A DETERMINISTIC, ANNEALING-BASED APPROACH FOR LEARNING AND MODEL SELECTION IN FINITE MIXTURE MODELS

Qi Zhao and David J. Miller

Department of Electrical Engineering
The Pennsylvania State University
e-mail:qzz100@psu.edu,millerdj@ee.psu.edu

ABSTRACT

We address the longstanding problem of learning and model selection in finite mixtures. A common approach is to generate solutions of varying number of components (via the Expectation-Maximization (EM) algorithm) and then select the best model in the sense of a cost such as the Bayesian Information Criterion (BIC). A recent alternative uses component-wise EM (CEM) and, further, integrates model selection within CEM. Both approaches are susceptible to finding poor solutions, the first due to initialization sensitivity of EM and the second due to the sequential (greedy) nature of CEM. Deterministic annealing for clustering (DA) and mixture modeling (DAEM) provide potential for avoiding local optima. However, these methods do not encompass model selection. We propose a new technique with positive attributes of all these methods: it integrates learning and model selection, performs batch optimization over components, and has the character of DA, with the optimization performed over a sequence of decreasing temperatures. Unlike standard DA, with the partition entropy reduced as the temperature is lowered, our approach reduces entropy of binary random variables that express whether each component is *active* or *inactive*. At low temperature, the method achieves explicit model order selection. Experiments demonstrate favorable performance of our method, compared with several alternatives. We also give an interesting stochastic generative model interpretation for our method.

1. INTRODUCTION

Learning and model selection in finite mixture models and unsupervised clustering with estimation of the number of clusters are longstanding problems, albeit ones of great continuing interest, in the statistics and pattern recognition communities. There is a vast literature on clustering and cluster validity procedures for choosing between competing solutions. A number of techniques have also been proposed for the related problem of choosing the number of components (the model order) in finite mixture models, e.g.,[1],[2]. There are several aspects which contribute to the difficulty of learning and model order selection in mixtures. First, there is the choice of the criterion function (henceforth referred to as the *model cost*) by which one evaluates the quality of solutions and hence, by which one chooses the best model. A variety of model costs have been proposed, including Akaike's Information Criterion, Minimum Message Length (MML), Bayesian Information Criterion (BIC), cross validation, as well as other measures. To date, there is no full consensus on the proper cost [2].

Second, there is the learning procedure for estimating the parameters of the model. Learning is typically performed via the Expectation-Maximization (EM) algorithm whose virtue – guaranteed monotonic ascent in the likelihood function – also means that the method only finds *local* optima of this function. The solution quality is thus (in some cases quite) sensitive to the parameter initialization. If the model is well-learned, but at the wrong model order, and poorly learned at the true order, the best model in the sense of the model cost may very well be a solution at the wrong order. Suboptimal learning can thus greatly contribute to errors in model selection. In order to overcome problems of local optima, for *each* model order one may need to re-run EM numerous times based on random parameter initialization and then select the solution with greatest likelihood. The saved solutions at different orders can then be compared in terms of the model cost. However, this procedure may be quite computationally demanding.

Finally, there is the question of whether one should perform learning *followed* by model selection or, alternatively, *integrated* learning *and* model selection, e.g. [1]. In the latter case, the parameters and the model order are *jointly* chosen to minimize the model cost. Integrated learning and model selection can somewhat mitigate the local optima sensitivity of EM [1],[3]. For example, in [3], it was found that better solutions at a given order are generally obtained if component annihilations (those consistent with decreases in the model cost) are embedded within EM iterations. Essentially, starting from a large model order, integrated learning and component pruning can remove poorly initialized components, thus mitigating suboptimal initializations. While the approach in [1] does integrate learning and model selection, the learning algorithm is of necessity (based on the particular choice of model cost) not the batch EM algorithm, but rather the component-wise EM (CEM) algorithm. This is a sequential (greedy) algorithm which optimizes over one component at each step. Greedy optimization techniques generally increase sensitivity to initialization and to the order in which the parameters are visited. Another issue with [1] is the choice of model cost. [1] minimizes the following criterion based on MML:

$$\mathcal{L}(\theta, \mathcal{Y}) = \frac{N}{2} \sum_{m:\alpha_m > 0} \log \frac{M\alpha_m}{12} + \frac{k_{nz}}{2} \log \frac{M}{12}$$

THIS WORK WAS SUPPORTED BY NATIONAL SCIENCE FOUNDATION GRANT NSF IIS-0082214.

$$+ \quad \frac{k_{nz}(N+1)}{2} - \log p(\mathcal{Y}|\theta)$$

where N is the number of parameters specifying each component, M the number of samples in the data set $\mathcal{Y}$, k_{nz} the number of components with non-zero mass, and α_m the mass for component m. Experimentally, we have found that this tends to overestimate the number of components, due to the term $\log(M\frac{\alpha_m}{12})$, which contributes negatively to the cost if the mass is sufficiently small. Thus, this cost favors the existence of some components with small mass. On the other hand, BIC tends to *underestimate* the order. This issue is discussed further in our experimental results.

While EM and CEM are both susceptible to finding poor solutions, deterministic annealing (DA) techniques aim to avoid local optima when learning clustering [4] and mixture model [5] solutions. However, the DA framework does not include model selection – for the method in [4], the number of clusters will continue to grow as the temperature is lowered unless a hard ceiling is imposed on the cluster number. Likewise, at the limiting temperature, the cost minimized by [5] is the negative log likelihood – this can always be further decreased by adding new mixture components.

Here, we introduce a new method that incorporates positive attributes from each of the previous methods. Our method integrates learning and model selection, performs batch optimization over the components, and has the character of deterministic annealing, with the optimization performed over a sequence of decreasing "temperatures". We do not propose a new model cost. Our approach can be implemented for a number of existing costs. In this work, we have used BIC. Our approach differs from existing DA methods in its ability to directly estimate the model order, as well as in several other respects elaborated in the sequel. In section 2, we develop our new method. In section 3, we provide experimental comparisons. The paper concludes with future work.

2. FORMULATION

Consider an (at most) K-component mixture density function,

$$g(\underline{x}) = \sum_{j=1}^{K} v_j \alpha_j f_j(\underline{x}|\theta_j) \tag{1}$$

with $f_j(\cdot)$ a component density specified by a parameter set θ_j, $0 \leq \alpha_j \leq 1$ the component's mass, $v_j \in \{0,1\}$, and with the constraint $\sum_{j=1}^{K} v_j\alpha_j = 1$. The $\{v_j\}$ indicate which components participate in the mixture. Denote the full parameter set by $\Theta = \{\{\theta_j\},\{\alpha_j\},\{v_j\}\}$. Suppose we wish to choose Θ to minimize the BIC model cost for a given data set $\{\underline{x_i}, i = 1,\ldots,M\}$:

$$\mathrm{BIC}(\Theta) = \sum_{j=1}^{K} v_j \frac{N_j}{2}\log M - \sum_{i=1}^{M}\log\sum_{j=1}^{K} v_j\alpha_j f_j(\underline{x_i}|\theta_j), \tag{2}$$

constrained by $\sum_{j=1}^{K} v_j\alpha_j = 1$. Here, N_j is the number of free parameters for component j. [1]

There are several barriers to minimizing (2):

1. sensitivity to parameter initialization and, thus, susceptibility to finding poor local optima, as previously discussed.
2. The coupling of the $\{v_j\}$ through the constraint.

This latter difficulty precludes pure batch optimization, with the $\{v_j\}$ updated in parallel, independent of one another, since this approach cannot ensure that $v_j = 0, j = 1,\ldots,K$ does not occur, i.e., that all components do not "drop out". One valid optimization strategy, proposed in [3] in a different learning context, involves an alternating minimization, with an EM step for optimizing $\{\{\theta_j\},\{\alpha_j\}\}$ given fixed $\{v_j\}$, and with a sequential cyclic step for optimizing the $\{v_j\}$ given fixed $\{\{\theta_j\},\{\alpha_j\}\}$. While this approach was found to be effective in [3], it is sensitive to initialization and thus to finding poor local optima.

Alternatively, we next propose a DA-based approach. Accordingly, we now let $V_j \in \{0,1\}$ be a random variable, with associated probabilities $P_j \equiv \mathrm{Prob}[V_j = 1]$, indicating the probability that component j is active. Suppose we directly write down a probabilistic generalization of (2), i.e.,

$$\widetilde{\mathrm{BIC}}(\tilde{\Theta}) = \sum_{j=1}^{K} P_j \frac{N_j}{2}\log M - \sum_{i=1}^{M}\log\sum_{j=1}^{K} P_j\alpha_j f_j(\underline{x_i}|\theta_j), \tag{3}$$

where $\tilde{\Theta} = \{\{\theta_j\},\{\alpha_j\},\{P_j\}\}$. We will not impose the constraint $\sum_{j=1}^{K} P_j\alpha_j = 1$. Instead, we constrain $P_j \in [0,1]$ and $\sum_{j=1}^{K}\alpha_j = 1$. This will be justified shortly. Let us check the plausibility of (3). The term $\sum_{j=1}^{K} P_j\frac{N_j}{2}\log M$ makes sense – this is the expected model penalty, based on the pmfs $\{(P_j, 1-P_j)\}$. [2] What about the term $\sum_{i=1}^{M}\log\sum_{j=1}^{K} P_j\alpha_j f_j(\underline{x_i}|\theta_j)$? This has the *form* of a log-likelihood function, with mass $P_j\alpha_j$ for component j. Unfortunately, with $P_j \in [0,1]$ and $\sum_{j=1}^{K}\alpha_j = 1$, we have $\sum_{j=1}^{K} P_j\alpha_j \leq 1$, i.e., the masses do not necessarily sum to one. In fact, equality only occurs in the special case where $P_j = 1$ for all j such that $\alpha_j > 0$. Accordingly, in general, it does *not* appear that $\sum_{i=1}^{M}\log\sum_{j=1}^{K} P_j\alpha_j f_j(\underline{x_i}|\theta_j)$ is a log-likelihood function. However, suppose we postulate one additional component density $f_0(\underline{x}|\theta_0)$ with θ_0 chosen in the following special way: θ_0 is such that $f_0(\underline{x_i}|\theta_0) \leq \epsilon$, $i = 1,\ldots,M$, with ϵ a *very* small value. This is easily achieved in principle, e.g., in the Gaussian density case if the mean μ_0 is positioned far outside of the convex hull of the data, and with the variance $\sigma_0^2(l)$ in each direction (l) chosen to be small. Clearly,

$$L = \sum_{i=1}^{M}\log(\sum_{j=1}^{K} P_j\alpha_j f_j(\underline{x_i}|\theta_j) + (1 - \sum_{j=1}^{K} P_j\alpha_j) f_0(\underline{x_i}|\theta_0)) \tag{4}$$

is a valid log-likelihood. Moreover, with θ_0 chosen to satisfy the ϵ constraints, and with ϵ made arbitrarily small, we have that

$$L \equiv L(\tilde{\Theta}) = \sum_{i=1}^{M}\log\sum_{j=1}^{K} P_j\alpha_j f_j(\underline{x_i}|\theta_j). \tag{5}$$

I.e., the term in (3) *is* a log-likelihood if the mass fraction $(1 - \sum_{j=1}^{K} P_j\alpha_j)$ is reserved for additional components with negligible likelihood of producing the $\{\underline{x_i}, i = 1,\ldots,M\}$. The log-likelihood function (4) (essentially (5)) is consistent with the following stochastic generation of the data.

[1] In writing (2) it is assumed that each datum is generated independently according to the mixture.

[2] With probability P_j, component j is active, and the cost of its parameters is incurred in this case.

For each sample:

1. Randomly select a component according to the mass distribution $\{P_j\alpha_j, j = 1, \ldots, K, (1 - \sum_{j=1}^{K} P_j\alpha_j)\}$.
2. If component $m \in \{1, \ldots, K\}$ is chosen, generate $\underline{x}$ according to $f_m(\underline{x}|\theta_m)$; else, generate $\underline{x}$ according to $f_0(\underline{x}|\theta_0)$.

Clearly if $\sum_j P_j\alpha_j < 1$ and $f_0(\underline{x_i}|\theta_0) < \epsilon, \forall i$, with ϵ arbitrarily small, then as M grows the model becomes a highly improbable generator for the data set ! Accordingly, if we would like this model to be a *good* explanation for the data, we need to choose $\sum_j P_j\alpha_j = 1$. Let us consider an optimization procedure which *ultimately* satisfies this constraint and, moreover, ultimately also satisfies $P_j \in \{0, 1\}$, i.e., $P_j = v_j$. Relaxing these constraints at the outset will allow us to define a *deterministic annealing* procedure useful for avoiding local optima of the cost $\mathrm{BIC}(\theta)$.

We first define the sum of entropies:

$$H = -\sum_j (P_j \log P_j + (1 - P_j) \log(1 - P_j)), \qquad (6)$$

which measures the level of average uncertainty in whether components are active or inactive. We can then pose the constrained minimization problem:

$$\min_{\tilde{\Theta}=\{\{\theta_j\},\{\alpha_j\},\{P_j\}\}} \widetilde{\mathrm{BIC}}(\tilde{\Theta}) \quad \text{subject to } H = H_0. \qquad (7)$$

The associated unconstrained Lagrangian objective function is $F = \widetilde{\mathrm{BIC}}(\tilde{\Theta}) - TH$, with T a Lagrangian multiplier. At a given T, the problem is thus $\min_{\tilde{\Theta}} F$. At high T, we maximize H, yielding $P_j = 1/2, \forall j$ and a set of remaining parameters

$$\begin{aligned} \{\{\theta_j\}, \{\alpha_j\}\} &= \arg \min_{\{\alpha'_j\},\{\theta'_j\}} \widetilde{\mathrm{BIC}}(\{\alpha'_j\}, \{\theta'_j\}, \{P_j\})|_{\{P_j=\frac{1}{2}\}} \\ &= \arg \max_{\{\alpha'_j\},\{\theta'_j\}} L(\{\alpha'_j\}, \{\theta'_j\}, \{P_j\})|_{\{P_j=\frac{1}{2}\}} \end{aligned} \qquad (8)$$

As T (interpretable as 'temperature') is lowered, the constraint on high entropy is relaxed. The model penalty term $\sum_{j=1}^{K} P_j \frac{N_j}{2} \log T$ thus begins to impart an influence, forcing the pmfs $\{P_j, 1 - P_j\}$ to skew either towards more probable "activity" or "inactivity" as the components still seek to best fit the data. Ultimately, at zero T, we are directly minimizing $\widetilde{\mathrm{BIC}}(\tilde{\Theta})$, generally achieved by $P_j \to v_j \ \forall j$. Moreover, it will be seen (shortly) from the associated re-estimation equations that as $P_l \to 0$ for some l, we also have that $\alpha_l \to 0$. Thus, with $\sum_{j=1}^{K} \alpha_j = 1$ always enforced, we have that $1 - \sum_{j=1}^{K} \alpha_j P_j \to 0$, i.e. our desired constraint $\sum_{j=1}^{K} \alpha_j P_j = 1$ is satisfied at zero T. Essentially, as T is lowered, there is greater and greater impetus to explain the data while also accounting for the penalty. This necessitates "stealing" more and more mass from (the deficient) component θ_0. At $T = 0$, all mass concentrates on "active" components.

We thus suggest a procedure akin to a *deterministic annealing* algorithm, optimizing F starting from high T, and tracking the solution through a sequence of decreasing temperatures. This method differs from standard DA in several important respects:
1) In DA, it is the *partition* entropy (associated with probabilistic assignments of points to clusters) that is lowered, whereas we lower the entropy of binary random variables indicating whether components are *active* or *inactive*.
2) In standard DA, the number of components *grows* via a sequence of phase transitions and the solution at high T is independent of initialization. In our approach, we start with *large* K (larger than the 'true' model size) and at high T the solution is a (locally optimal) maximium likelihood estimate based on these K components[3]. Accordingly, at high T there *is* dependency on initialization. As T is lowered, however, poorly chosen initial components are gradually removed. If K is made sufficiently large, then the annealing process conducts a search over a rich set of candidate components and the solution at low T should be largely insensitive to the initialization.
3) Standard DA [4],[5] does not provide a way to do model selection, whereas our annealing method automatically achieves model selection at low T.

Minimization at a given temperature:

With the $\{P_j\}$ fixed, the remaining parameters can be minimized directly via the EM algorithm. For example, if $f_j(\cdot)$ are multivariate Gaussian densities, we have the (almost) familiar EM equations for masses, means, and covariance matrices:

$$\alpha_j^{(t+1)} = \frac{1}{M} \sum_i P(j|\underline{x_i})^{(t)} \qquad (9)$$

$$\underline{m_j}^{(t+1)} = \frac{\sum_i \underline{x_i} P(j|\underline{x_i})^{(t)}}{\sum_i P(j|\underline{x_i})^{(t)}} \qquad (10)$$

$$\Sigma_j^{(t+1)} = \frac{\sum_i (\underline{x_i} - \underline{m_j}^{(t+1)})(\underline{x_i} - \underline{m_j}^{(t+1)})^{\mathrm{T}} P(j|\underline{x_i})^{(t)}}{\sum_i P(j|\underline{x_i})^{(t)}} \qquad (11)$$

where

$$P(j|\underline{x})^{(t)} = \frac{P_j^{(t)} \alpha_j^{(t)} f_j(\underline{x}|\theta_j^{(t)})}{\sum_k P_k^{(t)} \alpha_k^{(t)} f_k(\underline{x}|\theta_k^{(t)})} \qquad (12)$$

Note that from (9) and (12), it is clear that $\alpha_j \to 0$ as $P_j \to 0$.

One can also write down fixed point iterations(FPIs) for the $\{P_j\}$ parameters. These equations can be obtained starting from the necessary optimality conditions $\frac{\partial F}{\partial P_j} = 0 \ \forall j$. However, these FPIs, although *typically* well-behaved, are not guaranteed to descend in F. Even so, we have used these FPIs to good effect in our simulations. Alternatively, P_j can be parameterized using a softmax function, i.e. $P_j = e^{\lambda_j}/(1 + e^{\lambda_j})$, λ_j real, with the $\{\lambda_k\}$ chosen to minimize F via gradient descent. Thus, at each T, we perform 1) EM and 2) gradient descent on $\{\lambda_j\}$, alternately, until a convergence criterion is met. Then T is lowered.

3. EXPERIMENTAL RESULTS

We have compared our approach, denoted DAMS (deterministic annealing-based model selection), with two other methods. One is the standard method where EM is run for different model orders, with the solution with the lowest model cost saved. We denote this method by EMS (exhaustive model selection) and use BIC as the model cost. The other method, from [1], is denoted CEM. We used 2-dimensional synthetic data sets for evaluation, each with 900 points. One type has 3 Gaussian components. Each element in the means of these components is randomly generated by a Gaussian

[3]With $P_j = \frac{1}{2} \forall j$ at high T, the objective reduces to standard maximum likelihood, as indicated in (8).

distribution $N(0,1)$. The covariance matrix is $C = A^T A$, where each element in A is also randomly generated by a Gaussian distribution $N(0,1)$. We constrain $|C| > 0.001$. The mass probabilities of these three components are randomly generated based on a uniform distribution. The other type has 6 Gaussian components, with each mean element generated by $N(0,4)$, and $0.01 < |C| < 0.1$. The second type of data set has a clearer cluster structure. We tried 90 data sets of the first kind and 20 data sets of the second. Tables 1 and 2 display the results. If the estimated model order is smaller than the true one, the fraction of points in error ('average error') is calculated by mapping each estimated component (and all the points it owns in a MAP sense) to the closest true component; else the error is calculated based on a mapping of each true component to the closest estimated component. In our method, we chose the initial temperature as 100, the temperature scaling factor as 0.7, and the final temperature as 0.1 because the "active" probabilities have already converged at this temperature.

EMS vs. DAMS: With many random initializations for varying the number of components, EMS can work very well. However, the computation is prohibitive. For roughly fair computational comparison, we restrict the number of initializations to one for each model order. We exhaustively explored the orders from 1 to 8 for the data sets with 3 components, and from 3 to 12 for the data sets with 6 components. This still needs $\sim$ twice the computation time of our method. From Table 1, we can see EMS performs well except that it has a larger average BIC model cost compared with DAMS. However, when the number of components is 6, EMS with one initialization finds more local optima and thus shows a poorer performance in Table 2.

CEM vs. DAMS: Here we used source code for CEM provided by Figueiredo[1]. First, we tried the failure case mentioned in [1], where one mass is much smaller than the other three. Unlike CEM, our method finds the true solution. Next, for the first synthetic data set type, both methods started from 15 initial components. For the second type, DAMS started from 20 initial components, while CEM started from 30 initial components. CEM is still much faster than DAMS (roughly 10 times). From Tables 1 and 2, our method estimates the model order better than CEM. CEM tends to overestimate the model order, mentioned in the introduction, while our method tends to underestimate the order. This is attributable to use of the BIC criterion. Interestingly CEM has lower average error ratio than our method in Table 1. The reason is that for CEM in the overestimated case, the extra components often have small masses. These components do not affect the error ratio much. On the other hand, for DAMS in the underestimated case, the error ratio increases significantly because some whole components may be in error. The situation is different in Table 2, because each component has a relatively small mass. Thus, the loss of some whole components does not affect the error ratio greatly. Accordingly, the error ratio is lower for DAMS than CEM in Table 2.

Table 1. Results for data sets with 3 components.

	EMS	DAMS	CEM
1	2	5	0
2	22	28	9
3(true)	60	57	42
4	5	0	14
5	1	0	9
6	0	0	9
7	0	0	3
8	0	0	1
9	0	0	1
9	0	0	1
average error	0.1807	0.1888	0.1866
BIC	2673.1	2641.9	

Table 2. Results for data sets with 6 components.

	EMS	DAMS	CEM
4	1	2	0
5	4	6	3
6(true)	8	12	11
7	7	0	1
8	0	0	1
9	0	0	0
10	0	0	1
11	0	0	0
12	0	0	3
average error	0.1469	0.1377	0.1663
BIC	2640.3	2633.7	

4. FUTURE WORK

While we have addressed integrated learning and model selection here, more generally, our learning strategy provides an effective way to optimize over a mixture whose components may belong to one of several types or "flavors". Here, we considered 'active' and 'inactive' flavors. Alternatively, as in [3], we could use our approach to optimize components that generate data from either *known* classes or *unknown* classes. This is useful in the context of discovering new classes in mixed labeled/unlabeled data sets [3]. Finally, our approach could be used to optimize a *generalized* mixture model [6], where each component may be drawn from one of *several* candidate parametric density families. These applications will be investigated in future work.

References

[1] Mario A.T. Figueiredo and Anil K. Jain, "Unsupervised Learning of Finite Mixture Models," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, no. 3, pp. 381–396, Mar. 2002.

[2] G. Mclachlan and D. Peel, *Finite Mixture Models*, John Wiley and Sons, New York, 2000.

[3] D.J. Miller and J. Browning, "A mixture model and EM-based algorithm for class discovery, robust classification, and outler rejection in mixed labeled/unlabeled data sets," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 25, no. 11, pp. 1468–1483, Nov. 2003.

[4] K. Rose, E. Gurewitz, and G.C. Fox, "Vector Quantization by Deterministic Annealing," *IEEE Trans. Inform. Theory*, vol. 38, no. 4, pp. 1249–1257, July 1992.

[5] Naonori Ueda and Ryohei Nakano, "Deterministic Annealing EM Algorithm," *Neural Networks*, vol. 11, pp. 271–282, 1998.

[6] Yves Delignon, Abdelwaheb Marzouki, and Wojciech Pieczynski, "Estimation of Generalized Mixtures and Its Application in Image Segmentation," *IEEE Trans. Image Processing*, vol. 6, no. 10, pp. 1364–1375, Oct. 1997.