Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science

6.S064 Introduction to Machine Learning

Phase 3: mixture models, model selection (lecture 13)

Mixture models cont'd

We have previously introduced mixture models, and how to estimate them from *incomplete data* $\{x^{(1)}, \ldots, x^{(n)}\}$, i.e., data without cluster labels. In particular, we saw how to estimate a mixture of Gaussians model

$$P(x|\theta) = \sum_{i=1}^{k} p_i P(x|\mu^{(i)}, \sigma_i^2) \tag{1}$$

where each mixture component $P(x|\mu^{(i)}, \sigma_i^2)$ is a spherical Gaussian distribution

$$P(x|\mu^{(i)}, \sigma_i^2) = \frac{1}{(2\pi\sigma_i^2)^{d/2}} \exp(-\frac{1}{2\sigma_i^2}\|x - \mu^{(i)}\|^2) \tag{2}$$

The parameters θ of the mixture model include the mixing proportions $p_1, \ldots, p_k$, the means $\mu^{(1)}, \ldots, \mu^{(k)}$, and the variances $\sigma_1^2, \ldots, \sigma_k^2$. Our estimation criterion was to find the parameters that maximize the log-likelihood of $D = \{x^{(1)}, \ldots, x^{(n)}\}$, i.e., maximize

$$l(D;\theta) = \sum_{t=1}^{n} \log P(x^{(t)}|\theta) = \sum_{t=1}^{n} \log \left[\sum_{i=1}^{k} p_i P(x^{(t)}|\mu^{(i)}, \sigma_i^2) \right] \tag{3}$$

Note that, in the absence of cluster/component labels, we must sum over all the alternative ways of generating each data point $x^{(t)}$. This summation ties all the parameters θ together in the context of each data point, and results in a challenging (non-convex) optimization problem. However, we can still use the EM algorithm to find a locally optimal solution.

The EM-algorithm alternates between two steps: 1) softly assigning points to components (completing the data), and 2) optimizing each component separately from the completed data (as if the completed data were given a priori). Specifically,

- **E-step:** softly assign points to clusters based on posterior probabilities

$$p(i|t) = \frac{p_i P(x^{(t)}|\mu^{(i)}, \sigma_i^2)}{\sum_{j=1}^{k} p_j P(x^{(t)}|\mu^{(j)}, \sigma_j^2)}, \quad i = 1, \ldots, k, \quad t = 1, \ldots, n \tag{4}$$

- **M-step:** Given "weights" $p(i|t)$, estimate each Gaussian separately from their "weighted data"

$$\hat{n}_i = \sum_{t=1}^{n} p(i|t), \quad \hat{p}_i = \frac{\hat{n}_i}{n}, \quad \hat{\mu}^{(i)} = \frac{1}{\hat{n}_i} \sum_{t=1}^{n} p(i|t) x^{(t)}, \tag{5}$$

$$\hat{\sigma}_i^2 = \frac{1}{d\hat{n}_i} \sum_{t=1}^{n} p(i|t) \|x^{(t)} - \hat{\mu}^{(i)}\|^2 \tag{6}$$

The EM-steps are repeated until convergence (little change in the parameters and/or log-likelihood).

How we initialize the mixture parameters matters a great deal. For example, suppose we set $p_i = 1/k$, $\mu_i = \mu$, $\sigma_i^2 = \sigma^2$, $i = 1, \ldots, k$, for some μ and σ^2. In other words, we initialize all the Gaussians to be the same. How will the EM-algorithm proceed in this case? In the E-step, we softly assign points to Gaussians based on their probabilities of generating the points. But, given our initialization, there's no difference between the component Gaussians. As a result, $p(i|t) = 1/k$, $i = 1, \ldots, k$, for all the points. In the subsequent M-step, all the Gaussians see the same weighted data, resulting again in the same (but updated) Gaussians. The whole mixture in this case acts like a single Gaussian distribution. It is important that the initialization produces clearly distinct components.

Another failure mode of the mixture estimation process (as we have defined it so far) is that one (or more) of the component Gaussians could end up wrapping themselves around a single point. In other words, we could set $\mu^{(i)} = x^{(t)}$ for some t, and reduce the corresponding variance σ_i^2 to zero. Specifically, if $\mu^{(i)} = x^{(t)}$

$$P(x^{(t)}|\mu^{(i)}, \sigma_i^2) = P(x^{(t)}|x^{(t)}, \sigma_i^2) = \frac{1}{(2\pi\sigma_i^2)^{d/2}} \exp(\,0\,) \tag{7}$$

which can be made arbitrarily large by decreasing σ_i^2. The resulting Gaussian will turn into a narrow spike around the specific point. In terms of the log-likelihood objective, this is highly advantageous as we can increase the objective value without a bound! What is going on? The problem is that we defined the likelihood objective in terms of generating actual points. As we are dealing with densities (Gaussians over a continuous space), the likelihood of generating a particular point (a real valued vector) is actually zero. In other words, we should really write $P(x|\mu, \sigma^2)\Delta x$ where Δx is a little volume element around the observed point (or, better yet, we should integrate over the volume element). Δx represents the fact that we are unlikely have observed the point with infinite precision. An easier route around this problem is to regularize the variances so that the mixture cannot exploit this caveat.

Regularization and the EM algorithm

Let's begin with the problem of estimating a single Gaussian. We will adjust our log-likelihood objective in such a way that, in addition to the standard log-likelihood, we will add a regularization penalty – now a prior probability – over the potentially difficult variance parameter σ^2. In other words, we will maximize *penalized log-likelihood*

$$\sum_{t=1}^{n} \log P(x|\mu, \sigma^2) + \log P(\sigma^2|\alpha, s^2) \tag{8}$$

where the *hyper-parameters* (parameters of the prior) α and s^2 encode the "default" answer for the variance, and how much we are pushing σ^2 towards it. This is similar to the squared norm regularization penalty in the context of classification or regression.

How should we specify $P(\sigma^2|\alpha, s^2)$? One general way to construct such a prior (conjugacy) is to imagine "prior data" and set the prior distribution equal to the corresponding likelihood (with normalization). The characteristics of the prior data will then determine what the default answer is for the variance, and how strongly we believe in it (how much data we imagine). The more prior data we have, the more the estimate of σ^2 will go towards the default answer. More specifically, suppose our model is a zero mean Gaussian distribution, and we observe α points exactly s^2 distance away from the origin. Then

$$l(D'; \sigma^2) = \left[\frac{1}{(2\pi\sigma^2)^{d/2}} \exp(-\frac{1}{2\sigma^2} s^2) \right]^{\alpha} \tag{9}$$

and we would make $P(\sigma^2|\alpha, s^2)$ proportional to this likelihood. In other words, by specifying such a prior, we are pushing σ^2 towards an answer we would get if we observed α points exactly s^2 distance away from the mean (origin in this case).

How will the prior change the estimated value of σ^2 in the penalized log-likelihood formulation? Recall that if we estimated the variance without any prior penalty, we would get

$$\hat{\sigma}^2 = \frac{1}{dn} \sum_{t=1}^{n} \|x^{(t)} - \hat{\mu}\|^2 \tag{10}$$

By including the prior, we obtain

$$\hat{\sigma}^2 = \frac{1}{d(n+\alpha)} \left[\sum_{t=1}^{n} \|x^{(t)} - \hat{\mu}\|^2 + \alpha\sigma^2 \right] \tag{11}$$

The number of data points is now $n + \alpha$ representing the combination of actual data (size n) and the imagined prior data (size α). Moreover, in the sum that adds the squared distances of observed points from the mean, we included αs^2 to represent α

observations exactly s^2 away from the mean. By increasing α, we steer the variance estimate $\hat{\sigma}^2$ towards the "default" s^2/d. In particular, in the absence of any data, $\hat{\sigma}^2 = s^2/d$. If there's only a single data point $x^{(1)}$, and $\mu = x^{(1)}$ (since this parameter is not regularized), we get $\hat{\sigma}^2 = \frac{1}{d(1+\alpha)}(0 + \alpha s^2)$ rather than zero.

The effect of the prior is analogous within the EM algorithm. We add a prior distribution (penalty in the log-likelihood) for each of the variance parameters σ_i^2. As a result, the E-step remains as before. After all, in this step, the parameters are assumed fixed (we are merely using the current mixture to complete the data). The M-step changes only in terms of the variance update, replacing Eq.(6) with

$$\hat{\sigma}_i^2 = \frac{1}{d(\hat{n}_i + \alpha)} \left[\sum_{t=1}^{n} p(i|t) \|x^{(t)} - \hat{\mu}^{(i)}\|^2 + \alpha s^2 \right], \quad i = 1, \ldots, k \tag{12}$$

analogously to the single Gaussian case. While the change appears small, the effect can be substantial, especially for large k where $\hat{n}_i = \sum_{t=1}^{n} p(i|t)$ may be small for some of the Gaussian components.

Model selection for mixtures

How do we select the number of mixture components, i.e., k? We can certainly run the EM algorithm with each plausible value of k such as $k = 1, 2, \ldots, 10$. As k increases, we may have to run the algorithm multiple times for the same k, however, using different random initializations, in order to find the k mixture (parameters $\hat{\theta}$) that give the highest (penalized) log-likelihood. Let $l(D; \hat{\theta})$ be the log-likelihood value that we achieve with such parameters. Could we just use $l(D; \hat{\theta})$ for selecting k? Unfortunately, not. $l(D; \hat{\theta})$ increases (more precisely, does not decrease) as k increases, regardless of data. See Figure 1 for an example. This is because a mixture with $k + 1$ components already contains a mixture with k components. The additional component could always be set to be equal to one of the others, effectively exploring only a k-mixture. But this is not the best use of the additional degrees of freedom that come with the $k+1^{th}$ component. Note that, in practice, the log-likelihood may decrease with k because mixtures with more components are more difficult to estimate. So the log-likelihood values we can obtain may represent local rather than global optima.

Just as we needed to regularize the choice of the variance parameter earlier, we need to regularize the choice of k. In other words, we need to introduce a penalty for k that appropriately penalizes the fact that mixture models with larger k would fit better any finite training data, even if the data came from a single Gaussian. The penalty we use should take away this unfair advantage from larger k. We are looking for a *model selection* criterion for k.

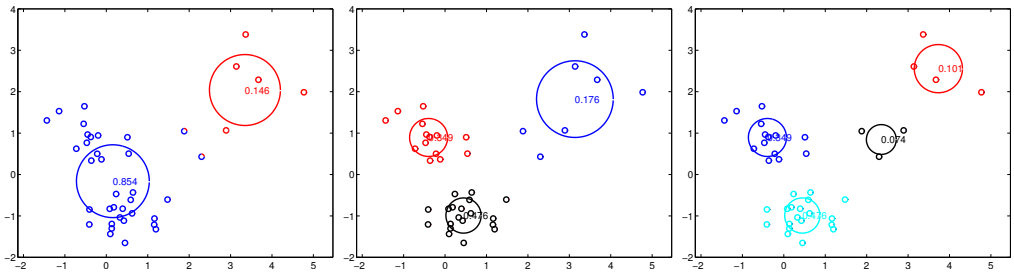
We could always use cross-validation. However, since finding a single $k-$ mixture is already challenging, using cross-validation can be computationally burdensome. Another strategy is to use a simpler approximate criterion. In particular, we will use a

criterion known as the *Bayesian Information Criterion* or BIC for short. It is given by

$$BIC(D;\hat{\theta}) = l(D;\hat{\theta}) - \frac{\#\text{ of param.}}{2}\log(n) \tag{13}$$

where the penalty term approximates the advantage that we would expect to get from larger k regardless of the data. The BIC score is easy to evaluate for each k as we already get $l(D;\hat{\theta})$ from the EM algorithm. All that we need in addition is to evaluate the penalty term. A mixture of k spherical Gaussians in d dimensions has exactly $k(d+2)-1$ parameters. Figure 1 below shows that the resulting BIC score indeed has the highest value for the correct 3-component mixture.

BIC is an asymptotic (large n) approximation to a statistically more well-founded criterion known as the *Bayesian score*. As such, BIC will select the right model (under certain regularity conditions) when n is (very) large relative to d. However, we will adopt the BIC criterion here even for smaller n due to its simplicity.



$l(D;\hat{\theta}) = -118.25$ $l(D;\hat{\theta}) = -98.64$ $l(D;\hat{\theta}) = -94.11$

$BIC(D;\hat{\theta}) = -131.16$ $BIC(D;\hat{\theta}) = -118.93$ $BIC(D;\hat{\theta}) = -121.78$

Figure 1: 2, 3, and 4 component mixtures estimated for the same data. The corresponding log-likelihoods and the BIC scores are shown below each plot.