# AMLT Practical Work: Distance Based Outlier Detection and Clustering Algorithms

Mohammad Bashir Kazimi *[1]

[1]Course Name: Advanced Machine Learning Techniques
[2]Master's in Artificial Intelligence (UPC, UB and URV)

January 4, 2017

## 1 Introduction

This project is composed of two parts. First part contains implementing and comparing two different outlier detection algorithms, more specifically distance based outlier detection algorithms;$k$th-nearest neighbor distance, and One-time sampling. The second part consists of implementing two clustering algorithms; Bisecting k-means and regular k-means, and comparing them.

## 2 Outlier Detection

An outlier refers to "an observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism" [2]. Existence of an outlier could highly affect the inference one gets when observing the data. It could also affect the performance of machine learning algorithms in classification, regression, etc.

Many techniques have been proposed on detecting outliers among which are distance based outlier detection methods; $k$th-nearest neighbor distance, and One-time sampling, and these two methods have been implemented and compared in this project and reported in the following two subsections.

### 2.1 The $k$th-nearest neighbor distance

The first distance based approach by Knorr and Ng [4] proposed to detect an outlier $x$ based on whether a *fraction* of instances have a distance from $x$ that is larger than a *threshold*. The main challenge in this method was defining a fraction and a threshold and the ranking of the outliers. As a solution to these problems, the $k$th-nearest neighbor distance method was proposed by [5]. They detect an outlier based its distance from its kth-nearest neighbor using the following formula:

$$q_{kthNN}(x) := d^k(x; X) \tag{1}$$

where $d^k(x; X)$ is the distance between $x$ and its kth-nearest neighbor in the data set $X$. This algorithm is also referred to as *dee-kay-en*

*mohammad.bashir.kazimi@est.fib.upc.edu

## 2.2 One-time Sampling

The kth-nearest neighbor approach is computationally quite expensive since for every instance we have to iterate through the whole data set to find the distance. Wu and Jermanine [7] proposed a sampling approach where instead of finding the kth-nearest neighbor distance by iterating over the whole data set, they used a random sample from the data set for each instance and found the approximate kth-nearest neighbor.

To increase accuracy, *One-time sampling* method by [6] uses a sample only once to find the the outliers based on their kth-nearest neighbor from the sample.

## 2.3 Experiment Setup

The two methods for outlier detection discussed above have been implemented in this work using Python. Also, an ipython notebook have been provided as an interface to interactively test and compare the two methods based on a small toy data set for validation and two other different data sets. The first data set tested is American Basketball players statistics taken from [1], and the second data set is the scripts from each episode of the Tv Series *How I met your mother*. It has first been preprocessed to obtain a vector representation of the episodes using the *term frequency, inverse document frequency* referred to as *tfidf*.

## 2.4 Results

The main improvement that the sampling technique has provided over the naive kth-nearest neighbor technique is the computational time. Therefore, we compare and report the results of computational time for the two methods on the toy data set and the other two real data sets mentioned previously.

### 2.4.1 Results for Basketball Statistics

The following graph shows the running time of the two algorithms on the Basketball Statistics Data Set.

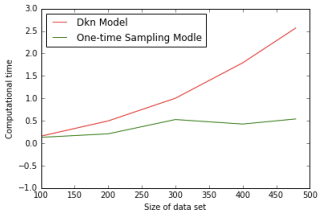
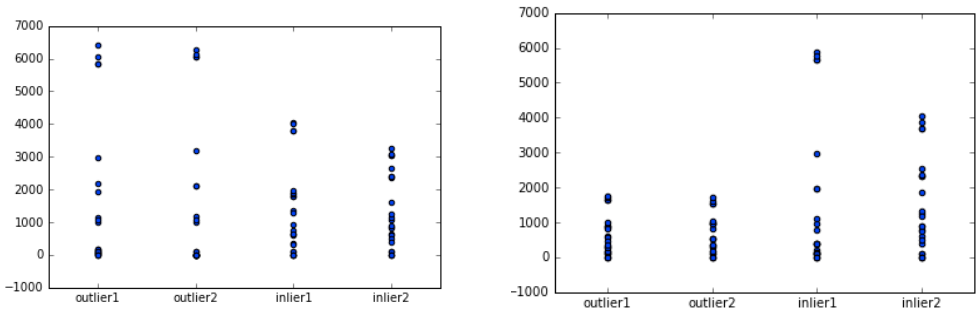



Figure 1: Computation Time For the Two Models for Different Data Set Sizes

As it can be inferred from the graph above, the higher the size of the data set, the bigger the difference between the time the two models take, and thus, One-time sampling proves to be much faster than *dee-kay-en*.

Since our data is not labeled as inlier or outlier, we cannot check the accuracy of the models, but to realize that the most different instances have been selected as outliers, we take two random instances from outliers detected by the models and two random instances that are not outliers and plot their values. The graphs below show exactly this.




(a) Sampling Model

(b) Dee-kay-en Model

Figure 2: Outlier and Inlier Samples for the two Models

From the graphs above, we can understand that the two models choose the most different instances as outliers.

### 2.4.2 Results for TV Series Scripts

The following graph shows the running time of the two algorithms on the TV Series script data set; How I Met Your Mother.

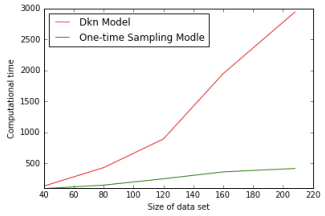
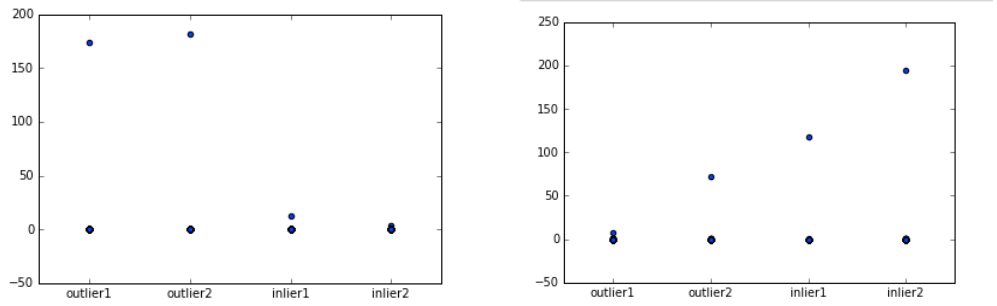



Figure 3: Computation Time For the Two Models for Different Data Set Sizes

As it can be inferred from the graph above, the higher the size of the data set, the bigger the difference between the time the two models take, and thus, One-time sampling proves to be much faster than *dee-kay-en*.

Since our data is not labeled as inlier or outlier, we cannot check the accuracy of the models, but to realize that the most different instances have been selected as outliers, we take two random instances from outliers detected by the models and two random instances that are not outliers and plot their values. The graphs below show exactly this.




(a) Sampling Model

(b) Dee-kay-en Model

Figure 4: Outlier and Inlier Samples for the two Models

From the graphs above, we can understand that the two models choose the most different instances as outliers.

# 3 Clustering Algorithms

To be able to group data into categories/classes each of which contain similar instances when the data is not labeled already is an important task. An important method to do this task is the unsupervised clustering method. The two main methods in clustering are agglomerative hierarchical clustering and K-means. In this project, K-Means and a variation of K-Means named Bisecting K-Means have been implemented. The two algorithms have been tested on the two data sets mentioned in the previous section and the results have been compared between them and also other clustering algorithms from existing Python library sklearn.

## 3.1 K-Means

In K-Means, data is expected to be grouped into $k$ different clusters where $k$ is determined by the user. The basic K-Means Algorithm is outlined as follows.

1. Select K instances as initial centeroids
2. Group all instances to the most similar centeroids
3. Recalculate the centeroids
4. Repeart steps 2 and 3 until centeroids don't change or the maximum number of iterations reached.

This algorithm has been implemented in this project.

## 3.2 Bisecting K-Means

In this algorithm, at each step, the data is divided and clustered into 2 clusters using the regular K-Means algorithm, and then the cluster with the higher number of instances (or with less similarity among its members) is split and clustered until desired number of clusters have been achieved.

The Bisecting K-Means Algorithm is outlined as follows.

1. Choose a cluster to split (initially the whole data set)

2. Cluster using K-Means with $k$ set to 2 in order to get 2 clusters
3. choose a cluster with the highest overall similarity to keep and choose the next one to repeat step 1 with
4. Repeat previous steps until desired number of clusters have been reached.

This algorithm has been implemented in this project.

## 3.3 Agglomerative Clustering Algorithm

The basic technique for agglomerative clustering is outlined as follows [3]

1. Compute the similarity between all pairs of clusters (initially each data instance pair)
2. Merge the two most similar clusters
3. Update the similarity matrix since we have fewer clusters now
4. repeat until a single cluster remains

This algorithm has not been implemented in this project, but to compare our implemented K-Means and Bisecting K-Means algorithms, a built-in agglomerative clustering technique named average linkage has been used.

## 3.4 Experiment Setup

The two methods for clustering; K-Means and Bisecting K-Means discussed above have been implemented in this work using Python. Also, an ipython notebook have been provided as an interface to interactively test and compare the two methods implemented, and the built-in average linkage method based on two other data sets discussed in the previous section for outlier detection.

## 3.5 Results

To test the performance of the methods, since we do not have the class labels for our data, we compare the cluster quality of the methods based on overall similarity of the clusters; which is the squared value of the cluster centeroid.

Here we give a bar graph of the overall similarity for the three models.

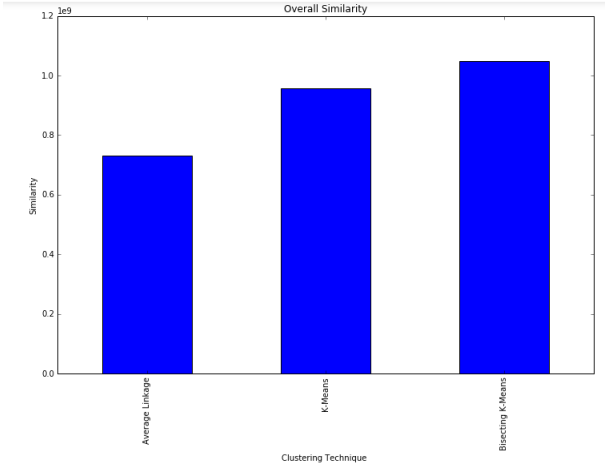



Figure 5: Overall similarity of clusters for Basketball data set by the clustering methods

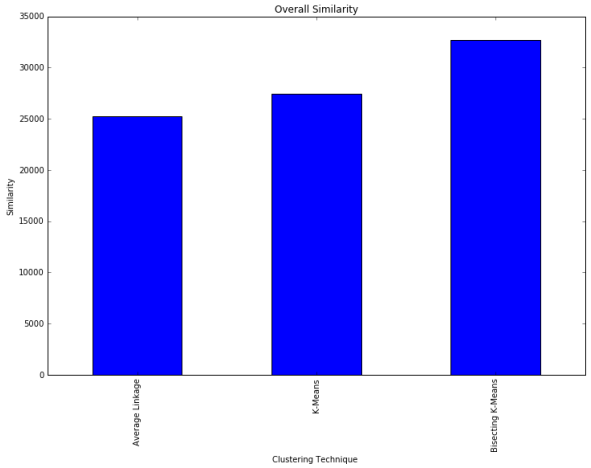



Figure 6: Overall similarity of clusters for How I met Your Mother Scripts data set by the clustering methods

# 4 Conclusion

In this work, two different topics have been discussed. The first one is *distance-based outlier detection*. Among the many methods in distance-based outlier detection, two of them; *the kth-nearest neighbor distance (dee-kay-en)* and *one-time sampling* have been implemented and compared. As the results show in section 2, the running time of kth-nearest neighbor is quadratic. Thus, a better algorithm one-time sampling has been proposed and proved to work faster, as discussed in section 2.

The second topic in this project is *clustering algorithms*. Two algorithms in partitional clustering; K-Means and Bisecting K-Means, have been implemented and compared to a built-in agglomerative algorithm. As discussed in section 3, partitional algorithms work better than agglomerative, and among the partitional methods, Bisecting K-Means works better than regular K-Means.

# References

[1] BasketballValue.com Data Files. http://basketballvalue.com/downloads.php. [Online; accessed 28-12-2016].

[2] G. Enderlein. Hawkins, d. m.: Identification of outliers. chapman and hall, london – new york 1980, 188 s., £ 14, 50. *Biometrical Journal*, 29(2):198–198, 1987.

[3] Michael Steinbach George Karypis and Vipin Kumar. A comparison of document clustering techniques.

[4] Edwin M Knox and Raymond T Ng. Algorithms for mining distancebased outliers in large datasets. Citeseer.

[5] Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. Efficient algorithms for mining outliers from large data sets. In *ACM SIGMOD Record*, volume 29, pages 427–438. ACM, 2000.

[6] Mahito Sugiyama and Karsten Borgwardt. Rapid distance-based outlier detection via sampling. In *Advances in Neural Information Processing Systems*, pages 467–475, 2013.

[7] Mingxi Wu and Christopher Jermaine. Outlier detection by sampling with accuracy guarantees. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 767–772. ACM, 2006.