Unsupervised Learning: Gaussian Mixture Models & Expectation Maximization

These slides are partially based on slides assembled by Eric Eaton, with grateful acknowledgement of the many others who made their course materials freely available online.
Soft Clustering

• Clustering typically assumes that each instance is given a “hard” assignment to exactly one cluster.

• Does not allow uncertainty in class membership or for an instance to belong to more than one cluster.

• *Soft clustering* gives probabilities that an instance belongs to each of a set of clusters.

• Each instance is assigned a probability distribution across a set of discovered categories (probabilities of all categories must sum to 1).
Gaussian Mixture Models

• Recall the Gaussian distribution:

\[
P(x | \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^d|\Sigma|}} \exp \left(-\frac{1}{2}(x - \mu)^\top \Sigma^{-1} (x - \mu)\right)
\]
The GMM assumption

- There are k components. The $i$’th component is called $\omega_i$.
- Component $\omega_i$ has an associated mean vector $\mu_i$. 

![Diagram showing three Gaussian components with their means $\mu_1$, $\mu_2$, and $\mu_3$.](image)
The GMM assumption

- There are \( k \) components. The \( i \)’ th component is called \( \omega_i \)
- Component \( \omega_i \) has an associated mean vector \( \mu_i \)
- Each component generates data from a Gaussian with mean \( \mu_i \) and covariance matrix \( \sigma^2 I \)

Assume that each datapoint is generated according to the following recipe:
The GMM assumption

- There are k components. The i’th component is called $\omega_i$.
- Component $\omega_i$ has an associated mean vector $\mu_i$.
- Each component generates data from a Gaussian with mean $\mu_i$ and covariance matrix $\sigma^2 I$.

Assume that each datapoint is generated according to the following recipe:

1. Pick a component at random. Choose component i with probability $P(\omega_i)$. 

Copyright © 2001, 2004, Andrew W. Moore
The GMM assumption

- There are $k$ components. The $i$'th component is called $\omega_i$.
- Component $\omega_i$ has an associated mean vector $\mu_i$.
- Each component generates data from a Gaussian with mean $\mu_i$ and covariance matrix $\sigma^2 I$.

Assume that each datapoint is generated according to the following recipe:

1. Pick a component at random. Choose component $i$ with probability $P(\omega_i)$.
2. Datapoint $\sim N(\mu_i, \sigma^2 I)$. 

Copyright © 2001, 2004, Andrew W. Moore
The **General** GMM assumption

- There are $k$ components. The $i$’th component is called $\omega_i$.
- Component $\omega_i$ has an associated mean vector $\mu_i$.
- Each component generates data from a Gaussian with mean $\mu_i$ and covariance matrix $\Sigma_i$.

Assume that each datapoint is generated according to the following recipe:

1. Pick a component at random. Choose component $i$ with probability $P(\omega_i)$.
2. Datapoint $\sim N(\mu_i, \Sigma_i)$.
Mixture Models

• Formally a Mixture Model is the weighted sum of a number of pdfs where the weights are determined by a distribution, $\pi$

$$p(x) = \sum_{i=0}^{k} \pi_i f_i(x)$$

where $\sum_{i=0}^{k} \pi_i = 1$
Gaussian Mixture Models

- GMM: the weighted sum of a number of Gaussians where the weights are determined by a distribution, $\pi$

$$p(x) = \pi_0 N(x|\mu_0, \Sigma_0) + \pi_1 N(x|\mu_1, \Sigma_1) + \ldots + \pi_k N(x|\mu_k, \Sigma_k)$$

where $\sum_{i=0}^{k} \pi_i = 1$

$$p(x) = \sum_{i=0}^{k} \pi_i N(x|\mu_k, \Sigma_k)$$
Expectation-Maximization for GMMs

Iterate until convergence:

On the \( t' \) th iteration let our estimates be

\[
\lambda_t = \{ \mu_1(t), \mu_2(t) \ldots \mu_c(t) \}
\]

E-step: Compute “expected” classes of all datapoints for each class

\[
P(w_i | x_k, \lambda_t) = \frac{p(x_k | w_i, \lambda_t)P(w_i | \lambda_t)}{p(x_k | \lambda_t)} = \frac{p(x_k | w_i, \mu_i(t), \sigma^2 I)p_i(t)}{\sum_{j=1}^{c}p(x_k | w_j, \mu_j(t), \sigma^2 I)p_j(t)}
\]

M-step: Estimate \( \mu \) given our data’s class membership distributions

\[
\mu_i(t+1) = \frac{\sum_k P(w_i | x_k, \lambda_t)x_k}{\sum_k P(w_i | x_k, \lambda_t)}
\]

Copyright © 2001, 2004, Andrew W. Moore
E.M. for General GMMs

Iterate. On the $t'$ th iteration let our estimates be

$$\lambda_t = \{ \mu_1(t), \mu_2(t) \ldots \mu_c(t), \Sigma_1(t), \Sigma_2(t) \ldots \Sigma_c(t), p_1(t), p_2(t) \ldots p_c(t) \}$$

E-step: Compute “expected” clusters of all datapoints

$$P(w_i | x_k, \lambda_t) = \frac{p(x_k | w_i, \lambda_t) P(w_i | \lambda_t)}{p(x_k | \lambda_t)} = \frac{p(x_k | w_i, \mu_i(t), \Sigma_i(t)) p_i(t)}{\sum_{j=1}^{c} p(x_k | w_j, \mu_j(t), \Sigma_j(t)) p_j(t)}$$

M-step: Estimate $\mu, \Sigma$ given our data’s class membership distributions

$$\mu_i(t+1) = \frac{\sum_k P(w_i | x_k, \lambda_t) x_k}{\sum_k P(w_i | x_k, \lambda_t)}$$

$$\Sigma_i(t+1) = \frac{\sum_k P(w_i | x_k, \lambda_t) [x_k - \mu_i(t+1)] [x_k - \mu_i(t+1)]^T}{\sum_k P(w_i | x_k, \lambda_t)}$$

$$p_i(t+1) = \frac{\sum_k P(w_i | x_k, \lambda_t)}{R}$$

$R = \#\text{records}$

$p_i(t)$ is shorthand for estimate of $P(\omega_i)$ on $t'$ th iteration

Just evaluate a Gaussian at $x_k$
Gaussian Mixture Example: Start
After first iteration
After 2nd iteration
After 3rd iteration
After 4th iteration
After 5th iteration
After 6th iteration
After 20th iteration
Some Bio Assay data
GMM clustering of the assay data
Resulting Density Estimator
Closing Thoughts

• GMMs are a "soft" clustering algorithm, that can be learned using EM.

• If you keep iterating EM, you will converge, but only a local optimum.

• You will see EM in other contexts as well, when doing inference with graphical models is hard – like Hidden Markov Models.
$\overline{X_n}$

$K = 3$

90% red
80% blue
2% green
$\{p_i\}$ → "prior"

$\{m_i\}$ → "mean"

$\{\Sigma_i\}$ → "variance"

$w_1$, $w_2$, $w_3$ with $p_1 = 0.1$, $p_2 = 0.8$, $p_3 = 0.1$
Coin 1
\[ P(H) = 0.5 \]
\[ P(\text{flip}) = 0.3 \]

Coin 2
\[ P(H) = 0.3 \]
\[ 0.2 \]

Coin 3
\[ P(H) = 0.7 \]
\[ 0.4 \]

Coin 4
\[ P(H) = 1 \]
\[ 0.1 \]

\[ P(H, \text{coin 2}) = 0.2 \cdot 0.3 = 0.06 \]
\[ \ell_p \] \[ \{ \mu_i \} \] \[ \{ \Sigma_i \} \] \[ N \text{ data points} \] \[ \{ x_i \} \]

\[ p(\text{data model}) = P_0(x_1, x_2, \ldots, x_N) = \frac{1}{\prod_{i=1}^{n} P(x_i)} \]

\[ = \frac{1}{\prod_{i=1}^{n} \sum_{k=1}^{K} P_k N(x_i | \mu_k, \Sigma_k)} \]

\[ \ln p = \sum_{i=1}^{n} \ln \left( \sum_{k=1}^{K} P_k N(x_i | \mu_k, \Sigma_k) \right) \]
1) "Colour" the data $\leftrightarrow$ E step
2) Move the means $\leftrightarrow$ M step

$p(w_i | x_k) = \frac{p(x_k | w_i) p(w_i)}{\sum_i p(x_k | w_i) p(w_i)}$

$N(x_k | \mu_i, \Sigma_i)$ prior $p_i$
Clustering with Gaussian Mixtures: Slide 30

\[ P(w_i | x_K) \]