STA 414/2104

Statistical Methods for Machine Learning and Data Mining

Radford M. Neal, University of Toronto, 2013

Week 8

Review — Maximum Likelihood for Gaussian Mixture Models

A K-component Gaussian mixture model for real vectors models an observation with the density function

$$P(x|\pi, \mu, \Sigma) \quad = \quad \sum_{k=1}^{K} \pi_k \, N(x|\mu_k, \Sigma_k)$$

A natural idea is to estimate π_k, μ_k, and Σ_k for $k = 1, \ldots, K$ by maximizing the likelihood. Assuming that data items are independent, the log likelihood is

$$L(\pi, \mu, \Sigma) \quad = \quad \sum_{i=1}^{n} \log P(x_i|\pi, \mu, \Sigma)$$

where x_i is the data vector for item i.

The EM algorithm

The EM Algorithm for Gaussian Mixture Models

We could use some general purpose optimization method (eg, gradient descent) to find the parmeters of a mixture model that maximize the likelihood (avoiding the singular solutions with $\Sigma_k \to 0$). But a method known as the *EM algorithm* is commonly used, because it is simple to implement, and very stable. (Though it can unfortunately also be rather slow.)

The idea: If we knew which mixture component each data item came from, estimating the mixing proportions and the parameters of each component distribution would be easy. We don't know this, but given an initial guess at the parameters, we can probabilistically assign a component to each data item, and then get a better estimate of the parameters based on these assignments.

This is sort of like the K-means algorithm, but in a probabilistic setting, with a proof that the algorithm will reach a (local) maximum of the likelihood.

Details of the EM Algorthim for Gaussian Mixture Models

Here are the details of the EM algorithm for a Gaussian mixture model with Σ_k being diagonal, with diagonal elements of σ^2_{kj}.

We alternate between "E" (Expectation) steps and "M" (Maximization) steps:

E Step: Using the current values of the parameters, compute the "responsibilities" of components for data items, by applying Bayes' Rule:

$$r_{ik} \;=\; P(\text{data item } i \text{ came from component } k \,|\, x_i) \;=\; \frac{\pi_k \, N(x_i|\mu_k, \Sigma_k)}{\sum\limits_{k'} \pi_{k'} \, N(x_i|\mu_{k'}, \Sigma_{k'})}$$

M Step: Using the current responsibilities, re-estimate the parameters, using weighted averages, with weights given by the responsibilities:

$$\pi_k \;=\; \frac{1}{n}\sum_i r_{ik}, \quad \mu_k \;=\; \sum_i r_{ik}x_i \Big/ \sum_i r_{ik}, \quad \sigma_k^2 \;=\; \sum_i r_{ik}(x_i - \mu_k)^2 \Big/ \sum_i r_{ik}$$

We start with some initial guess at the parameter values (perhaps random), or perhaps with some initial guess at the responsibilities (in which case we start with an M step). We continue alternating E and M steps until there is little change.

The EM Algorithm in General

Consider model for observed data x (which might be a vector of n independent items) that is accompanied by a latent (unobserved) z (also possibly a vector of n independent values). A model with parameters θ describes the joint distribution of x and z, as $P(x, z|\theta)$.

We want to estimate θ by maximum likelihood, which means finding the θ that maximizes

$$P(x|\theta) \quad = \quad \sum_z P(x, z|\theta)$$

(This assumes z is discrete; if it's continuous the sum is replaced by an integral.)

We assume that this isn't easy. But suppose that we *can* easily find the θ that maximizes $P(x, z|\theta)$, for any known x and z. We try to use (something related to) this capability in an iterative algorithm for maximizing $P(x|\theta)$.

The EM Algorithm in General — Details

The general EM algorithm alternates these steps:

E Step: Using the current value of the parameter, θ, find the distribution, Q, for the latent z, given the observed x:

$$Q(z) \quad = \quad P(z|x, \theta)$$

M Step: Maximize the expected value of $\log P(x, z|\theta)$ with respect to θ, where the expectation is with respect to the distribution Q found in the E step:

$$\theta \quad = \quad \arg\max_{\theta} E_Q[\log P(x, z|\theta)]$$

For many models (specifically, those in the "exponential family"), maximizing $E_Q[\log P(x, z|\theta)]$ will be feasible if maximizing $\log P(x, z|\theta)$ for known z is feasible.

Justification of the EM algorithm

To see that the EM algorithm maximizes (at least locally) the log likelihood, consider the following function of the distribution Q over z and the parameters θ:

$$
\begin{aligned}
F(Q,\theta) &= E_Q[\log P(x,z|\theta)] \;-\; E_Q[\log Q(z)] \\
&= \log P(x|\theta) \;+\; E_Q[\log P(z|x,\theta)] \;-\; E_Q[\log Q(z)] \\
&= \log P(x|\theta) \;-\; E_Q[\log(Q(z)/P(z|x,\theta))]
\end{aligned}
$$

The final term above is the “Kullback-Leibler (KL) divergence” between the distribution $Q(z)$ and the distribution $P(z|x,\theta)$. One can show that this divergence is always non-negative, and is zero only when $Q(z) = P(z|x,\theta)$.

We can now justify the EM algorithm by showing that

a) The E step maximizes $F(Q,\theta)$ with respect to Q — a consequence of KL divergence being minimized when $Q(z) = P(z|x,\theta)$.

b) The M step maximizes $F(Q,\theta)$ with respect to θ — clear since $E_Q[\log Q(z)]$ doesn't depend on θ.

c) The maximum of $F(Q,\theta)$ occurs at a θ that maximizes $P(x|\theta)$ — if instead $P(x|\theta^*) > P(x|\theta)$ for some θ^*, then $F(Q^*,\theta^*) > F(Q,\theta)$ with $Q^*(z) = P(z|x,\theta^*)$.

How this Translates to the Mixture Version

For the mixture example, the model parameters are $\theta = (\pi, \mu, \sigma)$.

We'll let the latent variables be $z_{ik} = 1$ if data item i comes from component k, and 0 otherwise.

In the E step, we find the distribution of the z_{ik} given x_i and the model parameters. It turns out that all we actually need from this distribution is the expected value of each z_{ik} (same as the probability that $z_{ik} = 1$), which we define to be r_{ik}, and find by Bayes' Rule as shown before.

In the M step, we need to maximize $E_Q\Big(\sum_{i=1}^{n} \log P(x_i, z_i|\theta)\Big)$.

Suppose we knew the value of both x_i and $z_i = (z_{i1}, \ldots, z_{iK})$ for data item i. The log probability (dropping constant factors) for that item can be written as

$$\log \left[\prod_{k=1}^{K} \left(\pi_k \prod_{j=1}^{p} \left(\frac{1}{\sigma_{kj}} \exp(-(1/2)(x_{ij} - \mu_{kj})^2 / \sigma_{kj}^2) \right) \right)^{z_{ik}} \right]$$

Note that all but one factor in the outer product will have the value one.

We maximize the expected value of the sum of the above for all i, with respect to the distribution of z_i found in the E step. We'll see how this works out next...

Details of the Mixture Version of EM

Taking the expectation of the log probability of data item i with respect to the distribution of z_i (denoted by Q), we get

$$E_Q \left\{ \log \left[\prod_{k=1}^{K} \left(\pi_k \prod_{j=1}^{p} \left(\frac{1}{\sigma_{kj}} \exp(-(1/2)(x_{ij}-\mu_{kj})^2/\sigma_{kj}^2) \right) \right)^{z_{ik}} \right] \right\}$$

$$= \quad E_Q \left\{ \sum_{k=1}^{K} z_{ik} \left(\log(\pi_k) - \frac{1}{2} \sum_{j=1}^{p} \left(\log(\sigma_{kj}^2) + (x_{ij}-\mu_{kj})^2/\sigma_{kj}^2 \right) \right) \right\}$$

$$= \quad \sum_{k=1}^{K} r_{ik} \left(\log(\pi_k) - \frac{1}{2} \sum_{j=1}^{p} \left(\log(\sigma_{kj}^2) + (x_{ij}-\mu_{kj})^2/\sigma_{kj}^2 \right) \right)$$

where $r_{ik} = E_Q(z_{ik})$. To maximize the sum of the above for all i, we separately maximize $\sum_{i=1}^{n} \sum_{k=1}^{K} r_{ik} \log(\pi_k)$ with respect to π, and $-\frac{1}{2} \sum_{i=1}^{n} r_{ik}(x_{ij}-\mu_{kj})^2$ with respect to each μ_{kj}, and finally $-\frac{1}{2} \sum_{i=1}^{n} r_{ik} \left(\log(\sigma_{kj}^2) + (x_{ij}-\mu_{kj})^2/\sigma_{kj}^2 \right)$ with respect to each σ_{kj}^2. This gives the algorithm presented earlier.

How Many Mixture Components Should We Use?

Non-Bayesian Ways of Setting K

If we estimate the parameters of a mixture of K distributions by maximum likelihood, we will "overfit" if we choose K to be too big. Letting $K = n$ is the extreme — then every data point can have its own mixture component, which can give it infinite probability density.

Many, many schemes have been devised for picking an appropriate value for K, most without a convincing justification.

One plausible way is to look at performance on a validation set, with parameters estimated from a separate estimation set (or use S-fold cross validation).

To do this, we need a measure of performance.

We could use the average log probabability of the validation observations.

Or we could use something else that reflects our actual intended use of the results. For instance, we might intend to use a mixture model to fill in missing covariates in a regression model, in which case we might use squared error in prediction of left-out values from other values.

Bayesian Mixture Models

A different approach is to use a Bayesian model, in which we don't predict using a single estimate of the parameters. This will avoid overfitted solutions in which each component models just one data point, infinitely well.

We need to specify a prior distributions for the parameters (eg, mean vector and covariance matrix) of each mixture component.

We might let this prior be independent for each component.

We *cannot* let this prior be improper. If we do, only one component will be used, since the prior probability of a second component having reasonable parameter values will be zero!

We also need a prior for the mixing proportions...

A Prior for Mixing Proportions

A Bayesian mixture model needs to have a prior distribution for the mixing proportions, $\pi_1, \ldots, \pi_K$. One possibility is the *Dirichlet distribution*, which has the following density on the simplex where $\pi_k > 0$ and $\sum \pi_k = 1$:

$$P(\pi_1, \ldots, \pi_K) \quad = \quad \frac{\Gamma(\Sigma_k \alpha_k)}{\Pi_k \, \Gamma(\alpha_k)} \prod_{k=1}^{K} \pi_k^{\alpha_k - 1}$$

The parameters $\alpha_1, \ldots, \alpha_K$ can be any positive reals.

If $Z_1, Z_2, \ldots$ are i.i.d. given π, with $P(Z_i = k) = \pi_k$, the posterior distribution after observing $z_1, \ldots, z_n$, with n_1 of the z_i having value 1, n_2 having value 2, etc. is a Dirichlet distribution with parameters $\alpha_1 + n_1, \ldots, \alpha_K + n_K$.

The predictive distribution for Z_{n+1} given $Z_1, \ldots, Z_n$ is

$$P(Z_{n+1} = k \,|\, Z_1 = z_1, \ldots, Z_K = z_k) \quad = \quad \frac{n_k + \alpha_k}{n + \Sigma_k \alpha_k}$$

Implementing Bayesian Mixture Models

Bayesian mixture models are usually implemented using Markov chain Monte Carlo methods, which we aren't covering in this course.

Just like the EM algorithm for maximum likelihood fitting of mixtures can converge to a bad local maximum, a Markov chain Monte Carlo method for a mixture model can get stuck for a long time in a local mode of the posterior distribution — though it should get out eventually.

Choosing K Using Marginal Likelihood

We can choose among Bayesian mixture models with different numbers of components, K, using the marginal likelihood for each value of K.

The marginal likelihood for these models is rather hard to compute, but it's possible.

But does this procedure make sense? Do we really believe that there is some true (even if unknown) value of K?

Consider a mixture model for symptoms of patients, where we hope the mixture components will represent "diseases". Do we expect only K diseases, for some reasonably small value of K?

As the number of patients, n, increases, we actually expect to see more and more diseases (some of which will be quite rare).

Letting K be Infinite

We can let K go to infinity in a Bayesian mixture model with a Dirichlet prior for $\pi_1, \ldots, \pi_K$ — giving what's called a *Dirichlet process mixture model.*

If we use a Dirichlet prior for $\pi_1, \ldots, \pi_K$ with all parameters being α/K, the limiting form of the "law of succession" for the predictive distribution of Z_i, representing which mixture component to use for item i, is

$$P(Z_i = k \mid z_1, \ldots, z_{i-1}) \;=\; \frac{n_{i,k} + \alpha/K}{i - 1 + \alpha} \;\rightarrow\; \frac{n_{i,k}}{i - 1 + \alpha}$$

$$P(Z_i \neq Z_j \text{ for all } j < i \mid z_1, \ldots, z_{i-1}) \;\rightarrow\; \frac{\alpha}{i - 1 + \alpha}$$

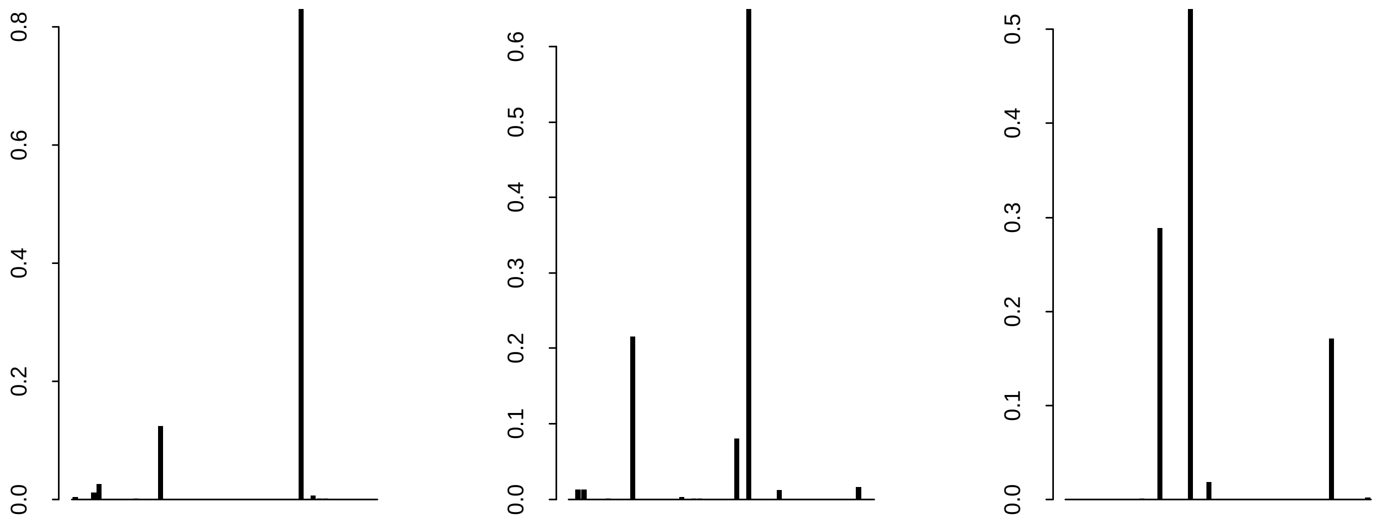
where $n_{i,k}$ is the number of $z_1, \ldots, z_{i-1}$ that are equal to k.

So even with infinite K, behaviour is reasonable: The probability of the next data item being associated with a new mixture component is neither 0 nor 1.

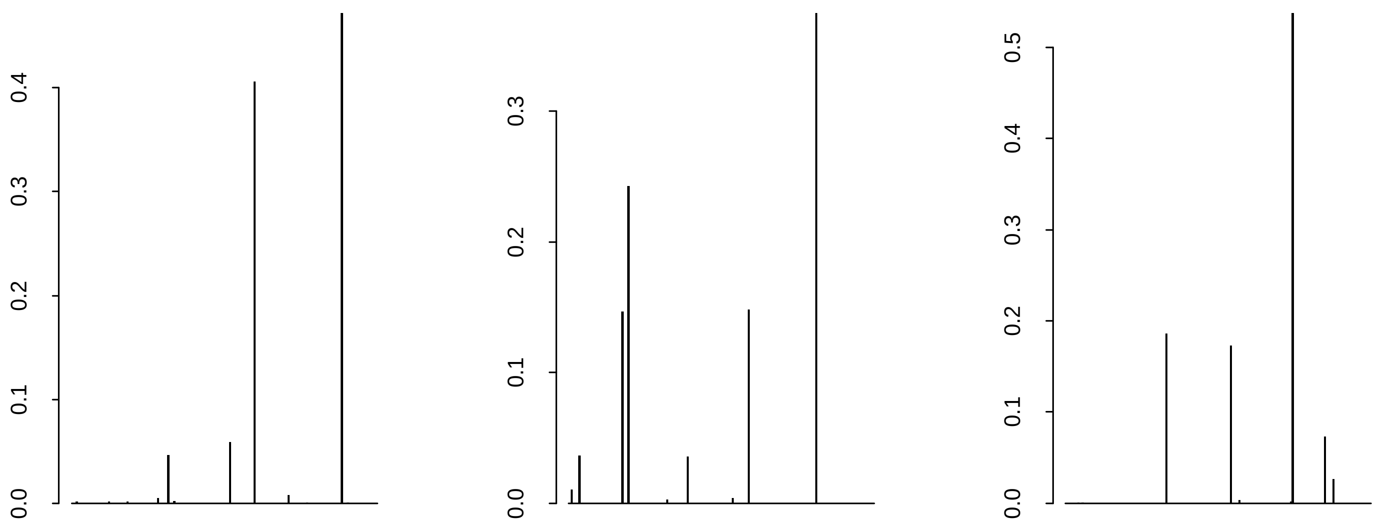
The Prior for Mixing Proportions as K Increases

Three random values from priors for $\pi_1, \ldots, \pi_K$:

$\alpha = 1, \quad K = 50:$



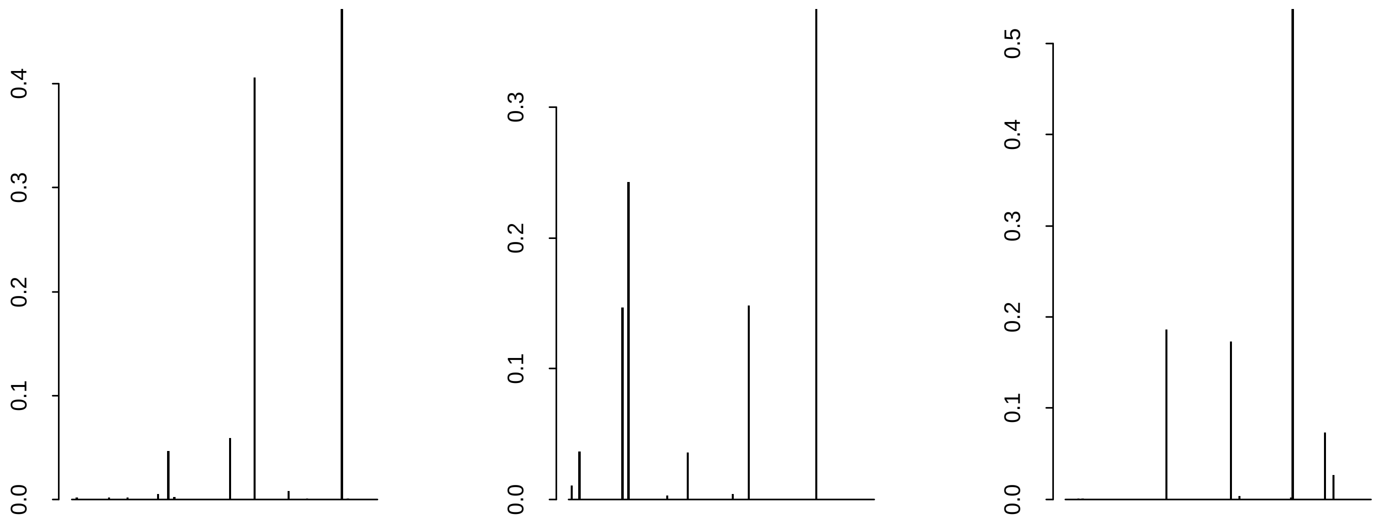
$\alpha = 1, \quad K = 150:$



The Prior for Mixing Proportions as α Varies

Three random values from priors for $\pi_1, \ldots, \pi_K$:

$\alpha = 1, \quad K = 150:$



$\alpha = 5, \quad K = 150:$