Lecture 20,21 Nov. 3,6 -2006

Data Visualization

STAT 442 / 890, CM 462

Lecture: Ali Ghodsi Scribes: Stefan Pintilie

1 Combinatorial Algorithms:

Suppose we have n data points which are indexed $1 \dots n$ and suppose we would like to cluster these points into k clusters $1 \dots K$. We need to assign each point to one cluster. $k = C(i)$ where $C(i)$ is the encoder. The goal is to find a grouping of data such that distances between points within a certain cluster are small and distances between points that are assigned to different clusters tends to be large.

If we need to minimize the distance within each cluster then we need:

$$\min_{c*} w(C) = \sum_{k=1}^{K} \sum_{C(i)=k} \sum_{C(j)=k} d_{ij}$$

Consider now the sum of all of the distances between all of the points. Call this distance T.

$$\begin{aligned} T &= \sum_{k=1}^{K} \sum_{C(i)=k} \Big(\sum_{C(j)=k} d_{ij} + \sum_{C(j)\neq k} d_{ij} \Big) && (1) \\ &= \sum_{k=1}^{K} \sum_{C(i)=k} \sum_{C(j)=k} d_{ij} + \sum_{k=1}^{K} \sum_{C(i)=k} \sum_{C(j)\neq k} d_{ij} && (2) \\ &= w(C) + b(C) && (3) \end{aligned}$$

Where $w(C)$ is the previous within cluster distance and $b(C)$ is the between cluster distance.

Since T is a constant it is sufficient to minimize $w(C)$ because at the same time we maximise $b(C)$. The problem is that the minimization of $w(C)$ is a computationally intensive task and will only be feasible for small data sets. The number of distinct assignments, and thus the complexity of the algorithm, for n points and k clusters is:

$$S(n,k) = \frac{1}{k!}\sum_{k=1}^{K}(-1)^{K-k}\binom{K}{k}k^n$$

So for example

$$S(10,4) = 34105 \tag{4}$$

$$S(19,4) \doteq 10^{10} \tag{5}$$

Clearly as the data set becomes large any such calculation takes too long.

Idea: Take a greedy descent approach. First initialize $C(i)$ to a starting value. Then, take the slope in such a way that the criterion $w(C)$ improves at each step. Stop the algorithm when there is no further improvement.

$$w(C) = \sum_{k=1}^{K}\sum_{C(i)=k}\sum_{C(j)=k} d_{ij} \tag{6}$$

$$= \sum_{k=1}^{K}\sum_{C(i)=k}\sum_{C(j)=k} ||x_i - x_j||^2 \tag{7}$$

$$= \sum_{k=1}^{K}\eta_k\sum_{C(i)=k} ||x_i - \bar{x_k}||^2 \tag{8}$$

Where η_k is the number of points in cluster k.

Aside: For any set of observations S:

$$\bar{x_S} = argmin_m \sum_{i\epsilon S} ||x_i - m||^2$$

Now,

$$c* = \min_{C} \sum_{k=1}^{K} \eta_k \sum_{C(i)=k} ||x_i - \bar{x_k}||^2 \quad (9)$$

$$= \min_{C,\{m\}_{k=1}^{K}} \sum_{k=1}^{K} \eta_k \sum_{C(i)=k} ||x_i - m||^2 \quad (10)$$

We can optimize 10 in two steps.

1. Given C, 10 is minimized yielding $\{m_1 \dots m_k\}$ to be the mean of all data points in each cluster.

2. Given $\{m\}_{k=1}^{K}$, 10 is minimized by assigning each x_i to its closest $\bar{x_k}$. So $c* = argmin_{1 \leq k \leq K} ||x_i - \bar{x_k}||$.

1.1 K-Means Clustering

1. Set K, the number of clusters.

2. Assign points to these K clusters randomly.

3. Iterate through the stapes of the previous procedure until there is no improvement in $w(C)$.