CSCE 633: Machine Learning

Lecture 24: Unsupervised Learning: EM

Texas A&M University

10-18-19

Last Time

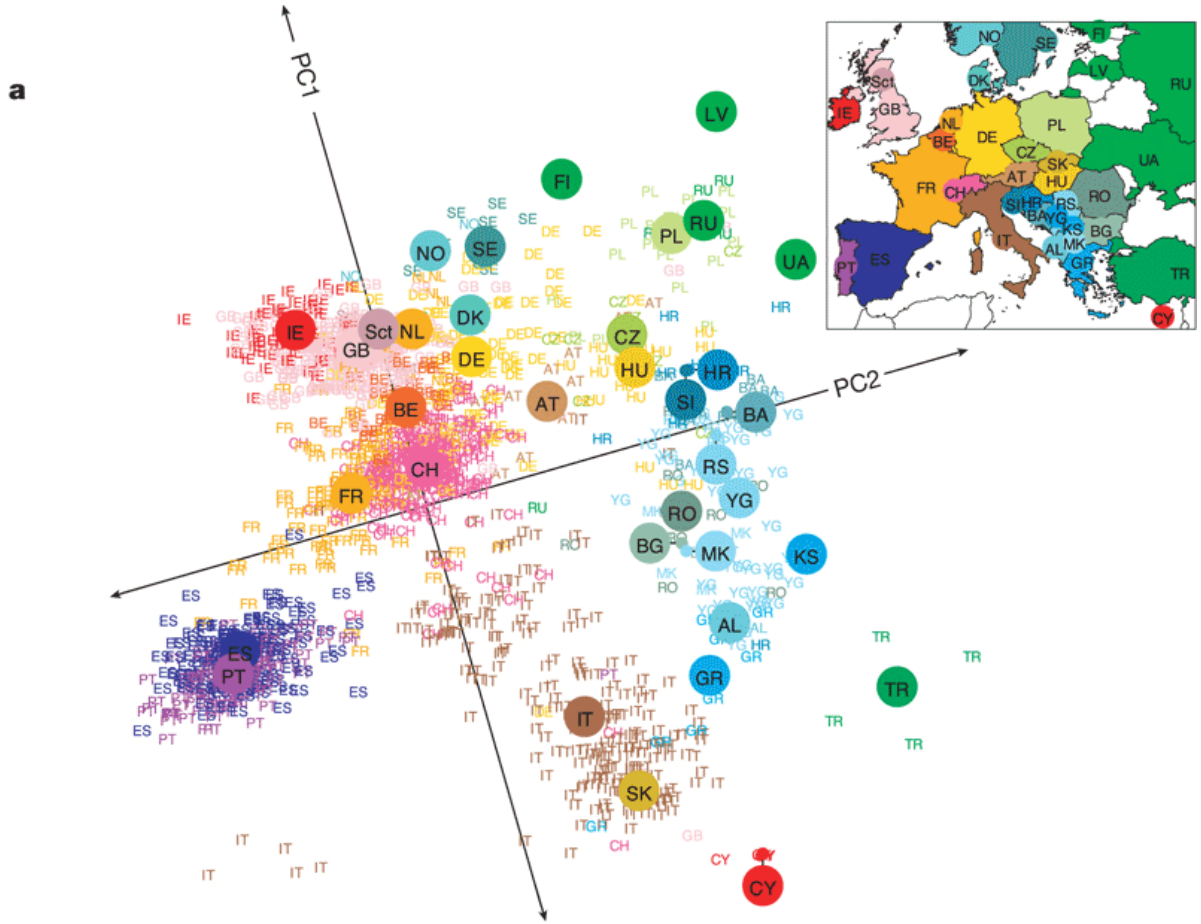
- PCA
- Clustering, K-Means

Goals of this lecture

- Expectation Maximization

Clustering

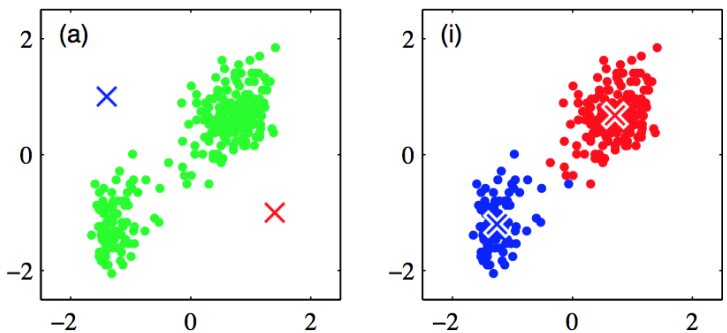
Finding patterns/structure/sub-populations in data



K-means Clustering

Representation

- Input: Data $\mathcal{D} = \{\mathbf{x_1}, \ldots, \mathbf{x_N}\}$
- Output: Clusters $\boldsymbol{\mu_1}, \ldots, \boldsymbol{\mu_K}$
- Decision: Cluster membership, the cluster id assigned to sample $\mathbf{x_n}$, i.e. $A(\mathbf{x_n}) \in \{1, \ldots, K\}$
- Evaluation metric: Distortion measure
  $J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\mathbf{x_n} - \boldsymbol{\mu_k}\|_2^2$, where $r_{nk} = 1$ if $A(\mathbf{x_n}) = k$, 0 otherwise
- Intuition: Data points assigned to cluster k should be close to centroid $\boldsymbol{\mu_k}$



K-means Clustering

Evaluation metric: $\min_{r_{nk}} J = \min_{r_{nk}} \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\mathbf{x_n} - \boldsymbol{\mu_k}\|_2^2$

Optimization:

- Step 0: Initialize $\boldsymbol{\mu_k}$ to some values
- Step 1: Assume the current value of $\boldsymbol{\mu_k}$ fixed, minimize J over r_{nk}, which leads to the following cluster assignment rule
  $r_{nk} = \begin{cases} 1, & \text{if } k = arg\min_j \|\mathbf{x_n} - \boldsymbol{\mu_j}\|_2^2 \\ 0, & \text{otherwise} \end{cases}$
- Step 2: Assume the current value of r_{nk} fixed, minimize J over $\boldsymbol{\mu_k}$, which leads to the following rule to update the prototypes of the clusters $\boldsymbol{\mu_k} = \frac{\sum_n r_{nk}\mathbf{x_n}}{\sum_n r_{nk}}$
- Step 3: Determine whether to stop or return to Step 1

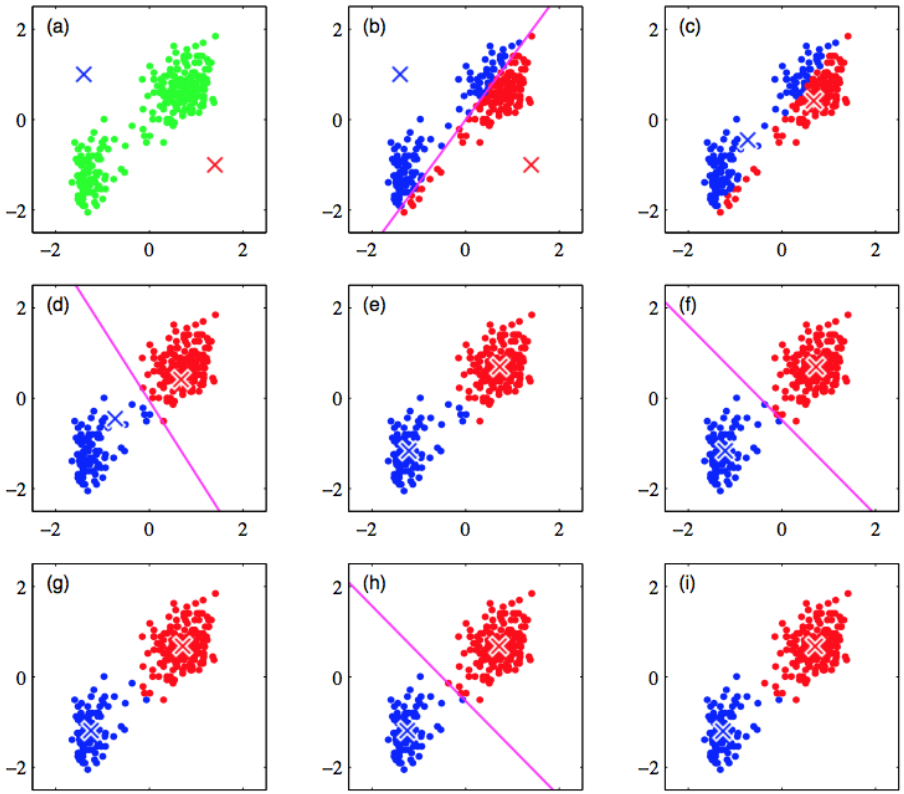
K-means Clustering

Remarks

- The centroid $\boldsymbol{\mu_k}$ is the means of data points assigned to the cluster k, hence the name K-means clustering.
- The procedure terminates after a finite number of steps, as the procedure reduces J in both Step 1 and Step 2
- There is no guarantee the procedure terminates at the global optimum of J. In most cases, the algorithm stops at a local optimum, which depends on the initial values in Step 0 $\rightarrow$ random restarts to improve chances of getting closer to global optima

K-means Clustering

Example



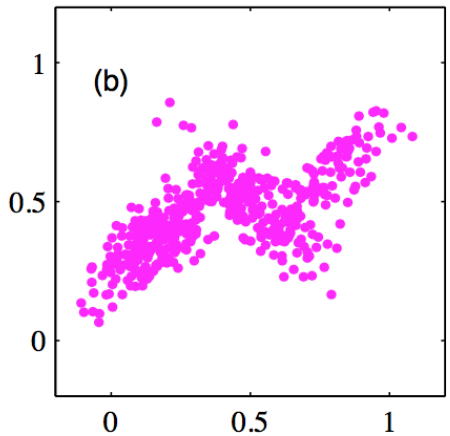
K-means Clustering

Application: vector quantization

- We can replace our data points with the centroids $\boldsymbol{\mu_k}$ from the clusters they are assigned to $\rightarrow$ vector quantization
- We have compressed the data points into
  - a codebook of all the centroids $\{\boldsymbol{\mu_1}, \ldots, \boldsymbol{\mu_K}\}$
  - a list of indices to the codebook for the data points (created based on r_{nk})
- This compression is obviously lossy as certain information will be lost if we use a very small K

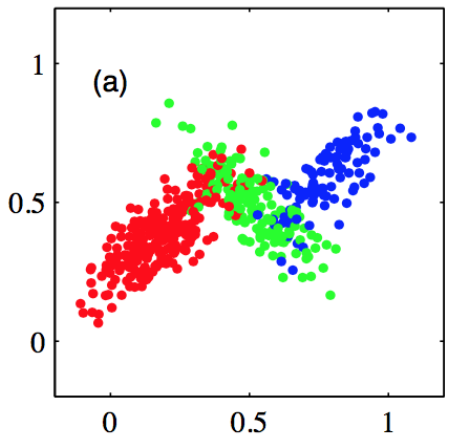
Probabilistic interpretation of clustering

- We want to find $p(\mathbf{x})$ that best describes our data
- The data points seem to form 3 clusters
- However, we cannot model $p(\mathbf{x})$ with simple and known distributions, e.g. one Gaussian



Probabilistic interpretation of clustering

- Instead, we will model each region with a Gaussian distribution $\rightarrow$ Gaussian mixture models (GMMs)
- Question 1: How do we know which (color) region a data point comes from?
- Question 2: What are the parameters of Gaussian distributions in each region?
- We will answer both in an unsupervised way from data $\mathcal{D} = \{\mathbf{x_1}, \ldots, \mathbf{x_n}\}$



Gaussian mixture models: formal definition

A Gaussian mixture model has the following density function for $\mathbf{x}$

$$p(\mathbf{x}) = \sum_{k=1}^{K} \omega_k \mathcal{N}(\mathbf{x}; \boldsymbol{\mu_k}, \boldsymbol{\Sigma_k})$$

- K: number of Gaussians
- $\boldsymbol{\mu_k}$,$\boldsymbol{\Sigma_k}$: mean & covariance of k^{th} component
- ω_k: component weights, how much component k contributes to the final distribution

$$\omega_k > 0\ ,\quad \forall k \quad \text{and} \quad \sum_{k=1}^{K} \omega_k = 1$$

ω_k can be represented by the prior distribution: $\omega_k = p(z = k)$, which decides which mixture to use

GMM as the marginal distribution of a joint distribution

- Consider the following joint distribution $p(\mathbf{x}, z) = p(z)p(\mathbf{x}|z)$
- z is a discrete random variable taking values between 1 and K, "selects" a Gaussian component
- We denote $\omega_k = p(z = k)$
- Assume Gaussian conditional distributions $p(\mathbf{x}|z = k) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu_k}, \boldsymbol{\Sigma_k})$
- Then the marginal distribution of $\mathbf{x}$ is

$$p(\mathbf{x}) = \sum_{k=1}^{K} \omega_k \mathcal{N}(\mathbf{x}; \boldsymbol{\mu_k}, \boldsymbol{\Sigma_k})$$

(which is the Gaussian mixture model)

GMM as the marginal distribution of a joint distribution

- The joint distribution between $\mathbf{x}$ and z (representing color) are

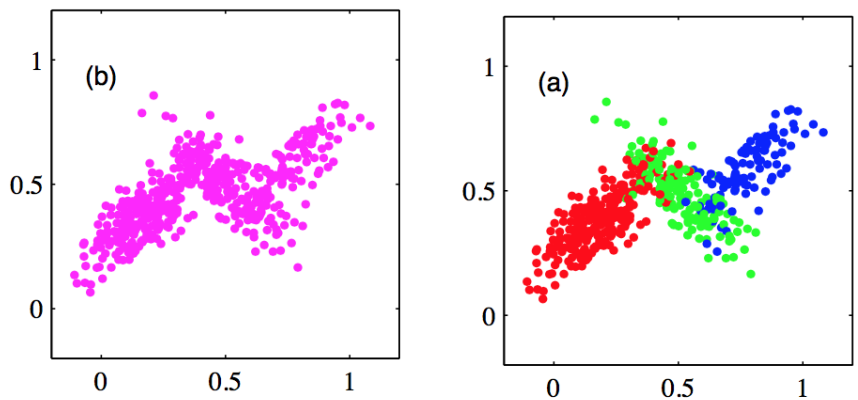
$$p(\mathbf{x}|z = red) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu_1}, \boldsymbol{\Sigma_1})$$

$$p(\mathbf{x}|z = blue) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu_2}, \boldsymbol{\Sigma_2})$$

$$p(\mathbf{x}|z = green) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu_3}, \boldsymbol{\Sigma_3})$$

- The marginal distribution is thus

$$\begin{aligned} p(\mathbf{x}) =& p(red)\mathcal{N}(\mathbf{x}; \boldsymbol{\mu_1}, \boldsymbol{\Sigma_1}) + p(blue)\mathcal{N}(\mathbf{x}; \boldsymbol{\mu_2}, \boldsymbol{\Sigma_2}) \\ &+ p(green)\mathcal{N}(\mathbf{x}; \boldsymbol{\mu_3}, \boldsymbol{\Sigma_3}) \end{aligned}$$



Parameter estimation for GMMs: the easy case with complete data

We know the component in which each sample belongs to

- Data $\mathcal{D} = \{(\mathbf{x_1}, z_1), \ldots, (\mathbf{x_N}, z_N)\}$
- We want to find $\boldsymbol{\theta} = \{\boldsymbol{\mu_k}, \boldsymbol{\Sigma_k}, \omega_k\}$
- Maximum log-likelihood: $\boldsymbol{\theta}^* = arg\max \log(p(\mathcal{D})|\boldsymbol{\theta})$

Solution

$$\omega_k = \frac{\sum_n \gamma_{nk}}{\sum_k \sum_n \gamma_{nk}} \qquad \boldsymbol{\mu_k} = \frac{1}{\sum_n \gamma_{nk}} \sum_n \gamma_{nk} \mathbf{x_n}$$

$$\boldsymbol{\Sigma_k} = \frac{1}{\sum_n \gamma_{nk}} \sum_n \gamma_{nk} (\mathbf{x_n} - \boldsymbol{\mu_k})(\mathbf{x_n} - \boldsymbol{\mu_k})^T$$

where $\gamma_{nk} = 1$ if $z_n = k$

Parameter estimation for GMMs: the easy case with complete data

Understanding the intuition

$$\omega_k = \frac{\sum_n \gamma_{nk}}{\sum_k \sum_n \gamma_{nk}} \qquad \boldsymbol{\mu_k} = \frac{1}{\sum_n \gamma_{nk}} \sum_n \gamma_{nk} \mathbf{x_n}$$

$$\boldsymbol{\Sigma_k} = \frac{1}{\sum_n \gamma_{nk}} \sum_n \gamma_{nk} (\mathbf{x_n} - \boldsymbol{\mu_k})(\mathbf{x_n} - \boldsymbol{\mu_k})^T$$

- For ω_k: count the number of data points whose z_n is k and divide by the total number of data points
- For μ_k: get all the data points whose z_n is k, compute their mean
- For Σ_k: get all the data points whose z_n is k, compute their covariance

Parameter estimation for GMMs: incomplete data

Trick: estimation with soft γ_{nk}

$$\gamma_{nk} = p(z_n = k|\mathbf{x_n})$$

$$\omega_k = \frac{\sum_n \gamma_{nk}}{\sum_k \sum_n \gamma_{nk}}$$

$$\boldsymbol{\mu_k} = \frac{1}{\sum_n \gamma_{nk}} \sum_n \gamma_{nk} \mathbf{x_n}$$

$$\boldsymbol{\Sigma_k} = \frac{1}{\sum_n \gamma_{nk}} \sum_n \gamma_{nk} (\mathbf{x_n} - \boldsymbol{\mu_k})(\mathbf{x_n} - \boldsymbol{\mu_k})^T$$

Every data point $\mathbf{x_n}$ is assigned to a component fractionally according to $p(z_n = k|\mathbf{x_n})$, also called responsibility

Parameter estimation for GMMs: incomplete data

Trick: estimation with soft γ_{nk}

Since we do not know θ to begin with, we cannot compute the soft γ_{nk}

But we can invoke an iterative procedure and alternate between estimating γ_{nk} and using the estimated γ_{nk} to compute $\boldsymbol{\theta} = \{\boldsymbol{\mu_k}, \boldsymbol{\Sigma_k}, \omega_k\}$

- Step 0: guess $\boldsymbol{\theta}$ with initial values
- Step 1: compute γ_{nk} using current $\boldsymbol{\theta}$
- Step 2: update $\boldsymbol{\theta}$ using computed γ_{nk}
- Step 3: go back to Step 1

Questions: i) is this procedure correct, for example, optimizing a sensible criterion? ii) practically, will this procedure ever stop instead of iterating forever? The answer lies in the Expectation Maximization (EM) algorithm, a powerful procedure for model estimation with unknown data

Expectation Maximization: motivation and setup

- EM is used to estimate parameters for probabilistic models with hidden/latent variables

$$p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{z} p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})$$

  where $\mathbf{x}$ is the observed random variable and $\boldsymbol{\theta}$ is the hidden
- We are given data $\mathcal{D} = \{\mathbf{x_1}, \ldots, \mathbf{x_N}\}$, where the corresponding hidden variable values z are not included
- Our goal is to obtain the maximum likelihood estimate of $\boldsymbol{\theta}$

$$\boldsymbol{\theta}^* = arg\max_{\boldsymbol{\theta}} \sum_{n=1}^{N} \log p(\mathbf{x_n}|\boldsymbol{\theta}) = arg\max_{\boldsymbol{\theta}} \sum_{n=1}^{N} \log \sum_{\mathbf{z_n}} p(\mathbf{x_n}, \mathbf{z_n}|\boldsymbol{\theta})$$

The above objective function is called incomplete log-likelihood and it is computationally intractable (since it needs to sum over all possible values of $\mathbf{z}$ and then take the logarithm)

Expectation Maximization: Complete log-likelihood

- Incomplete log-likelihood

$$l(\boldsymbol{\theta}) = \sum_{n=1}^{N} \log \sum_{\mathbf{z_n}} p(\mathbf{x_n}, \mathbf{z_n}|\boldsymbol{\theta})$$

- EM uses a clever trick to change this into sum-log form by defining:

$$\begin{aligned} Q_q(\boldsymbol{\theta}) &= \sum_{n=1}^{N} \log \mathbb{E}_{\mathbf{z_n} \sim q(\mathbf{z_n})} p(\mathbf{x_n}, \mathbf{z_n}|\boldsymbol{\theta}) \\ &= \sum_{n=1}^{N} \sum_{\mathbf{z_n}} q(\mathbf{z_n}) \log p(\mathbf{x_n}, \mathbf{z_n}|\boldsymbol{\theta}) \end{aligned}$$

  where $q(\mathbf{z})$ is a distribution over z

The above is called expected (complete) log-likelihood, since it takes the form of log-sum which is computationally tractable

Expectation Maximization: Choice of $q(\mathbf{z})$

- We will choose a special $q(\mathbf{z}) = p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})$, i.e., the posterior probability of z

$$Q(\boldsymbol{\theta}) = Q_{\mathbf{z} \sim p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})}(\boldsymbol{\theta})$$

- We will show that

$$l(\boldsymbol{\theta}) = Q(\boldsymbol{\theta}) + \sum_{n=1}^{N} \mathbb{H}\left[p(\mathbf{z}|\mathbf{x_n}, \boldsymbol{\theta})\right]$$

where $\mathbb{H}$ is the entropy of the probabilistic distribution $p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})$

$$\mathbb{H}\left[p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})\right] = -\sum_{\mathbf{z_n}} p(\mathbf{z_n}|\mathbf{x}, \boldsymbol{\theta}) \log p(\mathbf{z_n}|\mathbf{x}, \boldsymbol{\theta})$$

Expectation Maximization: Choice of $q(z)$

Proof

$$Q_q(\theta) = \sum_n \sum_{z_n} q(z_n) \log p(x_n, z_n|\theta)$$

$$\begin{aligned}
Q(\theta) = Q_{z\sim p(z|x,\theta)}(\theta) &= \sum_n \sum_{z_n} p(z_n|x_n,\theta) \log p(x_n, z_n|\theta) \\
&= \sum_n \sum_{z_n} p(z_n|x_n,\theta)[\log p(x_n|\theta) + \log p(z_n|x_n,\theta)] \\
&= \sum_n \sum_{z_n} p(z_n|x_n,\theta) \log p(x_n|\theta) + \sum_n \sum_{z_n} p(z_n|x_n,\theta) \log p(z_n|x_n,\theta) \\
&= \sum_n \log p(x_n|\theta) \sum_{z_n} p(z_n|x_n,\theta) - \mathbb{H}\left[p(z|x_n,\theta)\right] \\
&= l(\theta) - \mathbb{H}\left[p(z|x_n,\theta)\right]
\end{aligned}$$

Expectation Maximization

- As before, $Q(\boldsymbol{\theta})$ cannot be computed, as $p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})$ depends on the unknown parameter values $\boldsymbol{\theta}$
- Instead, we will use a known value $\boldsymbol{\theta}^{old}$ to compute the expected likelihood

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{old}) = \sum_{n} \sum_{\mathbf{z_n}} p(\mathbf{z_n}|\mathbf{x_n}, \boldsymbol{\theta}^{old}) \log p(\mathbf{x_n}, \mathbf{z_n}|\boldsymbol{\theta})$$

- In the above, the variable is $\boldsymbol{\theta}$, while $\boldsymbol{\theta}^{old}$ is known
- By its definition $Q(\boldsymbol{\theta}) = Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{old})$
- But how does $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{old})$ relates to $l(\boldsymbol{\theta})$?
- We will show that

$$l(\boldsymbol{\theta}) \geq Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{old}) + \sum_{n} \mathbb{H}\left[p(\mathbf{z}|\mathbf{x_n}, \boldsymbol{\theta}^{old})\right]$$

- Thus, $Q(\boldsymbol{\theta})$ is better than $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{old})$, except that we cannot compute the former

Proof

$$
\begin{aligned}
l(\boldsymbol{\theta}) &= \sum_{n} \log p(\mathbf{x_n}|\boldsymbol{\theta}) = \sum_{n} \log \sum_{\mathbf{z_n}} p(\mathbf{x_n}, \mathbf{z_n}|\boldsymbol{\theta}) \\
&= \sum_{n} \log \sum_{\mathbf{z_n}} p(\mathbf{z}|\mathbf{x_n}, \boldsymbol{\theta}^{\boldsymbol{old}}) \frac{p(\mathbf{x_n}, \mathbf{z_n}|\boldsymbol{\theta})}{p(\mathbf{z}|\mathbf{x_n}, \boldsymbol{\theta}^{\boldsymbol{old}})} \\
&\geq \sum_{n} \sum_{\mathbf{z_n}} p(\mathbf{z}|\mathbf{x_n}, \boldsymbol{\theta}^{\boldsymbol{old}}) \log p(\mathbf{x_n}, \mathbf{z_n}|\boldsymbol{\theta}) \\
&\qquad - \sum_{n} \sum_{\mathbf{z_n}} p(\mathbf{z}|\mathbf{x_n}, \boldsymbol{\theta}^{\boldsymbol{old}}) \log p(\mathbf{z}|\mathbf{x_n}, \boldsymbol{\theta}^{\boldsymbol{old}}) \\
&= \underbrace{Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\boldsymbol{old}}) + \sum_{n} \mathbb{H}\left[p(\mathbf{z}|\mathbf{x_n}, \boldsymbol{\theta}^{\boldsymbol{old}})\right]}_{A(\boldsymbol{\theta}, \boldsymbol{\theta}^{\boldsymbol{old}}) \text{ : auxiliary function}}
\end{aligned}
$$

In the above we have used that $\log \sum_i w_i x_i \geq \sum_i w_i \log x_i$, $\forall w_i \geq 0$ s.t. $\sum_i w_i = 1$

Expectation Maximization

- Suppose we have an initial guess $\boldsymbol{\theta^{old}}$ and we maximize the auxiliary function

$$\boldsymbol{\theta^{new}} = arg \max_{\boldsymbol{\theta}} A(\boldsymbol{\theta}, \boldsymbol{\theta^{old}})$$

- With the new guess, we have

$$l(\boldsymbol{\theta^{new}}) \geq A(\boldsymbol{\theta^{new}}, \boldsymbol{\theta^{old}}) \geq A(\boldsymbol{\theta^{new}}, \boldsymbol{\theta^{old}}) = l(\boldsymbol{\theta^{old}})$$

- By maximizing the auxiliary function, we will keep increasing the likelihood (core of EM)

Expectation Maximization

Algorithm outline

- Step 0: Initialize $\boldsymbol{\theta}$ with $\boldsymbol{\theta^{(0)}}$
- Step 1 (E-step): Compute the auxiliary function using the current value of $\boldsymbol{\theta}$

$$A(\boldsymbol{\theta}, \boldsymbol{\theta^{(t)}})$$

- Step 2 (M-step): Maximize the auxiliary function

$$\boldsymbol{\theta^{(t+1)}} \leftarrow arg\max A(\boldsymbol{\theta}, \boldsymbol{\theta^{(t)}})$$

- Step 3: Increase t to $t+1$ and back to Step 1; or stop if $l(\boldsymbol{\theta}^{(t+1)})$ does not improve much

Remarks

- EM converges but only to a local optimum: global optimum is not guaranteed
- The E-step depends on computing the posterior probability $p(\mathbf{z_n}|\mathbf{x_n}, \boldsymbol{\theta}^{(t)})$
- The M-step does not depend on the entropy term, so we need only to do the following

$$\boldsymbol{\theta}^{(t+1)} \leftarrow arg\max A(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)}) = arg\max Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)})$$

We often call the last term Q-function

Expectation Maximization for GMMs

- Hidden variable follows Multinomial distribution
  $\mathbf{z_n} \sim \text{Multinomial}(\omega_1, \ldots, \omega_K)$
- The likelihood of observation $\mathbf{x_n}$ equals the probability of corresponding component $p(\mathbf{x_n}|\mathbf{z_n}, \boldsymbol{\theta}) = \prod_{k=1}^{K} \mathcal{N}(\mathbf{x_n}|\boldsymbol{\mu_k}, \boldsymbol{\Sigma_k})^{\gamma_{nk}}$
- Complete data log-likelihood

$$
\begin{aligned}
Q(\boldsymbol{\theta}) &= \mathbb{E}\left[\sum_n \log p(\mathbf{x_n}, \mathbf{z_n}|\boldsymbol{\theta})\right] = \mathbb{E}\left[\sum_n \log(p(\mathbf{z_n}|\boldsymbol{\theta})p(\mathbf{x_n}|\mathbf{z_n}, \boldsymbol{\theta}))\right] \\
&= \mathbb{E}\left[\sum_n \log\left(\prod_k \omega_k^{\gamma_{nk}} \prod_k \mathcal{N}(\mathbf{x_n}|\boldsymbol{\mu_k}, \boldsymbol{\Sigma_k})^{\gamma_{nk}}\right)\right] \\
&= \mathbb{E}\left[\sum_n \sum_k \gamma_{nk}\left(\log \omega_k + \log \mathcal{N}(\mathbf{x_n}|\boldsymbol{\mu_k}, \boldsymbol{\Sigma_k})\right)\right] \\
&= \sum_n \sum_k \mathbb{E}[\gamma_{nk}|\mathbf{x_n}, \boldsymbol{\mu_k}, \boldsymbol{\Sigma_k}]\gamma_{nk}\left(\log \omega_k + \log \mathcal{N}(\mathbf{x_n}|\boldsymbol{\mu_k}, \boldsymbol{\Sigma_k})\right)
\end{aligned}
$$

Expectation Maximization for GMMs

What is the E-step in GMM?

$$\mathbb{E}[\gamma_{nk}|\mathbf{x_n}, \boldsymbol{\mu_k}, \boldsymbol{\Sigma_k}] = p(\gamma_{nk} = 1|\mathbf{x_n}, \boldsymbol{\theta_k}) = \frac{p(\mathbf{x_n}|\gamma_{nk} = 1, \boldsymbol{\theta})p(\gamma_{nk} = 1|\boldsymbol{\theta_k})}{\sum_k p(\mathbf{x_n}|\gamma_{nk} = 1, \boldsymbol{\theta})p(\gamma_{nk} = 1|\boldsymbol{\theta_k})}$$

We compute the probability

$$\gamma_{nk} = p(z = k|\mathbf{x_n}, \boldsymbol{\theta}^{(t)})$$

$$\gamma_{nk} = \frac{\omega_k p(\mathbf{x_n}|\boldsymbol{\mu}_\mathbf{k}^{(t-1)}, \boldsymbol{\Sigma}_\mathbf{k}^{(t-1)})}{\sum_k \omega_k p(\mathbf{x_n}|\boldsymbol{\mu}_\mathbf{k}^{(t-1)}, \boldsymbol{\Sigma}_\mathbf{k}^{(t-1)})}$$

Expectation Maximization for GMMs

What is the M-step in GMM?

Maximize the auxiliary function

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)}) = \sum_n \sum_k p(z = k|\mathbf{x_n}, \boldsymbol{\theta}^{(t)}) \log p(\mathbf{x_n}, z = k|\boldsymbol{\theta})$$

$$= \sum_n \sum_k \gamma_{nk} \log p(\mathbf{x_n}, z = k|\boldsymbol{\theta})$$

$$= \sum_n \sum_k \gamma_{nk} \log(p(z = k)p(\mathbf{x_n}|z = k))$$

$$= \sum_n \sum_k \gamma_{nk} [\log \omega_k + \log \mathcal{N}(\mathbf{x_n}|\boldsymbol{\mu_k}, \boldsymbol{\Sigma_k})]$$

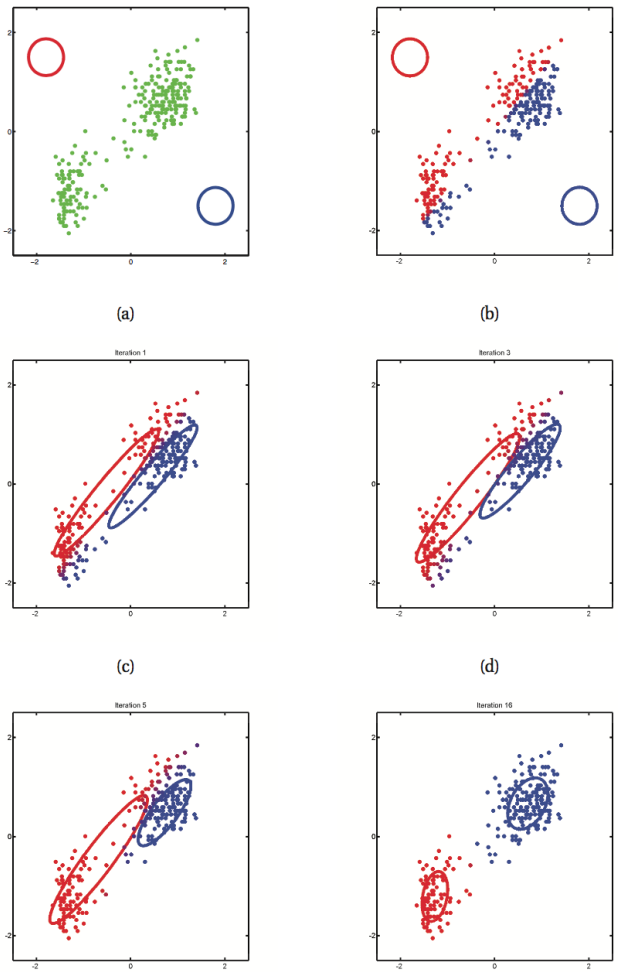
$$\Rightarrow \omega_k = \frac{\sum_n \gamma_{nk}}{N}$$

$$\Rightarrow \boldsymbol{\mu_k} = \frac{1}{\sum_n \gamma_{nk}} \sum_n \gamma_{nk} \mathbf{x_n}$$

$$\Rightarrow \boldsymbol{\Sigma_k} = \frac{1}{\sum_n \gamma_{nk}} \sum_n \gamma_{nk} (\mathbf{x_n} - \boldsymbol{\mu_k})(\mathbf{x_n} - \boldsymbol{\mu_k})^T$$

Expectation Maximization for GMMs

Example of EM implementation



GMMs and K-means

GMMs provide probabilistic interpretation for K-means

- Assume all Gaussian components have $\sigma^2\mathbf{I}$ as their covariance matrices
- Further assume $\sigma \to 0$
- Thus, we only need to estimate $\boldsymbol{\mu_k}$, i.e., means
- Then, the EM for GMM parameter estimation simplifies to K-means

For this reason, K-means is often called hard GMM or GMMs is called soft K-means. The soft posterior γ_{nk} provides a probabilistic assignment for $\mathbf{x_n}$ to cluster k represented by the corresponding Gaussian distribution

Takeaways and Next Time

- Expectation Maximization and GMM
- Next Time: More Unsupervised Learning