Machine Learning INF2008

Lecture 07: Unsupervised Learning

Donny Soh

Singapore Institute of Technology

A man is judged by the company he keeps

A man is judged by the company he keeps

A man is judged by the company he keeps

A man is judged by the company he keeps

A man is judged by the company he keeps

In general, two types of unsupervised learning

1. Partitional clustering algorithms:
    1. K-Nearest Neighbour
    2. K-means
    3. DBScan
    4. LDA

2. Hierarchical algorithms: finds successive clusters using previously established clusters
    1. Agglomerative (“bottom-up”): HLDA
    2. Divisive (“top-down”)

Nearest Neighbor Algorithm: Intuition

- For each new test datapoint with x-variables, the nearest neighbour algorithm simply finds the k number of datapoints closest to the datapoint.

- Upon finding these datapoints, it finds out which classes these datapoints belong to and takes a vote count and aligns itself with these datapoints.

- Suppose we have the example: “How should I go to work today".

- Typically most of us go to work either via bus or the train. Let’s assume grab suddenly has this great offer if you use grabshare.

- You soon realize that if you share the ride with at least 2 more colleagues, not only do you spend less time commuting, you end up paying less for your trip as well!

Nearest Neighbor Algorithm: Intuition

So now every morning, you call up your 3 colleagues that stay closest to you.

As long as ***2*** of them agree to take grabshare with you that day, that will be the mode of your transport for that day.

What you are doing unknowingly is the k Nearest Neighbour algorithm.

You are looking for the nearest = 3 neighbours that live closest to you to share a grab ride.

Let's assume that the number of people that reply yes to you takes on the variable of *r*.

As long as *r* is greater or equal than the threshold value of ***2***, you will go ahead with grab. Else you will decide to take the public transport (bus or MRT).

Distance Measures

- Every sample is represented by a vector of numbers (eg wordvec).
- Classification / Regression is done by voting of samples from the k nearest points.
- Classification: the winner of the vote from k nearest points.
- Regression: the mean of the k nearest samples.

Euclidean Distance

$$d(x,y) = \sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$$

Manhattan Distance

$$d(x,y) = \sum_{i=1}^{k} \|x_i - y_i\|$$

Maximum Norm

$$d(x,y) = max_{1\le i\le p}\|x_i - y_i\|$$

Nearest Neighbor Algorithm: Issues

- Very prone to overfitting. A good choice of the value of k is the square root of the number of training samples.
- Doesn't work well when the number of training samples is large.
- Doesn't work well when the number of features is large (data is very sparse). (**why?**)

This is an example of what overfitting is.

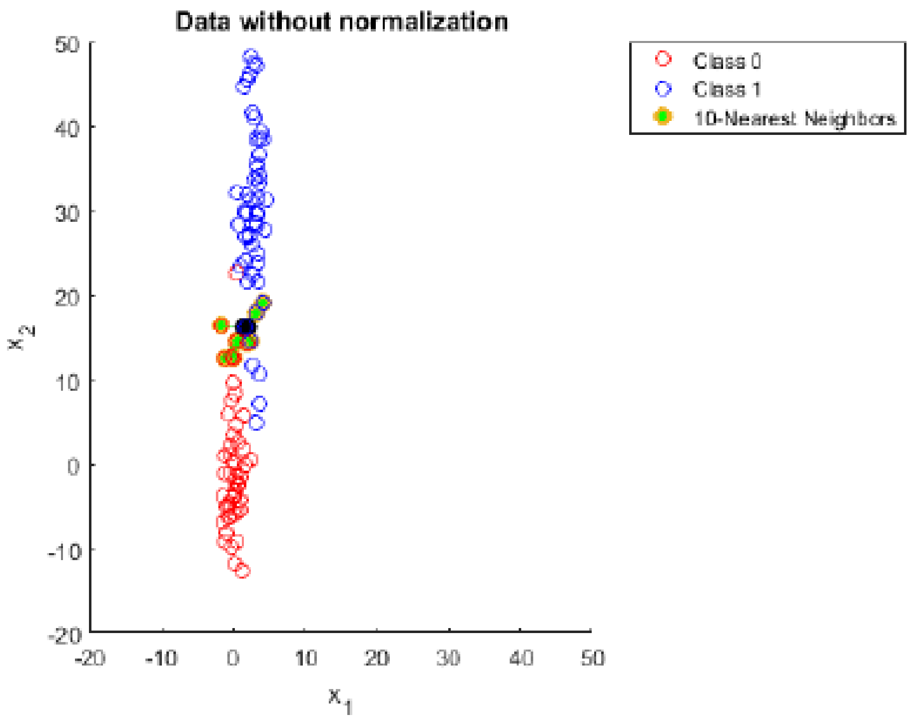
Instead of a simple boundary that divides between two classes, the boundary is very complex, which is liable to lead to many errors.

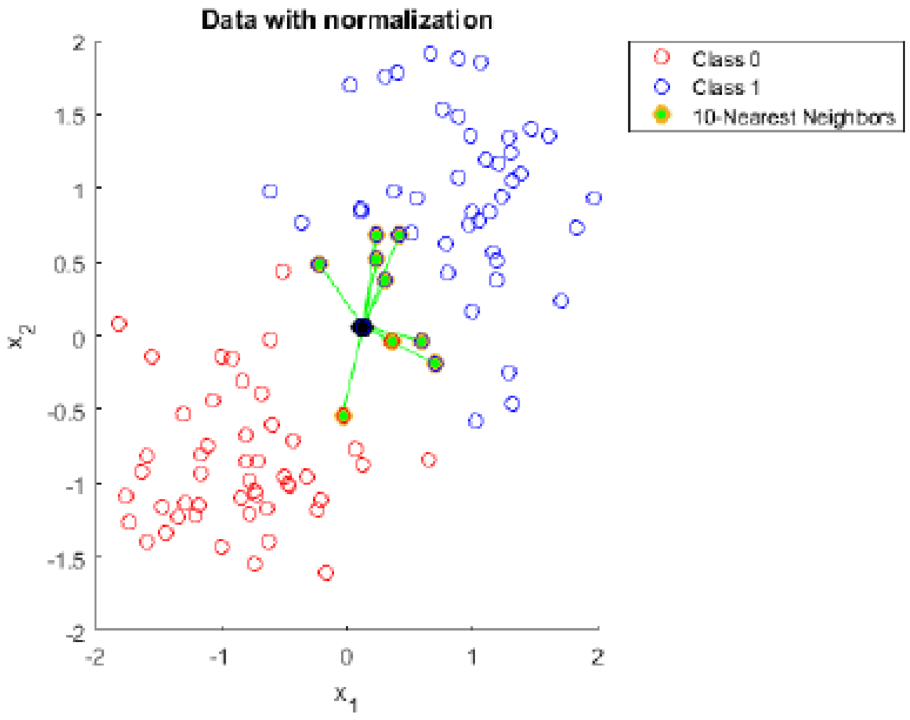
Nearest Neighbor Algorithm: Normalization

- Do remember (where possible) to scale your features. The kNN algorithm relies on a majority vote based on the class of the nearest k datapoints in the dataset.

- Consider a simple two class classification problem, where a Class 1 sample is chosen (black) along with it's 10-nearest neighbours (filled green). In the left figure, data is not normalized, whereas in the right one it is.

- Without normalization, all the nearest neighbours are aligned in the direction of the axis with the smaller range and this leads to an incorrect classification.





K-Means Clustering

- The k-means algorithm is an algorithm to cluster *n* objects into *k* clusters.
- The algorithm will partition all points into *k* disjoint clusters.
- Each cluster will have a centroid (centre point).
- These clusters will minimize the cost of the points to the *k* centroids.

$$Loss = \sum_{j=0}^{k} \sum_{i=0}^{n} \left\| x_i - \mu_j \right\|^2$$

K-Means Clustering: How it works?

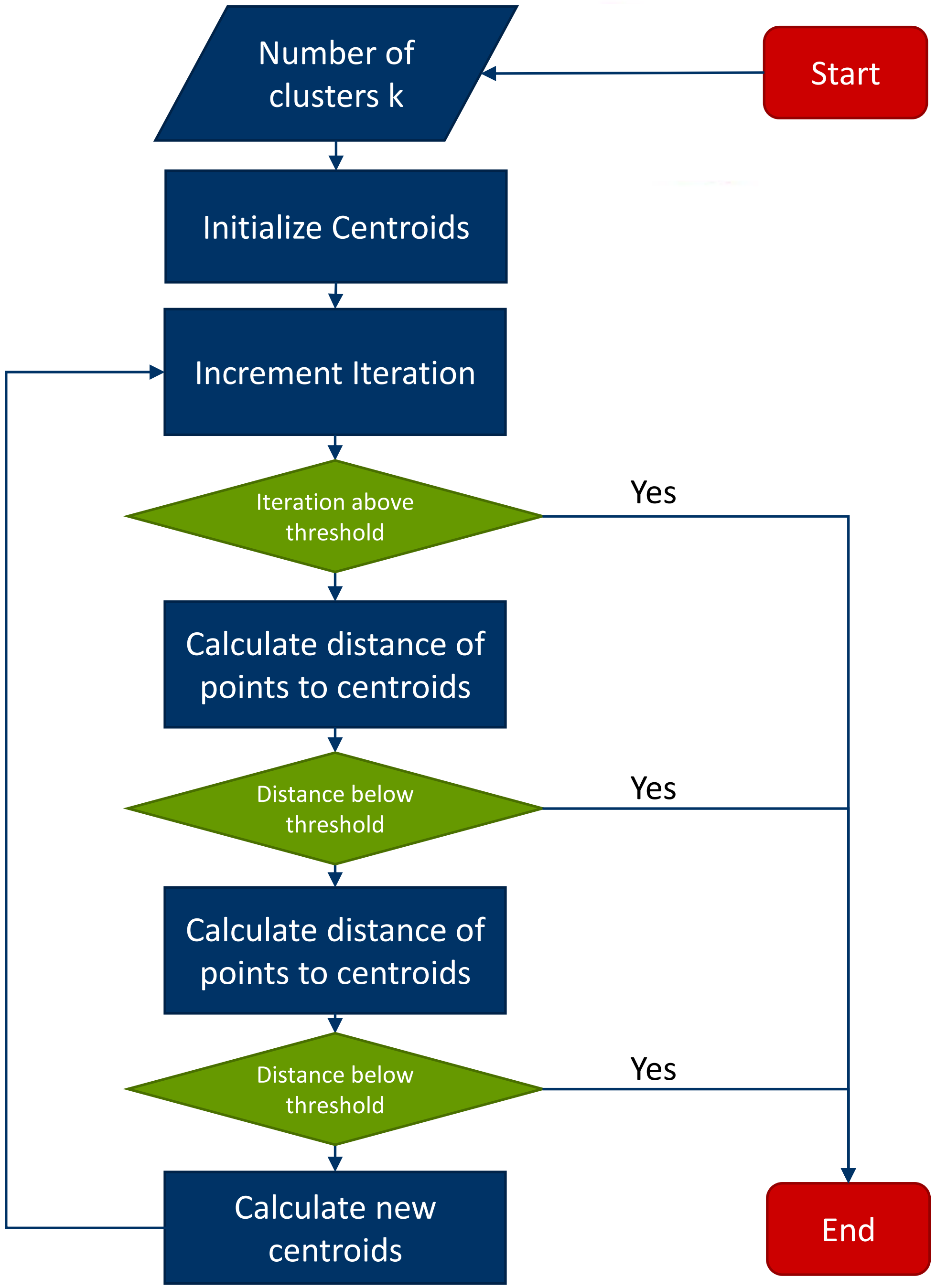
Initializes the data with *k* points. These points are referred to as centroids. (eg data points).

For every point in the dataset, it finds the points in the *k centroids* closest to these points in the dataset.

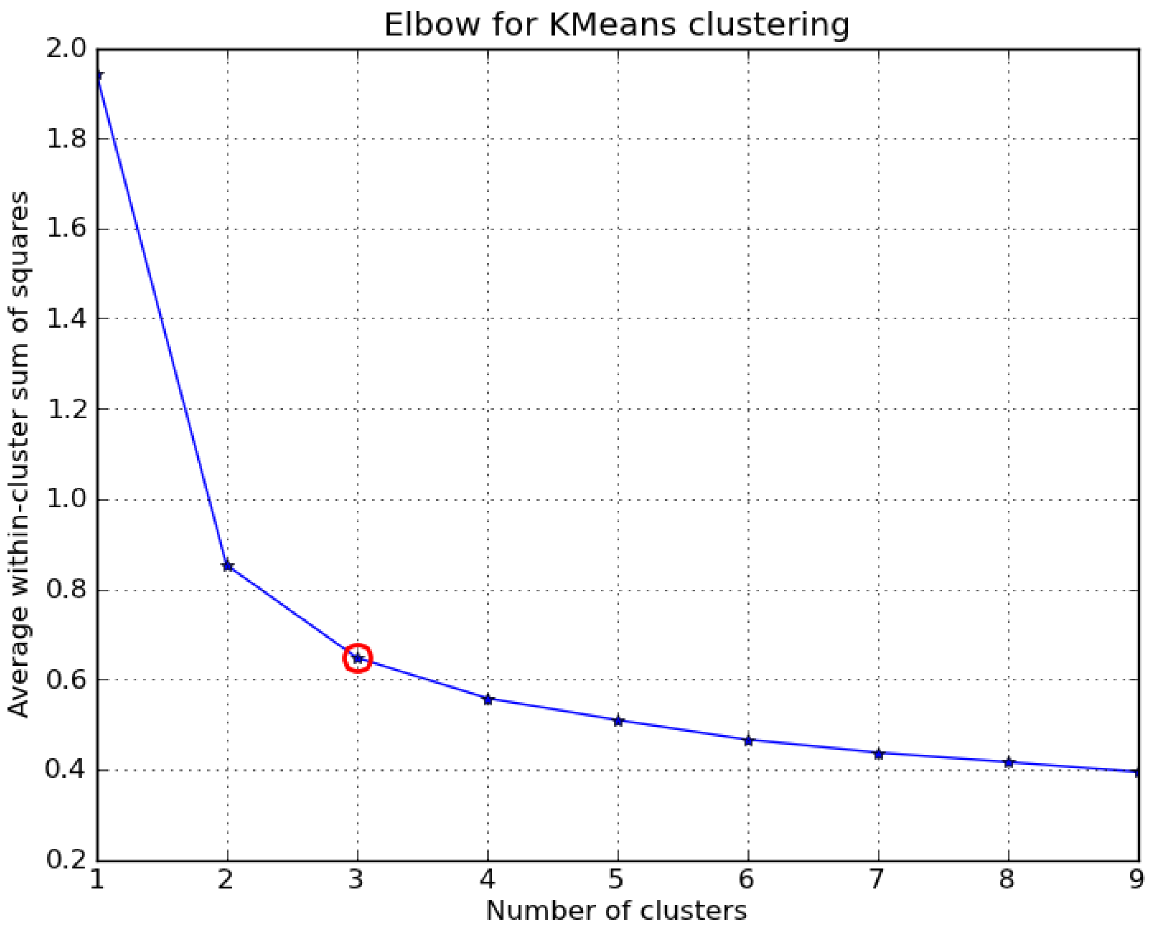
For each of this points, a new centroid is calculated.

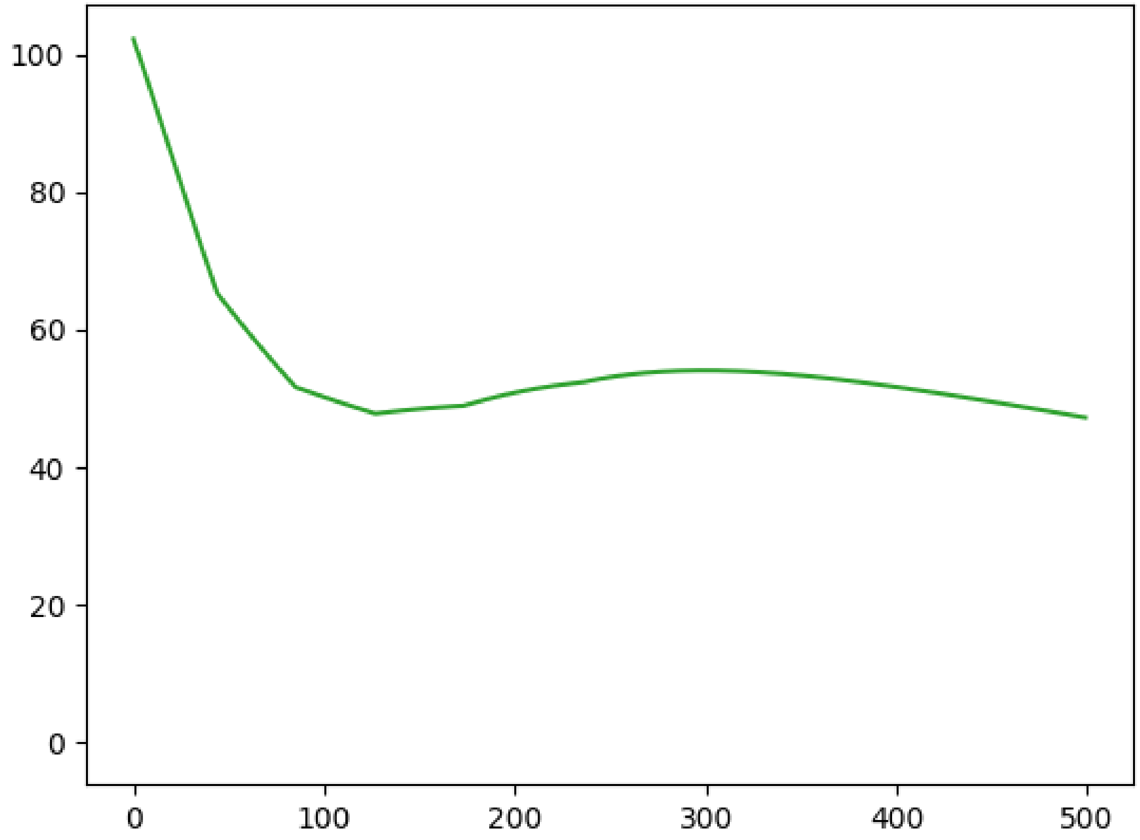
This process is continued until either

- the number of iterations for the clustering has been reached.
- the change in the loss goes below a certain threshold.
- the change in the centroid location goes below a certain threshold.



K-Means Clustering: How to pick K?





Other Clustering Methods