EECS 445 Discussion 11: Gaussian Mixture Models and Bayesian Networks, Part 1

EECS 445 Staff

April 7, 2021

Recall that a machine learning model that performs classification or regression is called a *discriminative model*. Here's an entirely different framework: let's use a *generative model* to try to learn the data distribution $p(\bar{x})$ directly! Generative models parameterize a probability distribution and learn parameter values from observed data. In other words, we want to create a probability distribution $p_{\bar{\theta}}(\bar{x})$ as similar as possible to $p(\bar{x})$.

1 MLE of the Univariate Gaussian

Let's review the MLE of the univariate Gaussian distribution from last week's discussion - this will be essential knowledge for the next section on Gaussian Mixture Models.

Let $x \sim \mathcal{N}(x;\mu,\sigma^2)$ and $\bar{\theta} = (\mu, \sigma^2)$, where x is an observation, μ is the mean of the distribution, and σ^2 is the variance of the distribution. What are the maximum likelihood estimates of μ and σ^2? The univariate normal distribution is as follows.

$$\mathcal{N}(x;\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\Big[-\frac{1}{2\sigma^2}(x-\mu)^2\Big] \tag{1}$$

We first compute the log-likelihood as follows:

$$\begin{aligned}
\ln \mathcal{L}(X) &= \ln \prod_{i=1}^{n} \mathcal{N}(x^{(i)};\mu,\sigma^2) \\
&= \sum_{i=1}^{n} \ln \Big[\mathcal{N}(x^{(i)};\mu,\sigma^2)\Big] \\
&= \sum_{i=1}^{n} \ln \Big[\frac{1}{\sqrt{2\pi\sigma^2}}\exp\big(-\frac{1}{2\sigma^2}(x^{(i)}-\mu)^2\big)\Big] \\
&= \sum_{i=1}^{n} -\ln(\sqrt{2\pi\sigma^2}) + \big(-\frac{1}{2\sigma^2}(x^{(i)}-\mu)^2\big) \\
&= \sum_{i=1}^{n} -\frac{1}{2}\ln(2\pi\sigma^2) + \big(-\frac{1}{2\sigma^2}(x^{(i)}-\mu)^2\big)
\end{aligned} \tag{2}$$

Next, we optimize with respect to μ by taking the derivative and setting it equal to 0:

$$\begin{aligned}
\nabla_\mu \ln \mathcal{L}(X) &= \sum_{i=1}^{n}(\frac{1}{\sigma^2}(x^{(i)} - \mu)) \\
0 &= \frac{1}{\sigma^2}\sum_{i=1}^{n}((x^{(i)} - \mu)) \\
0 &= \sum_{i=1}^{n} x^{(i)} - \sum_{i=1}^{n} \mu \\
0 &= \sum_{i=1}^{n} x^{(i)} - n\mu \\
n\mu &= \sum_{i=1}^{n} x^{(i)} \\
\hat{\mu} &= \frac{1}{n}\sum_{i=1}^{n} x^{(i)}
\end{aligned} \tag{3}$$

We achieve the intuitive result that the MLE of the mean is just the sample average of the observations. Next, we optimize with respect to σ^2:

$$\begin{aligned}
\nabla_{\sigma^2} \ln \mathcal{L}(X) &= \sum_{i=1}^{n} -\frac{1}{2\sigma^2} + \frac{(x^{(i)} - \mu)^2}{2(\sigma^2)^2} \\
0 &= \sum_{i=1}^{n} -\frac{1}{2\sigma^2} + \frac{(x^{(i)} - \mu)^2}{2(\sigma^2)^2} \\
0 &= \sum_{i=1}^{n} -\frac{1}{2\sigma^2} + \sum_{i=1}^{n} \frac{(x^{(i)} - \mu)^2}{2(\sigma^2)^2} \\
0 &= -\frac{n}{2\sigma^2} + \sum_{i=1}^{n} \frac{(x^{(i)} - \mu)^2}{2(\sigma^2)^2} \\
\frac{n}{2\sigma^2} &= \sum_{i=1}^{n} \frac{(x^{(i)} - \mu)^2}{2(\sigma^2)^2} \\
n &= \sum_{i=1}^{n} \frac{(x^{(i)} - \mu)^2}{\sigma^2} \\
\hat{\sigma^2} &= \sum_{i=1}^{n} \frac{(x^{(i)} - \hat{\mu})^2}{n}
\end{aligned} \tag{4}$$

Thus, we have $\bar{\theta}^* = \left(\frac{1}{n}\sum_{i=1}^{n} x^{(i)}, \sum_{i=1}^{n} \frac{(x^{(i)} - \mu)^2}{n}\right)$.

2 Gaussian Mixture Models

A **mixture distribution** is a special kind of probability distribution that consists of a weighted combination of multiple distributions. It is most commonly used to represent "sub-populations" within some overall population.

In a **Gaussian mixture model** (GMM), we assume that the underlying data distribution is a mixture of k Gaussian distributions. Each Gaussian is parameterized by a mean and covariance matrix. Training a GMM involves learning values for these means and covariances, as well as learning prior assignment probabilities to each Gaussian – these are called "mixing weights".

We will work with the following notation, assuming $\bar{x} \in \mathbb{R}^d$:

$$\begin{aligned} \gamma_1, \dots, \gamma_k \quad & \text{mixing weights} \\ \bar{\mu}_1, \dots, \bar{\mu}_k \quad & \text{cluster means} \\ \Sigma_1, \dots, \Sigma_k \quad & \text{cluster covariance matrices} \end{aligned} \tag{5}$$

$$\bar{\theta} = \{\gamma_1, \dots, \gamma_k, \bar{\mu}_1, \dots, \bar{\mu}_k, \Sigma_1, \dots, \Sigma_k\} \qquad \text{model parameters}$$

Note that we are assuming our Gaussians are all spherical, such that $\Sigma_j = \sigma_j^2 I$ for some scalar σ_j. After learning these values, we can generate new values from our GMM as follows:

$$\begin{aligned} z^{(i)} &\sim \text{Categorical}(\bar{\gamma}) && \text{cluster indicators} \\ f(\bar{x}^{(i)} \mid z^{(i)}; \theta) &\sim \mathcal{N}(\bar{\mu}_{z^{(i)}}, \Sigma_{z^{(i)}}) && \text{base distribution} \end{aligned} \tag{6}$$

In other words, we first sample which Gaussian the data point should come from and then we sample from that particular Gaussian to generate the data point.

2.1 Complete Data Case

We will begin by assuming that we magically know the cluster assignment $z^{(i)}$ for each data point, also called the hidden or *latent* values of the generative model. When this knowledge is assumed, we say that our data is "complete". Without any training data, the best we can do is assume that points are distributed based on our mixing weights—also called priors. This yields the following probability model for sampling a data point $\bar{x}^{(i)}$:

$$p_{\bar{\theta}}(\bar{x}^{(i)}) = \sum_{j=1}^{k} \gamma_j \mathcal{N}(\bar{x}^{(i)}; \bar{\mu}_j, \Sigma_j) \tag{7}$$

In the complete data case, the conditional probability of cluster assignment can only result in 0s and 1s (i.e., we have a "ground truth" cluster). We can embed this prior into our model as follows.

$$p(j|i) = \begin{cases} 1, & \text{if } z^{(i)} = j \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

$$p_{\bar{\theta}}(\bar{x}^{(i)}|z^{(i)}) = \sum_{j=1}^{k} p(j|i) \mathcal{N}(\bar{x}^{(i)}; \bar{\mu}_j, \Sigma_j) \tag{9}$$

We will later relax this assumption and allow the prior to take on more complex distributions.

Discussion Question 1. *Let $X = [\bar{x}^{(1)}, \dots, \bar{x}^{(n)}]$ and $Z = [z^{(1)}, \dots, z^{(n)}]$. Assume our data is independently sampled. Show that the following is true:*

$$\ln p_{\bar{\theta}}(X, Z) = \sum_{i=1}^{n} \sum_{j=1}^{k} p(j|i) \ln \Big(\gamma_j \mathcal{N}(\bar{x}^{(i)}; \bar{\mu}_j, \Sigma_j) \Big) \tag{10}$$

Solution:

$$\begin{aligned}\ln p_{\bar\theta}(X,Z) &= \ln\prod_{i=1}^{n} p_{\bar\theta}(\bar{x}^{(i)},\bar{z}^{(i)})\\ &= \ln\prod_{i=1}^{n}\sum_{j=1}^{k} p(j|i)\gamma_j\mathcal{N}(\bar{x}^{(i)};\bar\mu_j,\Sigma_j)\\ &= \sum_{i=1}^{n}\ln\sum_{j=1}^{k} p(j|i)\gamma_j\mathcal{N}(\bar{x}^{(i)};\bar\mu_j,\Sigma_j)\\ &= \sum_{i=1}^{n}\sum_{j=1}^{k} p(j|i)\ln\Big(\gamma_j\mathcal{N}(\bar{x}^{(i)};\bar\mu_j,\Sigma_j)\Big)\end{aligned}$$

Note that the last step can only be justified when $p(j|i) \in 0, 1$.

2.1.1 Maximum Likelihood Estimation

Now that we've defined our probabilistic model, we need to construct a method for optimizing our model's parameters with respect to the input data $\{\bar{x}^{(i)}, z^{(i)}\}_{i=1}^{n}$. The process is similar to our previous means of optimization—take the gradient with respect to the parameter of interest, set it equal to 0, and solve—but now it goes by a different name: **maximum likelihood estimation** (MLE).

Discussion Question 2. *Find γ_j^*, the MLE of the cluster prior of cluster j. Note that we have the additional constraint that $\sum_{j=1}^{k}\gamma_j = 1$.*

Solution: Our goal is to solve the following optimization problem

$$\max_{\alpha}\min_{\gamma_j}\quad -\sum_{i=1}^{n}\sum_{j=1}^{k} p(j|i)\ln\Big(\gamma_j\mathcal{N}(\bar{x}^{(i)};\bar\mu_j,\Sigma_j)\Big) - \alpha\left(1-\sum_{j=1}^{k}\gamma_j\right) \tag{11}$$

We find the gradient ∇_{γ_j} to be

$$-\sum_{i=1}^{n}\frac{p(j|i)}{\gamma_j} + \alpha \tag{12}$$

Let $\hat{n}_j = \sum_{i=1}^{n} p(j|i)$ be the number of points in cluster j, then by setting equal to zero and solving for γ_j^* we find that

$$\gamma_j^* = \frac{\hat{n}_j}{\alpha} \tag{13}$$

Using this constraint along with our original constraint that $\sum_{j=1}^{k}\gamma_j = 1$ as well as the property that $\sum_{j=1}^{k}\hat{n}_j = n$ (this is just the sum of points over all clusters), we find that $\alpha = n$, giving us that the optimal value of γ_j is:

$$\gamma_j^* = \frac{\hat{n}_j}{n} \tag{14}$$

2.2 Incomplete Data Case

Recall the complete data log-likelihood formulation from above:

$$\ln p_{\bar{\theta}}(X, Z) = \sum_{i=1}^{n} \sum_{j=1}^{k} p(j|i) \ln \left(\gamma_j \mathcal{N}(\bar{x}^{(i)}; \bar{\mu}_j, \Sigma_j) \right)$$

In the incomplete data case, we don't know which cluster a point is in (mathematically, $p(j|i)$ is unknown). That means, at least initially, the best we can do is just *randomly guess*. We initially guess random parameters for each of our distributions. Then, we use an two-step iterative process called the **expectation-maximization** (EM) algorithm to find distribution parameters that fit our data well.

2.2.1 Expectation-Maximization

Estimation: Given our current guess of the parameters, calculate a "soft" assignment. Basically, we want to construct a probability distribution for which cluster a point is in. Formally, for a given cluster j and point i,

$$p(j|i) = \frac{\gamma_j \mathcal{N}(\bar{x}^{(i)}|\bar{\mu}_j, \sigma_j^2)}{\sum_{t=1}^{k} \gamma_t \mathcal{N}(\bar{x}^{(i)}|\bar{\mu}_t, \sigma_t^2)} \tag{15}$$

Maximization: We already know the maximum likelihood estimators for a known dataset. Since we calculated the soft assignments, we "know" the assignments, and we can calculate the MLEs we found in lecture:

$$\begin{aligned} \hat{n}_j &= \sum_{i=1}^{n} p(j|i) \\ \hat{\gamma}_j &= \frac{\hat{n}_j}{n} \\ \hat{\bar{\mu}}^{(j)} &= \frac{1}{\hat{n}_j} \sum_{i=1}^{n} p(j|i) \bar{x}^{(i)} \\ \hat{\sigma^2}_j &= \frac{1}{\hat{n}_j} \sum_{i=1}^{n} p(j|i) ||\bar{x}^{(i)} - \hat{\bar{\mu}}^{(j)}||^2 \end{aligned} \tag{16}$$

Now we have new parameters for each cluster, so the soft assignments to the clusters will be different, and we can repeat these two steps until convergence.

Note that expectation-maximization can be viewed as an instance of **coordinate descent**, which you have previously seen in the k-means algorithm for clustering. Thus, like k-means, the EM algorithm is guaranteed to converge. However, there are similar issues with converging to local, rather than global, maxima.

Let's do an example. The following table contains information on the number of days it has rained in the past month for 5 different cities:

city	# of rainy days
A	2
B	3
C	4
D	6
E	8

Let's work through the process of assigning these 5 cities to $k = 2$ soft clusters using the EM algorithm in the following questions.

Discussion Question 3. *Let $\bar{\theta}^{(0)} = [0.5, 0.5, 4, 5.2, 0.81, 1]$. What are the spherical Gaussian distributions our current model is composed of?*

Solution: $\mathcal{N}(4, 0.81)$ and $\mathcal{N}(5.2, 1)$. Remember that the first 2 values in $\bar{\theta}$ correspond to the $\hat{\gamma}$ values.

Discussion Question 4. *The table below contains the soft cluster assignment values for a respective city and cluster. Fill in the missing values.*

	cluster 1	*cluster 2*
A	0.940	0.060
B	0.870	0.129
C	*?*	*?*
D	0.114	0.885
E	*?*	*?*

Solution:
First, let's find $p(1|C)$:

$$p(1|C) = \frac{\hat{\gamma}_1 \mathcal{N}(\bar{x}^{(C)}|\bar{\mu}_1, \sigma_1^2)}{\sum_{t=1}^{k} \hat{\gamma}_t \mathcal{N}(\bar{x}^{(C)}|\bar{\mu}_t, \sigma_t^2)} = \frac{0.5(0.443)}{0.5(0.443) + 0.5(0.194)} = \mathbf{0.695}$$

City C must be in either cluster 1 or cluster 2, so the total probability should sum up to 1. Therefore, $p(2|C) = 1 - p(1|C) = 1 - 0.695 = \mathbf{0.305}$.

Next, let's find $p(1|E)$:

$$p(1|E) = \frac{\hat{\gamma}_1 \mathcal{N}(\bar{x}^{(E)}|\bar{\mu}_1, \sigma_1^2)}{\sum_{t=1}^{k} \hat{\gamma}_t \mathcal{N}(\bar{x}^{(E)}|\bar{\mu}_t, \sigma_t^2)} = \frac{0.5(0.000022)}{0.5(0.008) + 0.5(0.000022)} = \mathbf{0.003}$$

Since $p(1|E) = 0.003$, $p(2|E) = 1 - 0.003 = \mathbf{0.997}$. Think about why the probability values are so different for this city.

The completed table looks like this:

	cluster 1	cluster 2
city *A*	0.940	0.060
city *B*	0.870	0.129
city *C*	**0.695**	**0.305**
city *D*	0.114	0.885
city *E*	**0.003**	**0.997**

Discussion Question 5. *Given the completed table from Discussion Question 4, Find $\bar{\theta}^{(1)}$.*

Solution:
Model parameters for Cluster 1:

$$\begin{aligned}\hat{n}_1 &= \sum_{i\in\{A,\ldots,E\}} p(1|i)\\ &= 0.94 + 0.87 + 0.695 + 0.114 + 0.003\\ &= \mathbf{2.624}\end{aligned}$$

$$\begin{aligned}\hat{\gamma}_1 &= \frac{\hat{n}_1}{n}\\ &= \frac{2.624}{5}\\ &= \mathbf{0.525}\end{aligned}$$

$$\begin{aligned}\hat{\mu}^{(1)} &= \frac{1}{\hat{n}_1}\sum_{i\in\{A,\ldots,E\}} p(1|i)\bar{x}^{(i)}\\ &= \frac{1}{2.624}\big[(0.94)(2) + (0.87)(3) + (0.695)(4) + (0.114)(6) + (0.003)(8)\big]\\ &= \mathbf{3.04}\end{aligned}$$

$$\begin{aligned}\hat{\sigma}_1^2 &= \frac{1}{\hat{n}_1}\sum_{i\in\{A,\ldots,E\}} p(1|i)||\bar{x}^{(i)} - \hat{\mu}^{(1)}||^2\\ &= \frac{1}{2.624}\big[(0.94)(2-3.04)^2 + (0.87)(3-3.04)^2 + (0.695)(4-3.04)^2 +\\ &\qquad (0.114)(6-3.04)^2 + (0.003)(8-3.04)^2\big]\\ &= \mathbf{1.04}\end{aligned}$$

Model parameters for Cluster 2:

$$\hat{n}_2 = \sum_{i\in\{A,\ldots,E\}} p(2|i) = \mathbf{2.376}$$

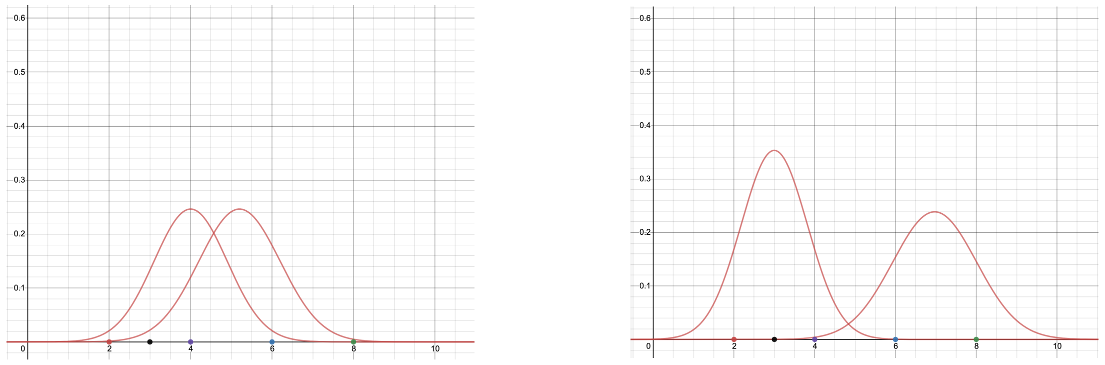
$$\hat{\gamma}_2 = \frac{\hat{n}_2}{n} = \mathbf{0.475}$$

$$\hat{\mu}^{(2)} = \frac{1}{\hat{n}_2}\sum_{i\in\{A,\ldots,E\}} p(2|i)\bar{x}^{(i)} = \mathbf{6.319}$$

$$\hat{\sigma}_2^2 = \frac{1}{\hat{n}_1}\sum_{i\in\{A,\ldots,E\}} p(1|i)||\bar{x}^{(i)} - \hat{\mu}^{(1)}||^2 = \mathbf{2.98}$$

Putting them altogether, we get $\theta^{(1)} = [0.525, 0.475, 3.04, 6.319, 1.04, 2.98]$.

To help visualize the EM algorithm, here are visualizations of the distributions at initialization and the tenth iteration:



(a) Initialization (b) 10th iteration

Figure 1: Visualization of EM algorithm

3 Bayesian Networks

Bayesian networks describe a general framework for learning probability distributions. We use a graphical structure to express the dependencies in probability distributions of interests. These graphs allow us to quickly reason about the model at hand.

We want to learn a probabilistic model that describes our data; in order to do so, we will assume that the data we observe is generated by a set of probability distributions with parameters, and that the probability distributions may depend on each other in some way. To better describe these dependencies, we will draw them as a graph with the following assumption.

Fundamental Assumption of Bayesian Networks: Given its parents, a node x_i is conditionally independent from all other nodes in the graph.

That is, given nodes $x_1, \ldots, x_n$ in a graph, the probability density function over the nodes is: $p(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i | x_{pa_i})$, where x_{pa_i} is the set of parents of node x_i. This set may possibly be empty.

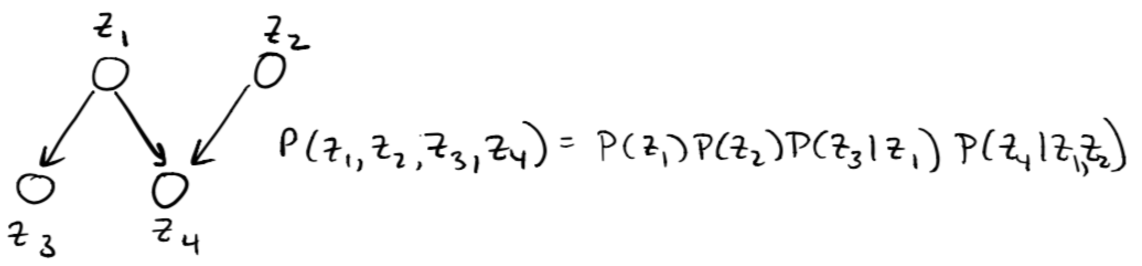

z_1
z_2
z_3
z_4
$P(z_1, z_2, z_3, z_4) = P(z_1)P(z_2)P(z_3|z_1)\,P(z_4|z_1,z_2)$

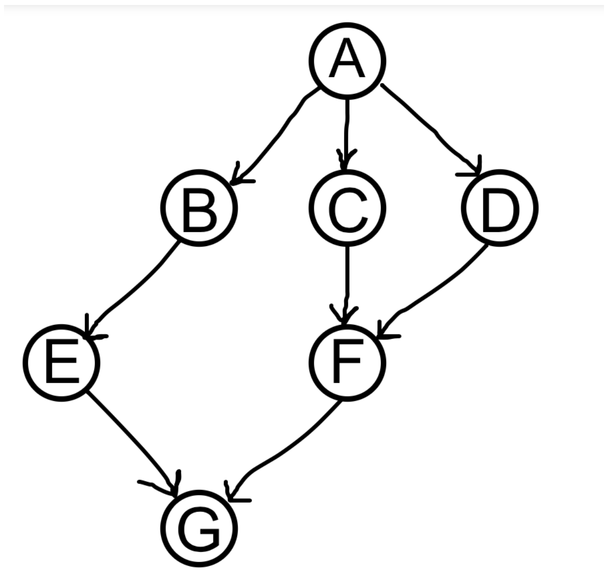
Figure 2: An example Bayesian network from lecture with the associated density function over the nodes.

3.1 Marginal and Conditional Independence

Given that we have a graph with the independence assumption above, it turns out that we can use that graph to deduce independence properties about the distributions.

The two related properties that we care about are **marginal independence** and **conditional independence**. X_1 and X_2 are marginally independent if knowing something about X_2 does not change the probability distribution of X_1, and vice versa. Similarly, X_1 and X_2 are conditionally independent *given* a third node X_3 if knowing X_2 and X_3 does not change our beliefs for X_1.

Discussion Question 6. *What is the probability density function of this graph?*



Solution: $P(A, B, C, D, E, F, G) = P(A)P(B|A)P(C|A)P(D|A)P(E|B)P(F|C, D)P(G|E, F)$