AINT351 - Clustering

Date: 31-10-16

K-Means clustering

Clustering

- Idea of clustering is to group patterns so that patterns are similar to one another within the same cluster
- Patterns are dissimilar to those in other clusters
- Need some sort of metric to determine how similar things are

Hierarchical clustering

- Breaks down the dataset into a series of nested clusters
- Single cluster at the top with all data
- N clusters at the bottom for each point
- Can be displayed as a dendrogram

K-means algorithm

- K-means as a simple clustering algorithm
- Implementing K-means is easy
- K is the number of clusters
- Often K is set by hand
- Assign the data to k clusters
- Calculate the centroids (means)
- Loop until convergence and do the following:
  1. For each point, put the point into the cluster whose centroid it is closest
  2. Recompute the cluster centroids
  3. Repeat loop until there is no change in clusters between two consecutive iterations

---

Mixture of Gaussians

limitations of simple Gaussian and PCA models

- They're a convenient way to reduce dimensionality of high dimensional data sets
- The problem is that they make very strong assumptions about the distribution of the data
    - Only the mean and variance of the data are teken into account

Mixture of models

- Clusters can overlap
- Data may exhibit non-binary strength of association to all clusters
- Probabilistic method
- Each cluster is a generative model
- Clusters have parameters (means & covarience)
- Can have different Gaussians for capturing different parts of the data sets

Recall K-means algorithm

- How is MoG implemented?
- Randomly assign Kdata points at the centroids
- Loop to find centres
- Mixture of guassians operates in an analogous fashion

EM algorithm

- Need to know the Guassian parameters (mean & variance) for each cluster to estimate data point cluster membership
- But need to know data assignments to estimate the parameters
- Solution is to use EM algorithm
- What's the value of prob that it has come from each class
- Have to compute the prob of how it came from a class

E-step:

- Look at each data point and calculate how likely it came from a given cluster

M-step:

- Re-estimate Gaussian paremeters to fit the assigned points
- Repeat until convergence achieved

NB: sensitive to starting point

- The probability goes up when it is closer to the mean
- If it is at the tail end of a guassian curve that isn't near another, the probability becomes more certain