QR} = \mathcal{N}(\boldsymbol{\mu}_Q; \boldsymbol{\mu}_R, \boldsymbol{\Sigma}_Q + \boldsymbol{\Sigma}_R)$. The $\mathbf{C}$ has a property that as we expand the Gaussians via exploration, it keeps on increasing and then at a certain point it starts to decrease. As the **C** passes its peak value, which is detected as its value decreases in the next iteration, we merge the two Gaussians as in [25].

The exploration and merging cycles continue until the component count reaches a predifined value k_{max} where $k_{max} \gg k_{optimal}$. Since in the begining of the algorithm $k = n$, this can pose significant computation load. An elegant way to bound the computational load is to sample (without replacement) k_{start} observations from the dataset and use them for initialization where $k_{start} < n$. So now for the Stochastic Exploration initialization approach the algorithmic complexity will be $\mathcal{O}(k_{start}^2)$ instead of $\mathcal{O}(n^2)$.

4 Component-Wise Simulated Annealing EM for Mixtures

The proposed CSAEM^2 algorithm is the combination of Component-Wise EM for Mixtures (CEM^2) and simulated annealing algorithm. As mentioned by [5] that CEM^2 is a serial decomposition of optimization problem with a coordinatewise maximization. It goes through E-step and then applies M-step to update only one component at a time. For simulated annealing we have a temperature parameter τ which is gradually decreased to zero. The temperature value defines the probability of taking annealing step. The initial value of temperature is set to a small value as higher value of temperature can completly disturb the discovered GMM. During annealing iterations the responsibility (contribution of the component q, for generating the data point i) is calculated as:

$$\gamma_{q,i} = \frac{\left(\pi_q \mathcal{N}(y_i; \mu_q, \Sigma_q)\right)^{\Phi_q}}{\sum\limits_{l=1, l\neq q}^{k} \pi_l \mathcal{N}(y_i; \mu_l, \Sigma_l) + (\pi_q \mathcal{N}(y_i; \mu_q, \Sigma_q))^{\Phi_q}}$$

where $0 \leq \Phi_q \leq 1$. The amount of annealing Φ_q is inversely proportional to the weight of a component q and is calculated as $\Phi_q = \frac{\pi_q}{\max(\pi)}$. The annealing induces the fuzziness in the membership of the components with no annealing $(\max(\Phi) = 1)$ for the component with highest weight and highest annealing $(\min(\Phi))$ for the component with lowest weight.

When CSAEM^2 converges (after annealing iterations) then we generate all the possible $(k-1)$ component GMMs and switch to the one which yield highest likelihood value. CSAEM^2 and components merging are repeatedly applied until the components count k reaches k_{min} (usually one). The parameters $\hat{\boldsymbol{\Theta}}_{(k)}$ of all the models generated after applying CSAEM^2 are stored as candidate GMMs.

We remove a component m if $n\pi_m < \alpha(d+1)$ with $\alpha = 2$. As mentioned by [16], a component requires the support of at least $d+1$ observations for avoiding the covariance matrix to become singular i.e. $n\pi_m \geq (d+1)$. For a large value for k_{max}, many or all components can get removed simultaneously with general EM [8]. With the CEM^2, if a component dies, its weight is immediately distributed among other components, thus increasing the survival probability of the other components.

For model selection we have used *Mixture Minimum Discription Length* (MMDL) criterion [7]. It is similar to the *Bayesian Information Criterion* (BIC) [20] but it has an additional term for representing skewness. It is defined as:

$$\mathcal{C}_{\texttt{MMDL}}\left(\hat{\boldsymbol{\Theta}}_{(k)}, k\right) = -\mathcal{L}(\hat{\boldsymbol{\Theta}}_{(k)}, \mathbf{y}) + \frac{N(k)}{2}\log n + \frac{N(1)}{2}\sum_{m=1}^{k}\log(\pi_m)$$

where $N(k)$ is the number of free parameters in a k component GMM.

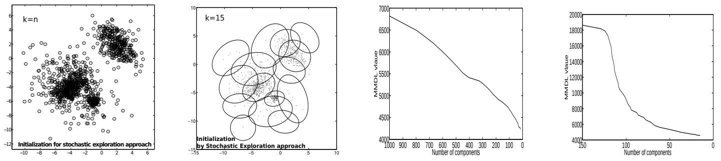
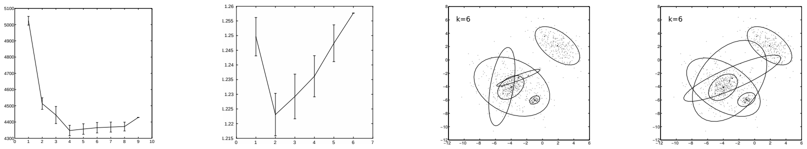


Fig. 1: Initialization results of sythetic datasets with stochastic exploration approach



(a) (b) (c) (d)

Fig. 2: (a) Result of 50 Monte Carlo simulations on synthetic Dataset 1 showing mean MMDL value (of 50 experiments) for components, with vertical bars depicting standard deviation in each value. (b) Same as (a) but for Dataset 2. GMM just before (c) and after (d) annealing step.

5 Experiments

We used following parameter values: $k_{start} = 150$ ($k_{start} = 250$ for Sythetic Dataset 2) , $k_{max} = 15, k_{min} = 1$, Annealing Iterations$= 400, \tau = 0.05, \epsilon = 10^{-5}$, $\lambda = max(10^{-4}, min(D > 0))$ where D is a $n \times n$ matrix with $D(Q,R) = \|\mathbf{y}^{(Q)} - \mathbf{y}^{(R)}\|^2$.

5.1 Results of Stochastic exploration initialization approach

Figure 1 shows the results of stochastic exploration approach on the synthetic dataset (section 5.2 : Dataset 1). The first column contains the highly overfitted $k = n$ component GMM while the second column contains the simplified $k = k_{max}$ component GMM obtained by stochastic exploration. The third column contains the MMDL value for the GMMs formed during transition from the $k = n$ component GMM to the $k = k_{max}$ component GMM. The initialization approach consistently decreases the MMDL value and thus improves the model representation while decreasing the number of components. The same behaviour can be observed in the last column which contains the MMDL value for the GMMs formed during transition from the $k = k_{start}$ component GMM to the $k = k_{max}$ component GMM.

5.2 Results of CSAEM^2

Experiment with Synthetic dataset
Dataset 1: 1000 samples were drawn from the four component bivariate GMM with

$$\pi_1 = \pi_2 = \pi_3 = 0.3, \pi_4 = 0.1, \boldsymbol{\mu}_1 = \boldsymbol{\mu}_2 = \begin{bmatrix} -4 \\ -4 \end{bmatrix} \boldsymbol{\mu}_3 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}, \boldsymbol{\mu}_4 = \begin{bmatrix} -1 \\ -6 \end{bmatrix}$$

$$\boldsymbol{\Sigma}_1 = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}, \boldsymbol{\Sigma}_2 = \begin{bmatrix} 6 & -2 \\ -2 & 6 \end{bmatrix}, \boldsymbol{\Sigma}_3 = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \boldsymbol{\Sigma}_4 = \begin{bmatrix} 0.125 & 0 \\ 0 & 0.125 \end{bmatrix}$$

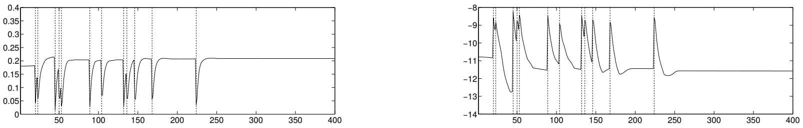
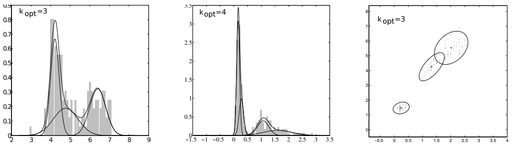


Fig. 3: The value of $\sum_{i=1}^{k} (\frac{1}{k} - \pi_i)^2$ (left) and $\sum_{i=1}^{k} \log(\pi_i)$ (right) during the annealing iterations of CSAEM^2. Vertical dashed lines depict the instances when annealing was applied.



(a) (b) (c)

Fig. 4: Results of CSAEM^2 on (a) Acidity, (b) Enzyme and (c) Iris datasets. In (a-b), data is encoded in histograms while (c) contains the projection of data on the two axis with highest variances

This GMM has also been used by [8], [24] and [14]. It provides a challenging scenario of overlaping components with two components having a common mean. We performed a Monte Carlo (MC) simulation of 50 experiments. The results are shown in Figure 2a. In all experiments our algorithm identified the right four component GMM.

Dataset 2: 800 samples were drawn from the two component 10 dimensional GMM with $\pi_1 = \pi_2 = 0.5, \boldsymbol{\mu}_1 = [0, \ldots, 0]^\top, \boldsymbol{\mu}_2 = [2, \ldots, 2]^\top, \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \mathbf{I}$. The GMM contain high dimensional fused components. Figure 2b contain the result of 50 MC experiments and every time our approach correctly identified the two component GMM.

Effect of annealing: The effect of annealing during CSAEM^2 can be observed in Figure 2(c-d). The annealing increases the coverage and thus the survival probability of the weak components while the components with high prior values are unaffected by annealing. Another interesting property of annealing is depicted in Figure 3. The maximum value of $f = \sum_{i=1}^{k} \log(\pi_i)$ is attained when $\pi_i = \frac{1}{k}$. Similarly the minimum value of $g = \sum_{i=1}^{k} (\frac{1}{k} - \pi_i)^2$ is attained when $\pi_i = \frac{1}{k}$. It can be observed in Figure 3 that there is a sharp decrease in the value of g and a sharp increase in the value of f after annealing cycles (Dataset 1). Thus annealing has a tendency of redistributing the values of π more equally.

Experiment with Real datasets

Now we consider Acidity and Enzyme datasets having skewed gaussians. They have been extensively studied by [18] and their optimal number of components are three and four respectively [18,15]. We performed a Monte Carlo simulation of 50 experiments and in all the experiments CSAEM^2 detected the same three and four component GMMs as shown in Figure 4(a-b). Then we also considered a well known relatively higher (four) dimensional Iris dataset [1]. The dataset contains three classes with 50 samples for each class. Again we performed a Monte Carlo simulation of 50 experiments and with hundred percent success rate detected the three component GMM as shown in Figure 4(c).

Comparison with similar EM approaches: [7,8] has presented similar EM based approaches where the components count starts from k_{max} and are brought down to k_{min}.

Table 1: Percentage frequency of choosing k clusters for 50 experiments by our approach and the approaches presented by Figueiredo et al. (1999,2002).

k	Our method	Figueiredo et al. (2002)	Figueiredo et al. (1999)
3	0	0	1
4	47	1	17
5	3	13	16
6	0	14	7
7	0	10	4
8	0	8	2
9	0	2	2
10	0	2	1

Table 2: Percentage frequency of choosing $k_{opt} = 4$ clusters for 100 experiments by our approach with random initialization and stochastic exploration initialization.

k_{max}	Random initialization	Our initialization
6	62	88
7	84	93
8	91	95
9	93	97

These algorithms explicitly target the components with low prior values for switching to a $(k-1)$ component GMM and thus often fail when there is a component with very low prior value. Our approach increases the survival probability of the weak components and hence overcomes this problem. Now we performed a Monte Carlo simulation of 50 experiments. 710 samples were drawn from the four component bivariate GMM:

$$\pi_1 = \frac{10}{71}, \pi_2 = \frac{20}{71}, \pi_3 = \frac{40}{71}, \pi_4 = \frac{1}{71}, \boldsymbol{\mu}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \boldsymbol{\mu}_2 = \begin{bmatrix} 3 \\ 4 \end{bmatrix}, \boldsymbol{\mu}_3 = \begin{bmatrix} 3 \\ -3 \end{bmatrix}$$

$$\boldsymbol{\mu}_4 = \begin{bmatrix} 8 \\ 8 \end{bmatrix}, \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_4 = \mathbf{I}, \boldsymbol{\Sigma}_3 = \begin{bmatrix} 2 & 1.9 \\ 1.9 & 2 \end{bmatrix}, \boldsymbol{\Sigma}_2 = \begin{bmatrix} 2 & -1.5 \\ -1.5 & 2 \end{bmatrix}$$

All algorithms used same $k_{max}, k_{min}, \epsilon$ and λ values. Although the components are quite well separated from each other, the approaches presented in [7,8] performed very poorly while our algorithm outperformed these methods as can be seen in Table 1.

Robustness against $k_{\mathbf{max}}$: Choosing $k_{max} \gg k_{optimal}$ provides robustness against initialization issues but it can be underestimated. Now we test the performance of our approach with relatively smaller values of k_{max} for the simulated dataset with a weak component. For comparison, the initial k_{max} component GMM is obtained with two methods: the random initialization procedure presented in Figueiredo et al. (2002) and our initialization procedure. The results are summarized in Table 2. We can see that our initialization has high frequency of detecting right number of components, even when starting with relatively smaller k_{max}.

6 Conclusion

We have proposed a novel nature inspired initialization approach for fitting a GMM. It utilizes search strategy where each component looks for its nearby component. Two components are merged when they have high overlap. CSAEM^2 is applied when the components count reaches k_{max}. A component is annihilated if it becomes too weak. $(k-1)$ components GMM is obtained by selecting the one which yields highest likelihood value. MMDL criterion is used to select the optimal model complexity. Our approach has shown promising results on challenging simulated and real datasets.

References

1. Anderson, E.: The irises of the gaspe peninsula. Bulletin of the American Iris society 59, 2–5 (1935)
2. Bensmail, H., Celeux, G., Raftery, A.E., Robert, C.P.: Inference in model-based cluster analysis. Statistics and Computing 7(1), 1–10 (1997)
3. Blekas, K., Lagaris, I.E.: Split–merge incremental learning (smile) of mixture models. In: Artificial Neural Networks–ICANN 2007, pp. 291–300. Springer (2007)
4. Calinon, S., Pervez, A., Caldwell, D.G.: Multi-optima exploration with adaptive gaussian mixture model. In: IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL). pp. 1–6 (2012)
5. Celeux, G., Chrétien, S., Forbes, F., Mkhadri, A.: A component-wise em algorithm for mixtures. Journal of Computational and Graphical Statistics (2012)
6. Dempster, A.P., Laird, N.M., Rubin, D.B., et al.: Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal statistical Society 39(1), 1–38 (1977)
7. Figueiredo, M.A., Leitão, J.M., Jain, A.K.: On fitting mixture models. In: Energy minimization methods in computer vision and pattern recognition. pp. 54–69. Springer (1999)
8. Figueiredo, M.A., Jain, A.K.: Unsupervised learning of finite mixture models. Pattern Analysis and Machine Intelligence, IEEE Transactions on 24(3), 381–396 (2002)
9. Koo, S., Lee, D., Kwon, D.S.: Incremental object learning and robust tracking of multiple objects from rgb-d point set data. Journal of Visual Communication and Image Representation 25(1), 108–121 (2014)
10. Krause, J., Ruxton, G.D.: Living in groups. Oxford University Press (2002)
11. Krause, J., Cordeiro, J., Parpinelli, R.S., Lopes, H.S.: A survey of swarm algorithms applied to discrete optimization problems. Swarm Intelligence and Bio-inspired Computation: Theory and Applications. Elsevier Science & Technology Books pp. 169–191 (2013)
12. Lee, D., Nakamura, Y.: Mimesis model from partial observations for a humanoid robot. The International Journal of Robotics Research 29(1), 60–80 (2010)
13. Lee, D., Ott, C., Nakamura, Y.: Mimetic communication with impedance control for physical human-robot interaction. In: Robotics and Automation, 2009. ICRA'09. IEEE International Conference on. pp. 1535–1542. IEEE (2009)
14. Luo, B., Wei, S., et al.: Estimation for the number of components in a mixture model using stepwise split-and-merge em algorithm. Pattern Recognition Letters 25(16), 1799–1809 (2004)
15. McGrory, C.A., Titterington, D.: Variational approximations in bayesian model selection for finite mixture distributions. Computational Statistics & Data Analysis 51(11), 5352–5367 (2007)
16. McLachlan, G., Peel, D.: Finite mixture models. John Wiley & Sons (2004)
17. Petersen, K.B., Pedersen, M.S.: The matrix cookbook. Technical University of Denmark pp. 7–15 (2008)
18. Richardson, S., Green, P.J.: On bayesian analysis of mixtures with an unknown number of components (with discussion). Journal of the Royal Statistical Society: series B (statistical methodology) 59(4), 731–792 (1997)
19. Roeder, K., Wasserman, L.: Practical bayesian density estimation using mixtures of normals. Journal of the American Statistical Association 92(439), 894–902 (1997)
20. Schwarz, G., et al.: Estimating the dimension of a model. The annals of statistics 6(2), 461–464 (1978)
21. Ueda, N., Nakano, R.: Deterministic annealing em algorithm. Neural Networks 11(2), 271–282 (1998)

22. Ueda, N., Nakano, R., Ghahramani, Z., Hinton, G.E.: Smem algorithm for mixture models. Neural computation 12(9), 2109–2128 (2000)
23. Valente, F., Wellekens, C., et al.: Variational bayesian gmm for speech recognition.
24. Zhang, B., Zhang, C., Yi, X.: Competitive em algorithm for finite mixture models. Pattern recognition 37(1), 131–144 (2004)
25. Zhang, Z., Chen, C., Sun, J., Luk Chan, K.: Em algorithms for gaussian mixtures with split-and-merge operation. Pattern Recognition 36(9), 1973–1983 (2003)