# Machine Learning Programming Assignment 5

Maya Santhira Sekeran, Alpha Mary Skaria, Tharakesh Paddolkar

January 21, 2018

## 1. K-means Clustering Algorithm

K-means is a method of clustering observations into a specific number of disjoint clusters. The" K" refers to the number of clusters specified. Various distance measures exist to determine which observation is to be appended to which cluster. The algorithm aims at minimizing the measure between the centroid of the cluster and the given observation by iteratively appending an observation to any cluster and terminate when the lowest distance measure is achieved.

### 1.1 Overview of Algorithm

1. The sample space is initially partitioned into K clusters and the observations are randomly assigned to the clusters.

2. For each sample: • Calculate the distance from the observation to the centroid of the cluster. • if the sample is closest to its own cluster then leave it else select another cluster.

3. Repeat steps 1 and 2 until no observations are moved from one cluster to another

When step 3 terminates the clusters are stable and each sample is assigned a cluster which results in the lowest possible distance to the centroid of the cluster.

## 2.Data Processing.

*Instance Reader class:* Reads the instances of a training set from a file.

*Metadata reader class* :Used for Reads the possible classes and domains of feature from a file.

*Instance class:* used for Features and the classification

*Training set class : Training set* possible classes and domains of features and spiliting the test size..Split the data into two parts at 0.67:0.33 ratio,

*KMean Class* : Initialize the centroids .compute a random feature value in between the feature space given by the training set.
iterate over all instances in given training set compute distance to centroids.
Compute new centroids and assigned instances to them.
get the classification of cluster by majority of instances in the cluster and by below methods are used..
public KMean(int _k)
public String classify(Instance<Integer> i)
public void learn(Trainingset<Integer> t)
private void initializeCentroids(Trainingset<Integer> t)
private double distance(Instance<Double> centroid, Instance<Integer> i).

*Validator class* Used for validation of test set and computing the confusion matrix.

### 3.Experiment Results.

For 100 samples, average accuracy will be total accuracy/ 100. When percentage of average accuracy subtracted to 100, it gives mean error rate.and value of k=4.

| Mean Error rate | Accuracy |
|---|---|
| 29.680701 | 70.3193 |

.Cluster 0 <-- unacc,

Cluster 1 <-- acc,

Cluster 2 <-- good,

Cluster 3 <-- vgood

clustered instances .

cluster 1: [308, 59, 2, 4]

cluster 2: [197, 77, 4, 7]

cluster 3: [217, 30, 13, 8]

cluster 4: [82, 97, 27, 26]

**Output:**

k = 4

Mean error over 100 samples: 0.29680701754385996

| | unacc | acc | good | vgood |
|---|---|---|---|---|
| unacc | 362 | 44 | 0 | 0 |
| acc | 73 | 48 | 0 | 0 |
| good | 12 | 11 | 0 | 0 |
| vgood | 11 | 9 | 0 | 0 |

From the confusion matrix, we can observe that most cases the class label ’unacc’ are classified

correctly as the majority of ’car data’ has ’unacc’.

**4.Comparison with KNN and Naive Bayes Classifie**r

Below are the results from the implementing previous classification algorithms.

| Classification algorithm | Accuracy |
|---|---|
| Naive Bayes Classifier | 69.4 % |
| KNN Classifier | 70.05% |
| K-Means | 70.31% |

In this experiment, the accuracy rate of KMeans was slightly better compared to Nave Bayes. And KNN.

..