COMPSCI 589
Lecture 17: Mixture Models

Benjamin M. Marlin

College of Information and Computer Sciences
University of Massachusetts Amherst

Slides by Benjamin M. Marlin (marlin@cs.umass.edu).
Created with support from National Science Foundation Award# IIS-1350522.

Outline

Mixture Models

- A mixture model is a probabilistic clustering model that is the unsupervised analogue of the Bayes Optimal Classifier where the unknown assignment of data cases to clusters take the place of the known class labels.

Mixture Models

- A mixture model is a probabilistic clustering model that is the unsupervised analogue of the Bayes Optimal Classifier where the unknown assignment of data cases to clusters take the place of the known class labels.
- We let $\mathbf{x}_i$ be a data case and $z_i \in \{1, ..., K\}$ be the index of the cluster data case i belongs to. z_i is often called the mixture indicator variable or the latent class.

Mixture Models

- A mixture model is a probabilistic clustering model that is the unsupervised analogue of the Bayes Optimal Classifier where the unknown assignment of data cases to clusters take the place of the known class labels.
- We let $\mathbf{x}_i$ be a data case and $z_i \in \{1, ..., K\}$ be the index of the cluster data case i belongs to. z_i is often called the mixture indicator variable or the latent class.
- Each cluster k specifies it's own distribution over the feature vectors $P(\mathbf{X} = \mathbf{x}|Z = k)$

Mixture Models

- A mixture model is a probabilistic clustering model that is the unsupervised analogue of the Bayes Optimal Classifier where the unknown assignment of data cases to clusters take the place of the known class labels.
- We let $\mathbf{x}_i$ be a data case and $z_i \in \{1, ..., K\}$ be the index of the cluster data case i belongs to. z_i is often called the mixture indicator variable or the latent class.
- Each cluster k specifies it's own distribution over the feature vectors $P(\mathbf{X} = \mathbf{x}|Z = k)$
- We also have a discrete distribution $P(Z = k) = \theta_k$, which describes the prior probability that a data case belongs to cluster k.

Data Distribution

- The joint distribution of the features and the mixture indicator variable is:

$$P(\mathbf{X} = \mathbf{x}, Z = k) =$$

Data Distribution

- The joint distribution of the features and the mixture indicator variable is:

$$P(\mathbf{X} = \mathbf{x}, Z = k) = P(\mathbf{X} = \mathbf{x} | Z = k) P(Z = k)$$

Data Distribution

- The joint distribution of the features and the mixture indicator variable is:

$$P(\mathbf{X}=\mathbf{x}, Z=k) = P(\mathbf{X}=\mathbf{x}|Z=k)P(Z=k)$$

- In clustering, we don't know what the right value of the mixture indicator variable is a priori, but we can marginalize it away to obtain a probability distribution on the feature vector only:

$$P(\mathbf{X}=\mathbf{x}) =$$

Data Distribution

- The joint distribution of the features and the mixture indicator variable is:

$$P(\mathbf{X}=\mathbf{x}, Z=k) = P(\mathbf{X}=\mathbf{x}|Z=k)P(Z=k)$$

- In clustering, we don't know what the right value of the mixture indicator variable is a priori, but we can marginalize it away to obtain a probability distribution on the feature vector only:

$$P(\mathbf{X}=\mathbf{x}) = \sum_{k=1}^{K} P(\mathbf{X}=\mathbf{x}|Z=k)P(Z=k)$$

Mixture Component Distributions

To define a specific mixture model, we need to define the form of $P(\mathbf{X} = \mathbf{x}|Z = k)$. Some common choices include:

Mixture Component Distributions

To define a specific mixture model, we need to define the form of $P(\mathbf{X} = \mathbf{x}|Z = k)$. Some common choices include:

- Bernoulli: $\prod_{d=1}^{D} \theta_{dk}^{[x_d=1]}(1 - \theta_{dk})^{[x_d=0]}$

Mixture Component Distributions

To define a specific mixture model, we need to define the form of $P(\mathbf{X} = \mathbf{x}|Z = k)$. Some common choices include:

- Bernoulli: $\prod_{d=1}^{D} \theta_{dk}^{[x_d=1]}(1 - \theta_{dk})^{[x_d=0]}$
- Independent Gaussian: $\prod_{d=1}^{D} \mathcal{N}(x_d; \mu_{dk}, \sigma_{dk}^2)$

Mixture Component Distributions

To define a specific mixture model, we need to define the form of $P(\mathbf{X} = \mathbf{x}|Z = k)$. Some common choices include:

- Bernoulli: $\prod_{d=1}^{D} \theta_{dk}^{[x_d=1]}(1-\theta_{dk})^{[x_d=0]}$
- Independent Gaussian: $\prod_{d=1}^{D} \mathcal{N}(x_d; \mu_{dk}, \sigma_{dk}^2)$
- Multivariate Gaussian: $\mathcal{N}(\mathbf{x}; \mu_k, \Sigma_k)$

Learning

- Given a data set $\mathcal{D} = \{\mathbf{x}_i\}_{i=1:N}$, we can learn the mixture model parameters by maximizing the log probability of the data give the parameters:

$$\mathcal{L} = \sum_{i=1}^{N} \log \left(\sum_{k=1}^{K} P(\mathbf{X}_i = \mathbf{x}_i | Z = k) P(Z = k) \right)$$

Learning

- Given a data set $\mathcal{D} = \{\mathbf{x}_i\}_{i=1:N}$, we can learn the mixture model parameters by maximizing the log probability of the data give the parameters:

$$\mathcal{L} = \sum_{i=1}^{N} \log\left(\sum_{k=1}^{K} P(\mathbf{X}_i = \mathbf{x}_i | Z = k) P(Z = k)\right)$$

- While we can do this directly using gradient-based optimization, it's often faster to use a special algorithm called *Expectation Maximization*.

Expectation Maximization for Gaussian Mixture Models

E-Step: In the first step of the algorithm, we compute the probability that each data case belongs to each cluster using Bayes rule. These probabilities are often called the responsibilities.

$$r_{ik} = P(Z_i = k|\mathbf{x}_i) =$$

Expectation Maximization for Gaussian Mixture Models

E-Step: In the first step of the algorithm, we compute the probability that each data case belongs to each cluster using Bayes rule. These probabilities are often called the responsibilities.

$$r_{ik} = P(Z_i = k|\mathbf{x}_i) = \frac{\theta_k \mathcal{N}(\mathbf{x}; \mu_k, \Sigma_k)}{\sum_{k'=1}^{K} \theta_{k'} \mathcal{N}(\mathbf{x}; \mu_{k'}, \Sigma_{k'})}$$

Expectation Maximization for Gaussian Mixture Models

M-Step: In the second step, we update the parameters using responsibility weighted averages.

Expectation Maximization for Gaussian Mixture Models

M-Step: In the second step, we update the parameters using responsibility weighted averages.

$$\theta_k = \frac{\sum_{i=1}^N r_{ik}}{N}, \qquad \mu_k = \frac{\sum_{i=1}^N r_{ik}\mathbf{x}_i}{\sum_{i=1}^N r_{ik}}$$

Expectation Maximization for Gaussian Mixture Models

M-Step: In the second step, we update the parameters using responsibility weighted averages.

$$\theta_k = \frac{\sum_{i=1}^N r_{ik}}{N}, \qquad \mu_k = \frac{\sum_{i=1}^N r_{ik}\mathbf{x}_i}{\sum_{i=1}^N r_{ik}}$$

$$\Sigma_k = \frac{\sum_{i=1}^N r_{ik}(\mathbf{x}_i - \mu_k)^T(\mathbf{x}_i - \mu_k)}{\sum_{i=1}^N r_{ik}}$$

A Special Case

Suppose we fix $\theta_k = 1/K$ and $\Sigma_k = I$. In this case we have:

A Special Case

Suppose we fix $\theta_k = 1/K$ and $\Sigma_k = I$. In this case we have:

$$\mathcal{N}(\mathbf{x}; \mu_k, \Sigma_k) = \frac{1}{|2\pi I|} \exp(-\frac{1}{2}||\mu_k - \mathbf{x}_i||_2^2)$$

and we obtain the following special case of the EM algorithm for multivariate Gaussians:

A Special Case

Suppose we fix $\theta_k = 1/K$ and $\Sigma_k = I$. In this case we have:

$$\mathcal{N}(\mathbf{x}; \mu_k, \Sigma_k) = \frac{1}{|2\pi I|} \exp(-\frac{1}{2}||\mu_k - \mathbf{x}_i||_2^2)$$

and we obtain the following special case of the EM algorithm for multivariate Gaussians:

$$r_{ik} = \frac{\exp(-\frac{1}{2}||\mu_k - \mathbf{x}_i||_2^2)}{\sum_{k'=1}^{K} \exp(-\frac{1}{2}||\mu_{k'} - \mathbf{x}_i||_2^2)}$$

A Special Case

Suppose we fix $\theta_k = 1/K$ and $\Sigma_k = I$. In this case we have:

$$\mathcal{N}(\mathbf{x}; \mu_k, \Sigma_k) = \frac{1}{|2\pi I|}\exp(-\frac{1}{2}||\mu_k - \mathbf{x}_i||_2^2)$$

and we obtain the following special case of the EM algorithm for multivariate Gaussians:

$$r_{ik} = \frac{\exp(-\frac{1}{2}||\mu_k - \mathbf{x}_i||_2^2)}{\sum_{k'=1}^{K}\exp(-\frac{1}{2}||\mu_{k'} - \mathbf{x}_i||_2^2)}$$

$$\mu_k = \frac{\sum_{i=1}^{N} r_{ik}\mathbf{x}_i}{\sum_{i=1}^{N} r_{ik}}$$

A Special Case

Suppose we fix $\theta_k = 1/K$ and $\Sigma_k = I$. In this case we have:

$$\mathcal{N}(\mathbf{x}; \mu_k, \Sigma_k) = \frac{1}{|2\pi I|} \exp(-\frac{1}{2}||\mu_k - \mathbf{x}_i||_2^2)$$

and we obtain the following special case of the EM algorithm for multivariate Gaussians:

$$r_{ik} = \frac{\exp(-\frac{1}{2}||\mu_k - \mathbf{x}_i||_2^2)}{\sum_{k'=1}^{K} \exp(-\frac{1}{2}||\mu_{k'} - \mathbf{x}_i||_2^2)}$$

$$\mu_k = \frac{\sum_{i=1}^{N} r_{ik}\mathbf{x}_i}{\sum_{i=1}^{N} r_{ik}}$$

This is often referred to as soft K-means.

Trade-Offs

- We can see that the original K-Means algorithm performs hard assignments during clustering, and implicitly assumes all clusters will have an equal number of points assigned as well as a unit covariance matrix.

Trade-Offs

- We can see that the original K-Means algorithm performs hard assignments during clustering, and implicitly assumes all clusters will have an equal number of points assigned as well as a unit covariance matrix.
- EM for Mixtures of Gaussians relaxes all of these assumptions. The objective still has multiple local optima, but EM also produces a guaranteed non-decreasing sequence of objective function values.

Trade-Offs

- We can see that the original K-Means algorithm performs hard assignments during clustering, and implicitly assumes all clusters will have an equal number of points assigned as well as a unit covariance matrix.
- EM for Mixtures of Gaussians relaxes all of these assumptions. The objective still has multiple local optima, but EM also produces a guaranteed non-decreasing sequence of objective function values.
- EM can also be used with any component densities/distributions to customize the model to a given data set.

Trade-Offs

- We can see that the original K-Means algorithm performs hard assignments during clustering, and implicitly assumes all clusters will have an equal number of points assigned as well as a unit covariance matrix.
- EM for Mixtures of Gaussians relaxes all of these assumptions. The objective still has multiple local optima, but EM also produces a guaranteed non-decreasing sequence of objective function values.
- EM can also be used with any component densities/distributions to customize the model to a given data set.
- As with K-Means, initialization is important, but the same heuristics can be applied. There are similar issues with interpreting output and selecting K.

Choosing K

- The Elbow Method: Simple, only requires one fit per value of K. Requires manual assessment of plot. Works for K-Means and Mixture Models.

Choosing K

- The Elbow Method: Simple, only requires one fit per value of K. Requires manual assessment of plot. Works for K-Means and Mixture Models.
- Cross-validation: Requires multiple fits per value of K. Automatic selection of best K. Works for GMMs, but often fails for K-Means.