Lecturer: David Blei
Scribe: Michael Golightly, Prakash Prabhu

Lecture # 10
March 6, 2008

1 Mixture Models

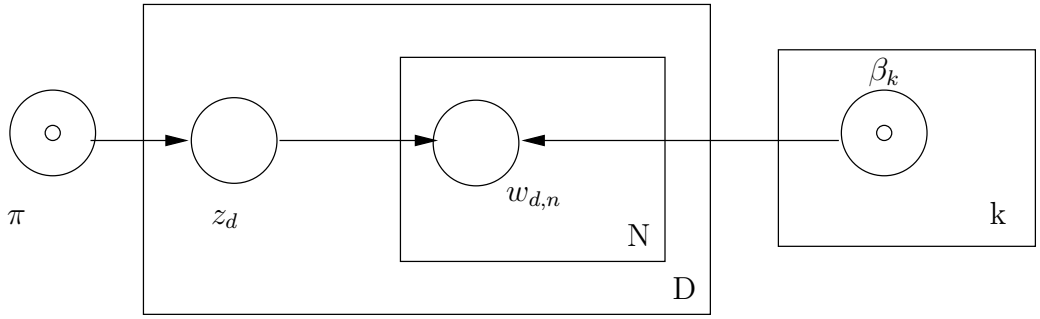

Figure 1: Mixture Model for Document Collections

- π is the parameter representing mixture proportions (a vector that sums to 1)
- D is the document plate
- Z_d is the random variable representing document clusters
- N is the number of words inside each document
- $w_{d,n}$ is a vector of n observed words in document d.
- β_k is the parameter representing mixture components (a mixture of multinomials)
- K is the distribution over words

Now that we have defined our model, we want to determine the maximum likelihood estimate of π and $\beta_{1:k}$ from our observed data $W_{d,1:N}{}_{d=1}^{D}$ This is analogous to finding the K different means that describe our data. An outline of the process is as follows:

For each document d:

- choose $Z_d \sim \pi, Z_d \in \{1, \ldots, k\}$
- choose each word $w_{d,n} \sim \beta_{z_d}$

The data is then given by

$$Data = \{w_{d,1:N}\}_{d:1}^{D}$$

2 Mixture Model Likelihood Function

We define the log likelihood function

$$L(\beta_{1:k}, \pi; Data) = \log \prod_{d=1}^{D} \sum_{z=1}^{k} p(z|\pi) \prod_{n=1}^{N} p(w_{d,n}|\beta_z) = \sum_{d=1}^{D} \log \sum_{z=1}^{k} p(z|\pi) \prod_{n=1}^{N} p(w_{d,n}|\beta_z)$$

To dissect this we note that the marginal probability of a document

$$p(W_{d,1:N}|\pi, \beta_{1:k}) = \sum_{z=1}^{k} p(z|\pi) \prod_{n=1}^{N} p(w_{d,n}|\beta_z)$$

The key difference from Naive Bayes is that here we do not observe the category of each document ($p(z|\pi)$), rather, we sum out ($\sum_{z=1}^{k} p(z|\pi)$) for each possible category according to the π distribution.

3 Expectation Maximization (EM) Algorithm

Problem: The above likelihood function does not decompose.
Solution: The Expectation Maximization (EM) algorithm.
The EM algorithm is a general purpose MLE algorithm for dealing with *any* latent variable model, like z_d above (as against Naive Bayes where there are no latent variables but only observed categories) as long as you can compute the posterior distribution. We continually iterate the algorithm until we have a converged MLE of π and $\beta_{1:k}$. HMMs, Kalman filters, and Baum-Welch are all examples of EM algorithms.

- Expectation(E)-step:

$$p(Z_d|W_{d,1:n}, \pi, \beta_{1:k}) = p(Z_d|\pi)\ell(\pi) \propto p(Z_d|\pi) \prod_{n=1}^{N} p(W_{d,n}|\beta_z)$$

  Given $w_{d,1:N}$ and the current model $\pi, \beta_{1:k}$ we compute the posterior distribution of Z_d (in the above equation, $\ell(\pi)$ is the likelihood of π). We are computing posterior distributions of latent variables given generated data and performing a soft assignment of each data point to a data cluster. This is like naive Bayes classification except we are doing a soft assignment that will be adjusted each iteration of the algorithm.

- Maximization(M)-step:

$$\pi^{new} \propto \sum_{d=1}^{D} p(Z_d|\pi, \beta_{1:k}, W_{d,1:N})$$

  In the M-step, we find a new model based on those posteriors. In this step, the posteriors are fixed and the model is refit to obtain a new setting of our parameters. This step is similar to the K-means step of recomputing the means. However, in EM the clusters are based on soft assignments. With EM we do not have to make hard choices! We could have data points that are 50% in one cluster and 50% in another etc. Having a probabilistic distribution makes E-M more straightforward to generalize this to new situations.

Expected number of times we see each mixture component assignment:

$$\pi_k^{new} \propto \sum_{d=1}^{D} E[1[Z_d = 1]|W_{d,1:N}]$$

normalizing constant = d = number of documents

Probability that we see B^{th} word in document of component k:

$$\beta_{k,v}^{new} \propto$$

expected number of times that you see word v in a document of class k:

$$E[\sum_{d=1}^{D}\sum_{n=1}^{N} 1[W_{d,n} = v]1[_d= k]] =$$

$$\sum_{d=1}^{D}\sum_{n=1}^{N} 1[W_{d,n} = v]p(Z_d = k|\beta_{1:k}, \pi, w_{d,n})$$

Expected number of times we see document of component k, where expectation is taken with respect to the posterior Z_d is π^k

4 Example

E-step: Go through each document, is it more about sports health or business? now have a distribution over sports health and business. K-means assigns each document to a cluster. M-step: For each word, how often was it softly classified? In K-means we re-estimated cluster centers, in EM we re- estimate word probabilities.

d = document n = word in each doc

$$C_{\beta_k} = \sum_{v=1}^{V}\sum_{d=1}^{D}\sum_{n=1}^{N} 1[w_{d,n} = v]p(Z_d = k|\dots) = \sum_{d=1}^{D} p(Z_d = k|\dots)\sum_{v}\sum_{n} 1[w_{d,n} = v] = N\sum_{d=1}^{D} p(Z_d = k|\dots)$$

EM algorithm tries to find a fixed point of the expected complete log likelihood. It is always increasing the original log likelihood. Each iteration of an E step and M step gives us a new π and β, each time increasing the log likelihood function.

$$E[\sum_{d=1}^{D}\{\log p(Z_d|\pi) + \sum_{n=1}^{N}\log p(W_{d,n}|\beta_{z_d})\}|w, \pi, \beta)]$$

function of π and β: posterior depends on them

First part means that if we have a complete log likelihood of all our random variables though, it is as if we observed them all. This is the objective function that the EM algorithm is optimizing, it keeps iterating until it converges.

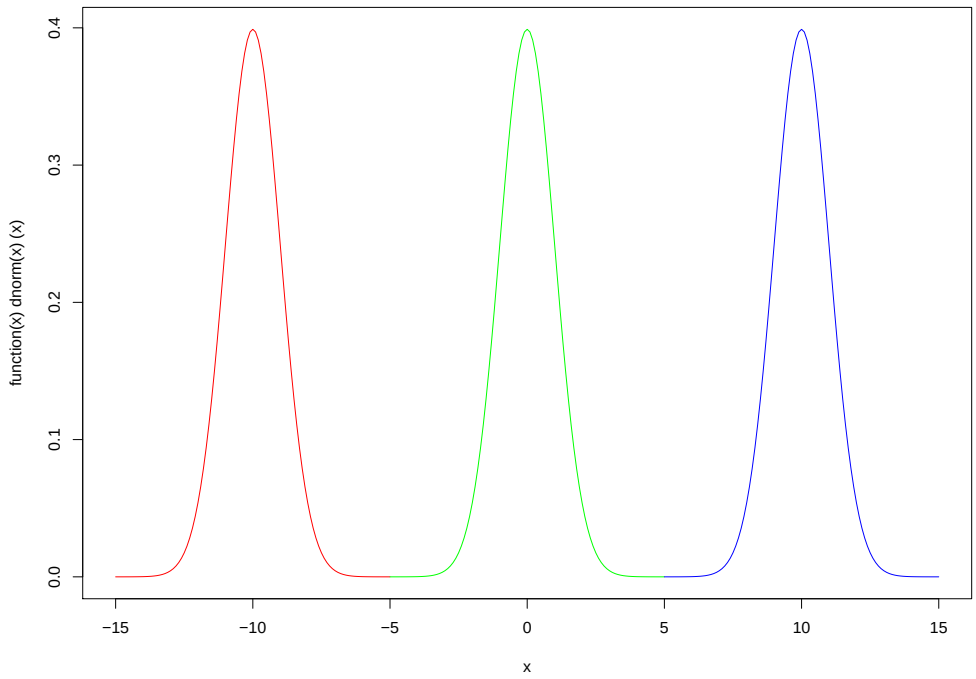

Figure 2: Mixture of 3 gaussians

5 Summary

Mixture models allow for *any* data generating distribution and mixture models *are* a distribution.

$$p(x|\pi,\theta) = \sum_{z=1}^{Z} p(z|\pi)p(x|\theta_z)$$

Statisticians have proved that any distribution – up to certain conditions – can be represented as a mixture of Gaussians.