TIM245 Lecture 15 (5/24/17)

Agenda

1) Review Lecture 14

2) Finish K-Means Algorithm

3) Density Based Methods and DB-Scan algorithm

4) Hierarchical Methods and Agglomerative clustering algorithm

5) Project Phase IV

6) Return graded midterms

② K-Means Algorithm

Given K, the number of clusters, the K-means algorithm works as follows:

1) Choose K random instances (seeds) to be the initial centroids

2) Assign each instance to the closest centroid based on some distance function (typically Euclidean)

3) Recompute the centroid using the current cluster membership

4) Repeat steps 2, 3 until convergence criteria is met

Convergence Criteria

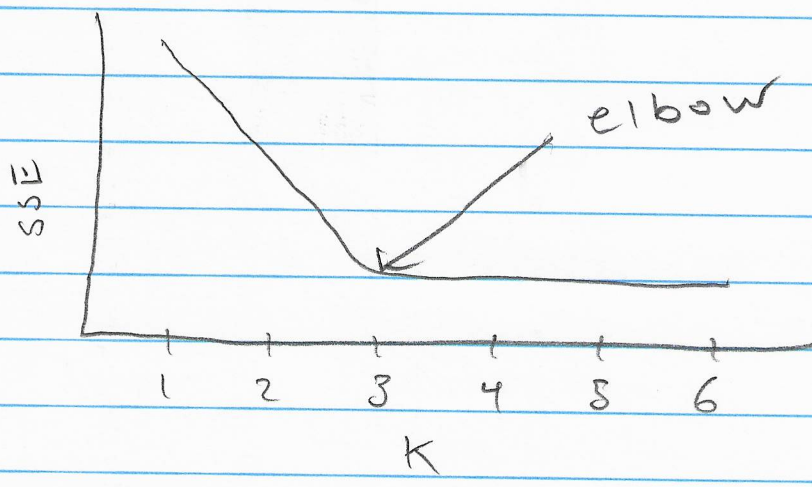
- No reassignments of instances to new clusters

- Decrease in the sum of squared error is below some threshold

$$SSE = \sum_{k=1}^{K} \sum_{i=1}^{n} (x_i - M_k)^2$$

How do we select K?

1) Elbow Method: plot SSE vs K and look for the "elbow"



2) Domain knowledge about the problem

e.g I know there are 5 different kinds of users

How do we interpret the Clusters?

1) Have Subject matter experts manually examine the centroids and a sample of instances for each cluster

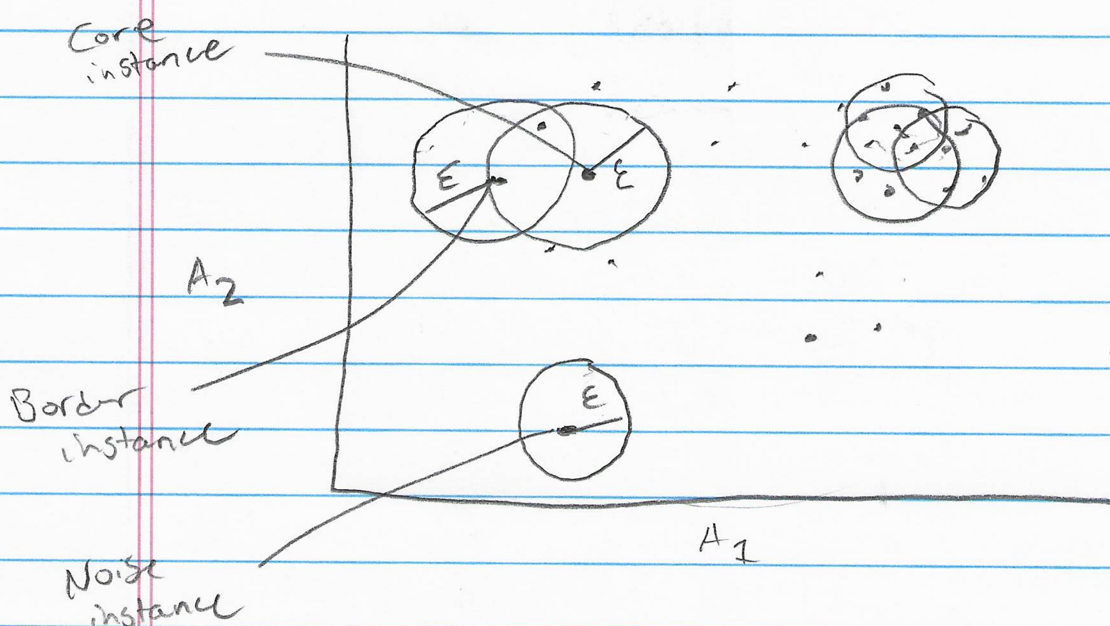
2) Use the assigned cluster as the target attribute for a classification model, e.g. decision tree

Advantages
- Simple and easy to explain
- Fast
- Works well in practice if K is set correctly

Disadvantages
- Sensitive to noise and outliers
- Trouble with non-globular clusters
- Fixed K

③ Density Based Clustering Methods and DB-scan

Basic Idea: Clusters are a set of density connected instances



Neighborhood (d_i) $\triangleq$ # instances inside of ε distance of d_i

MinPts $\triangleq$ user provided threshold for the minimum neighborhood size

Core instances $\triangleq$ Neighborhood (d_i) $\geq$ MinPts

Border instances $\triangleq$ inside neighborhood of a core instance

Noise instance $\triangleq$ anything else

DB Scan Algorithm

1) Compute the neighborhood for each instance $d_1, d_2, \ldots d_n$

2) Label each instance as core, border, or noise

3) Eliminate noise instances

4) Connect all core instances that are in the same neighborhood as eachother

5) Make each connected group of Core instances into a cluster

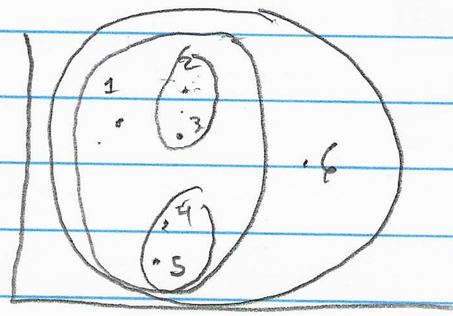
6) Assign each border instance to the closest cluster

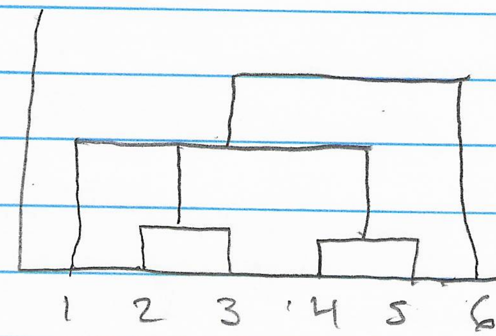
Advantages
- Resistant to outliers
- Number of clusters is not fixed
- Can handle clusters of arbitrary shapes and sizes
- doesn't depend on the seed (consistent results)

Disadvantages
- struggles with sparseness in high-dimensional spaces
- doesn't cluster all of the data
- computationally expensive

(4) Hierarchical Clustering

<u>Basic idea</u>: Clusters can be represented as a hierarchy





Dendrogram

Clusters are merged (Agglomerative) or split (divisive) in order to minimize a cost function

For agglomerative clustering, the cost function is the distance between the two clusters C_j C_K

- Single Linkage : minimum distance
- Complete Linkage : maximum distance
- Average Linkage : average distance

Agglomerative Clustering Algorithm

1) Compute a matrix of the cost of merging any two clusters

2) Perform the lowest cost merge

3) Update the cost matrix

4) Repeat until there is only one cluster

Advantages

- Works well for data that has a natural hierarchy
- Easy to interpret and select the best level of resolution for the problem

Disadvantages

- Expensive with respect to time and space
- Local optimization (merges are final)