Mixture of Gaussian clustering

CS434
Soft vs hard Clustering

• Kmeans performs Hard clustering:
  – Data point is deterministically assigned to one and only one cluster
  – But in reality clusters may overlap

• Soft-clustering:
  – Data points are assigned to clusters with certain probabilities
How can we extend Kmeans to make soft clustering

• Given a set of clusters centers $\mu_1, \mu_2, \ldots, \mu_k$, instead of directly assign all data points to their closest clusters, we can assign them partially (probabilistically) based on the distances

• This can be done by
  – assuming a probabilistic distribution (model) for each cluster
  – compute the probability that each point belongs to each cluster
  – Often referred to as Model-based clustering
Gaussian for representing a cluster

• What exactly is a cluster?
  – Intuitively it is a tightly packed ball-shape like thing

• We can use a Gaussian (normal) distribution to describe it

• Let’s first review what is a Gaussian distribution
Side track: Gaussian Distribution

- **Univariate Gaussian distribution:**
  \[ N(\mu, \sigma^2) \]
  - \( \mu \) – mean, center of the mass
  - \( \sigma^2 \) – standard deviation, spread of the mass

- **Multivariate Gaussian distribution:**
  \[ N(\mu, \Sigma) \]
  - \( \mu \) – \((\mu_1, \mu_2)\)
  - \( \Sigma \) – Covariance matrix
    \[
    \begin{pmatrix}
    \sigma_1^2 & \sigma_{12} \\
    \sigma_{12} & \sigma_2^2
    \end{pmatrix}
    \]
Different covariance matrices

Using different forms of covariance matrix allows for clusters of different shapes

Covariance matrix $\Sigma =$

$$
\begin{bmatrix}
\sigma^2 & 0 \\
0 & \sigma^2
\end{bmatrix}
$$

Covariance matrix $\Sigma =$

$$
\begin{bmatrix}
\sigma_1^2 & 0 \\
0 & \sigma_2^2
\end{bmatrix}
$$

Covariance matrix $\Sigma =$

$$
\begin{bmatrix}
\sigma_1^2 & \sigma_{12} \\
\sigma_{12} & \sigma_2^2
\end{bmatrix}
$$
Mixture of Gaussians

• Assume that we have k clusters in our data
• Each cluster contains data generated from a Gaussian distribution

• Overall process of generating data:
  – first randomly select one of the clusters according to a prior distribution of the clusters
  – draw a random sample from the Gaussian distribution of that particular cluster

• Similar to the generative model we have learned in Bayes Classifier, difference?
  – Here we don’t know the cluster membership of each data point (thus unsupervised)
Clustering using mixture of Gaussian models

• Given a set of data points, and assume that we know there are $k$ clusters in the data, we need to:
  – Assign the data points to the $k$ clusters (soft assignment)
  – Learn the gaussian distribution parameters for each cluster: $\mu$ and $\Sigma$
A simpler problem

• If we know the parameters of each Gaussian: 
  \((\mu_1, \Sigma_1); (\mu_2, \Sigma_2); \ldots; (\mu_K, \Sigma_K)\)
  
  – we can compute the probability of each data point belonging to each cluster

\[
P(x \in C_i | x) = \frac{P(x | x \in C_i) P(C_i)}{P(x)}
\]

\[
\propto \alpha_i \frac{1}{(2\pi)^{d/2}|\Sigma_i|^{1/2}} \exp\left[ -\frac{1}{2} (x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i) \right]
\]

• The same as in making prediction in Bayes classifier
Another simpler problem

• If we know what points belong to cluster $i$, we can estimate the gaussian parameters easily:

$$\alpha_i = \frac{n_i}{n}$$

$$\hat{\mu}_i = \frac{1}{n_i \sum_{x^j \in C_i}} \sum x^j$$

$$\hat{\Sigma}_i = \frac{1}{n_i \sum_{x^j \in C_i}} (x^j - \hat{\mu}_i)(x^j - \hat{\mu}_i)^T$$

Cluster prior  Cluster mean  Cluster covariance

• What we have is slightly different –
  – For each data point $x^j$, we have $P(x^j \in C_i|x^j)$ for $i=1,2,\ldots, K$
Modifications

Cluster prior
\[ \alpha_i = \frac{n_i}{n} \quad \Rightarrow \quad \alpha_i = \frac{1}{n} \sum_{j=1}^{n} P(x^j \in C_i | x^j) \]

Cluster mean
\[ \hat{\mu}_i = \frac{1}{n_i} \sum_{x^j \in C_i} x^j \quad \Rightarrow \quad \hat{\mu}_i = \frac{\sum_{j=1}^{n} x^j P(x^j \in C_i | x^j)}{\sum_{j=1}^{n} P(x^j \in C_i | x^j)} \]

Cluster covariance
\[ \hat{\Sigma}_i = \frac{1}{n_i} \sum_{x^j \in C_i} (x^j - \hat{\mu}_i)(x^j - \hat{\mu}_i)^T \quad \Rightarrow \quad \hat{\Sigma}_i = \frac{\sum_{j=1}^{n} P(x^j \in C_i | x^j) (x^j - \hat{\mu}_i)(x^j - \hat{\mu}_i)^T}{\sum_{j=1}^{n} P(x^j \in C_i | x^j)} \]
A procedure similar to Kmeans

• Randomly initialize the Gaussian parameters
• Repeat until converge

1. Compute \( P(x^i \in C_i | x^j) \) for all data points and all clusters
   This is called the E-step for it computes the expected values of the cluster memberships for each data point

2. Re-compute the parameters of each Gaussian
   This is called the M-step for it performs maximum likelihood estimation of parameters
Gaussian Mixture Example: Start
After first iteration
After 2nd iteration
After 3rd iteration
After 4th iteration
After 5th iteration
After 6th iteration
Q: Why are these two points red when they appear to be closer to blue?
Behavior of EM

- It is guaranteed to converge
  - It monotonically increases the log likelihood function
    \[
    l(D) = \sum_{i=1}^{n} \log p(x_i | \mu_j, \Sigma_j, j = 1, ..., k) = \sum_{i=1}^{n} \log \sum_{j=1}^{k} \alpha_j p(x_i | \mu_j, \Sigma_j)
    \]
- In practice it may converge slowly, one can stop early if the change in log-likelihood is smaller than a threshold
- Like K-means it converges to a local optimum
  - Multiple restart is recommended
Comparing to Kmeans

• Assignment step
  – GMM: soft assignment
  – Kmeans: hard assignment

• Parameter estimation step
  – GMM: estimate mean and covariance matrix
  – Kmeans: estimate mean only

• Which one converges faster
  – kmeans

• Which one is more flexible:
  – GMM can capture clusters of more flexible shapes
K-Means is a Special Case

- we get K-Means if we make following restrictions:
  - All Gaussians have the identity covariance matrix (i.e., spherical Gaussians)
  - Assume equal prior for all clusters
  - Use hard assignment for the E-step to assign data point to its most likely cluster
    - Assign a point to the cluster with highest probability

\[
P(x \in C_i \mid x) = \frac{P(x \mid x \in C_i)P(C_i)}{P(x)}
\]

\[
\propto \alpha_i \ast \frac{1}{(2\pi)^{d/2}|\Sigma_i|^{1/2}} \exp[-\frac{1}{2}(x - \mu_i)^T \Sigma_i^{-1}(x - \mu_i)]
\]
Evaluating clustering results

• By user interpretation
  – does a document cluster seem to correspond to a specific topic?
  – This can be difficult in many domains

• Internal measure: objective functions used in clustering can be used to gauge the quality of the clustering solutions
  – Sum-squared-error (also referred to as the scatter)

\[ J_e = \sum_{i=1}^{k} \sum_{x \in c_i} ||x - \mu_i||^2 \]

  – Likelihood of the data
  – Evaluate how well the clustering structure explains the (variations that we see in) data
External measures

• Use labeled data to perform evaluation and test how well the discovered cluster structure matches the class labels

• Rand index:
  – For each pair of objects see whether the clustering solution put them together (or not) the same way the class label did
  – Count the portion of correct pairs

• Purity:

• Normalized mutual information

\[
\frac{2I(y, c)}{H(y) + H(c)}
\]