# COS 513: MIXTURE MODELS AND E-M ALGORITHM

AHMET EMRE BARUT

## 1. Mixture Models

We start by comparing three different graphical models:

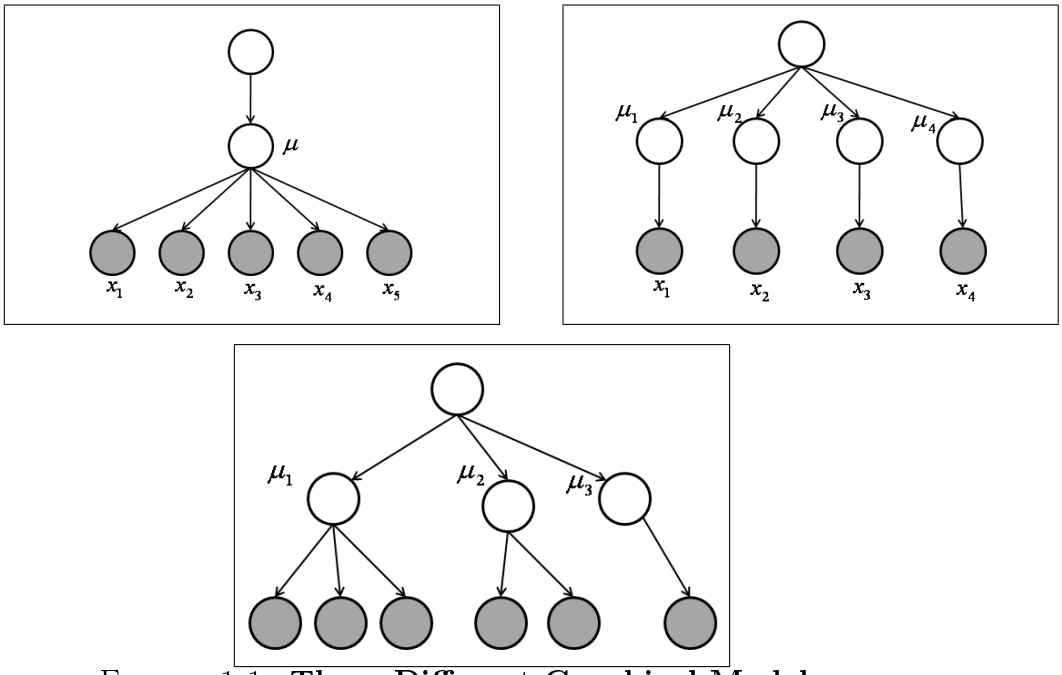


FIGURE 1.1. **Three Different Graphical Models**.
*Top left:* Figure 1.1a is a model where a maximum likelihood estimate will be enough.
*Top right:* Figure 1.1b is, under almost all cases, a useless model, as any data can be represented in this form.
*Bottom:* Figure 1.1c is a mixture model with three clusters.

In all cases, we assume that our focus is on the middle parameter (indexed by $\mu$). The first model is a fairly basic one, and a maximum likelihood estimate (MLE) will be sufficient enough. The second

graph (1.b), is not specific to any model and almost any dataset can be represented this way.

The third one however, gives a nice comprimise between the other two. It has some structure that we can exploit but it's still a good model for non-homogenous data. These models are called mixture models. Each group of the data is called a cluster (in this case we have $K = 3$ clusters). It's easy to see that, once the clusters are known, the problem of estimating the parameters becomes so much simpler. Unfortunately, with actual data, we don't know the graph structure. Therefore, estimation of the parameters and the cluster structure require more advanced methods. We will use the following graphical model for mixture models.

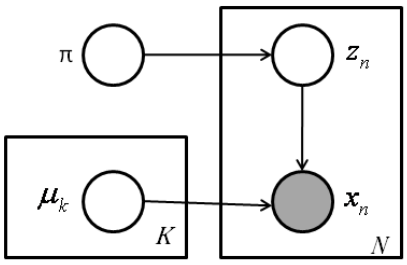


FIGURE 1.2. Mixture Model Graph

Here $z_n$ is a multinomial random variable and

$$z_n^k = \begin{cases} 1 & \text{if instance } n\text{is from component } k \\ 0 & \text{otherwise} \end{cases}.$$

$\pi = (\pi_1, \ldots, \pi_K)$ are the mixture proportions. In other words, $\pi_k$ is the probability of having $x_n$ belong to the $i$-th cluster[1]. They are used to represent our prior over choice of components. $\mu_1, \ldots, \mu_K$ are the mean parameters for the components 1 to $K$.

Before moving on, we also note that for the models considered here, we don't estimate the number of clusters $K$ and we fix it. Also, our goal is not to estimate the posterior distrubtions of either $\mu_k$ or $z_n$ but rather point estimation of these parameters.

**Complete Data Likelihood.** To estimate the parameters, we start with the simplest question: How do we write down the likelihood if

[1]Since $\sum_{i=1}^{K} \pi_i = 1$, we have $\pi_K = 1 - \sum_{i=1}^{K-1} \pi_i$. Then estimation of these parameters require $K - 1$ many unknows rather than $K$.

we knew $z_n^k$? Given the mixture distributions $\pi_k$ and the necesssary parameters for each component $\theta_k$, we can write the likelihood as

$$p(x, z|\theta, \pi) = \prod_n \sum_k z_n^k p\left(x_n|\theta_k\right) \pi_k,$$

where for each $k$, only one of the terms $z_n^k$ is equal to 1 and the rest are zero. We call this the **complete data likelihood**.

**Example.** In order to show how to derive the complete data likelihood, we consider the following example. We consider data coming from two clusters ($K = 2$) with four data points ($N = 4$). We denote the parameters for clusters by $\lambda_1$ and ${\lambda_2}^2$. The data are assumed to come from a Poisson distribution which has the following probability density function

$$p(x|\lambda) = \frac{e^{-\lambda}\lambda^x}{x!}.$$

The data points and their clusters are given in Table 1.1.

| | $x_i$ | $z_i$ |
|---|---|---|
| 1 | 2 | 1 |
| 2 | 7 | 2 |
| 3 | 3 | 1 |
| 4 | 9 | 2 |

TABLE 1.1. Data points and clusters for Complete Data Likelihood Example

The complete data likelihood equals

$$\begin{aligned} p(x, z|\lambda, \pi) &= \prod_n \sum_k z_n^k p\left(x_n|\lambda_k\right) \pi_k \\ &= \frac{e^{-\lambda_1}\lambda_1^2}{2!}\pi_1 \frac{e^{-\lambda_2}\lambda_2^7}{7!}\pi_2 \frac{e^{-\lambda_1}\lambda_1^3}{3!}\pi_1 \frac{e^{-\lambda_2}\lambda_2^9}{9!}\pi_2. \end{aligned}$$

We then take the log to get the log-likelihood

$$\begin{aligned} \log\left(p(x, z|\lambda, \pi)\right) &= \sum_n -\lambda_{z_n} + x_n \log\left(\lambda_{z_n}\right) - \log\left(x_n!\right) + \log\left(\pi_{z_n}\right) \\ &= -2\lambda_1 + (2+3)\log\left(\lambda_1\right) - \log\left(2!\right) - \log\left(3!\right) + 2\log\left(\pi_1\right) \\ &\quad -2\lambda_2 + (7+9)\log\left(\lambda_2\right) - \log\left(7!\right) - \log\left(9!\right) + 2\log\left(\pi_2\right). \end{aligned}$$

[2] $\theta = \lambda$

To get the maximum-likelihood estimate (MLE) for $\lambda_1$, we take the derivative of the log likelihood with respect to $\lambda_1$. Denoting $n_k$ the number of samples from cluster $k$, we can write down this derivative as

$$\frac{\partial \log(p(x, z|\lambda, \pi))}{\partial \lambda_1} = -n_1 + \frac{\sum_{z_n^k=1} x_n}{\lambda_1}.$$

If we set the derivative equal to zero, and solve for $\lambda_1$, we get the MLE estimate for $\lambda_1$

$$\hat{\lambda}_1 = \frac{\sum_{z_n^k=1} x_n}{n_1},$$

in other words, the estimate for $\lambda_1$ for this example is the sample average from cluster 1. We can repeat the same procedure to obtain the same result for $\lambda_2$.

With the complete data likelihood, we can also find the MLE estimates for the mixing probability $\pi_1$. First note that, $\pi_2 = 1 - \pi_1$. Then we can write the log-likelihood as

$$\begin{aligned}\log\left(p(x, z|\lambda, \pi)\right) &= -2\lambda_1 + (2+3)\log(\lambda_1) - \log(2!) - \log(3!) + 2\log(\pi_1) \\ &\quad -2\lambda_2 + (7+9)\log(\lambda_2) - \log(7!) - \log(9!) + 2\log(1-\pi_1).\end{aligned}$$

Taking the derivative with respect to $\pi_1$, we get

$$\frac{\partial \log\left(p(x, z|\lambda, \pi)\right)}{\partial \pi_1} = \frac{n_1}{\pi_1} - \frac{n_2}{1-\pi_1}.$$

Then, we set this value equal to zero and solve for $\pi_1$, which gives $\hat{\pi}_1 = \frac{n_1}{n_1+n_2}$. That is, MLE estimate for the mixing probabilities are given by the ratio of the sample sizes from each cluster.

**Marginal Likelihood.** In the formulation of the complete data likelihood and the above example, we assumed that we knew which clusters the points belong to. However, with real data, as we don't know which points belong to which cluster, we need to consider the likelihood only over $x$. To calculate this term , we make use of the complete data likelihood and we sum over possible $z$'s to obtain

$$p(x|\theta, \pi) = \sum_z p(x, z|\theta, \pi) = \sum_z \prod_n \sum_k z_n^k p\left(x_n|\theta_k\right) \pi_k.$$

The above expression is called the **marginal likelihood**. Unfortunately, evaluating the marginal likelihood is often computationally infeasible. To see this, consider a simple example with two clusters and 250 data points. The summation over possible clusters (where all combinations of $z_n^k$ are considered) requires summation over $K^N$ terms. In this example, $K = 2$ and $N = 250$, and $K^N$ is huge[3]. Furthermore,

[3]In fact, it is larger than the number of atoms in the observable universe.

we can't even consider the log of the marginal likelihood, as the summation over $z$ terms "blocks" the log.

From the formulation, it is obvious that a different method is needed to maximize the marginal likelihood in terms of the parameters $\pi$ and $\theta$. The E-M algorithm considered in the following section is one of "the best" (and most of the time "the only") techniques to solve this.

## 2. E-M Algorithm

The Expectation-Maximization (E-M) algorithm was introduced by Dempster et al. in their 1977 paper[4]. The algorithm gets around the problem of maximizing the marginal likelihood by assuming that it has the "expected complete data likelihood". The expected complete data likelihood replaces terms $z_n^k$ by the expected value of these random variables. Now, instead of binary variables $z_n^k$, we have

$$\tau_n^k = \mathbb{E}\left[z_n^k | x_n, \theta, \pi\right] = p\left(z_n^k = 1 | x_n, \theta, \pi\right) \propto p\left(z_n^k = 1 | \pi\right) p\left(x_n | \theta_k\right) \propto \pi_i p\left(x_n | \theta_k\right),$$

which take values between 0 and 1 and represent the probability that $x_n$ belongs to cluster $k$.

After putting in $\tau_n^k$ instead of $z_n^k$ to the complete data likelihood, and taking the log we obtain the **expected complete log-likelihood**, which is a lot simpler to evaluate compared to the marginal likelihood. The expected complete data likelihood can be calculated by

$$\mathbb{E}_\tau\left[\log\left(\mathbb{P}(x, z | \theta, \pi)\right)\right] = \sum_n \sum_k \tau_n^k \log \pi_i p\left(x_n | \theta_k\right).$$

We can then maximize over $\theta$ and $\pi$ given the expected complete data likelihood to find better fits to the data. This is how the E-M algorithm obtains new parameters. These iterations are repeated until convergence. See the following box for the pseudo-code.

[4] *A. P. Dempster, N. M. Laird, and D. B. Rubin.* 1977. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society, Series B, 34:1-38.

**Algorithm 1** E-M Algorithm

**Input**: $\mathcal{D} = (x_i|1:N)$ (data),
$K$(number of clusters),
$\theta_0, \pi_0$ (initial values for estimates of $\theta$ and $\pi$)

**Output**: $\theta_T, \pi_T$ (estimates for parameters $\theta$ and $\pi$)

- Set $t = 0$ (a counter)
- Repeat until convergence ($\|\theta_t - \theta_{t-1}\| < \epsilon, \|\pi_t - \pi_{t-1}\| < \epsilon$):
  - **E-Step:** Holding $\theta_t$ and $\pi_t$ fixed, evaluate
    $$\tau_n^k = \mathbb{E}\left[z_n^k | x_n, \theta_t, \pi_t\right]$$
    and write down the expected complete log-likelihood:
    $$\mathbb{E}_\tau\left[\log\left(\mathbb{P}(x, z|\theta_t, \pi_t)\right)\right]$$
  - **M-Step:** Holding $\tau_n^k$ fixed, let
    $$(\theta_{t+1}, \pi_{t+1}) = \arg\max_{\theta,\pi} \mathbb{E}_\tau\left[\log\left(\mathbb{P}(x, z|\theta, \pi)\right)\right]$$
  - $t = t + 1$ (increase the counter)

**Example.** We consider the same data set as in Section 1, but now, we assume that we don't know the underlying cluster structure. We ran the E-M algorithm four times with different initial $\lambda$ values for the two clusters. We stopped the algorithm when the change in the parameters $\left(\left|\lambda_1^{new} - \lambda_1^{old}\right| + \left|\lambda_2^{new} - \lambda_2^{old}\right|\right)$ was less than 0.001. Table 2.1 summarizes the results. The first column is the initial $\lambda$ values for the clusters and the second column is the number of iterations until convergence. The final estimates, $\hat{\lambda}_1$ and $\hat{\lambda}_2$ are given in the last two columns.

| $\lambda_1^{initial}$ | $\lambda_2^{initial}$ | Iterations until convergence | $\hat{\lambda}_1$ | $\hat{\lambda}_2$ |
|---|---|---|---|---|
| 1 | 4 | 19 | 2.683 | 7.401 |
| 2.5 | 8 | 11 | 2.683 | 7.401 |
| 4 | 1 | 19 | 7.401 | 2.683 |
| 30 | 35 | 45 | 7.401 | 2.683 |

TABLE 2.1. E-M Algorithm Results

It is seen that the algorithm converged to the same solution even under different initial $\lambda$ values (for the 3rd and the 4th runs, the clusters were chosen such that the 1st cluster was actually the 2nd one and etc.). However, this is not always the case for the E-M algorithm, and to avoid a local minimum, the algorithm should be run several times under different initial values.