A LITERATURE REVIEW OF K-MEANS CLUSTERING FOR MINING NEAREST WI-FI HOTSPOTS

A Literature Review

Presented to

Prof. Sharon Stranahan

Department of Computer Science

San José State University

In Partial Fulfillment

of the Requirements for the Class

CS200W Sec 02 2017

By

Yuvraj Singh Kanwar

November 6, 2017

# Table of Contents

[I. Introduction 1](#_Toc497925769)

[II. Evolution of K-means Clustering 2](#_Toc497925770)

[III. K-means Algorithm and its Result 3](#_Toc497925771)

[IV. Benefits of using K-means clustering 4](#_Toc497925772)

[V. Points to be careful about when using K-means 5](#_Toc497925773)

[VI. Conclusion 5](#_Toc497925774)

[References 6](#_Toc497925775)

# Introduction

Wi-fi is one of the most important technologies being used in the internet era as it allows the users to connect wirelessly to the internet. It has become a necessity, just like food and shelter. Many restaurants, malls, cafes and libraries provide free public wi-fi these days. Naturally, it becomes a valid option to have an application that can assist you in searching nearest wi-fi accurately. People do not have efficient solution to identify wi-fi hotspot near them so that they can just reach there and get connected to the internet. In the era of technology that we are living in, life without internet has become unimaginable. There is a very efficient way to solve the problem. We can generate an efficient java k-means clustering implementation to gather wi-fi hotspots near a user and identify the nearest available hotspot.

We can create a K-means Cluster implementation in java with an uncommon equal option which enforces a constraint of equal cardinality on the clusters and still efficiency being as spatially cohesive as possible. The main issue that needs to be dealt with is to be able to achieve a variation of k-means that will ensure that each k-means cluster will have uniform cluster size and each cluster has a uniform number of points. [3] This will ensure that the user always gets best possible results in the most efficient time. The K-Means clustering algorithm these days is developed to be a mainstay in many diverse fields for the data analyst [1]. However, there is one drawback in the algorithm which happens when it is applied to datasets, with data points in large dimensional real space Rn, and the number of desired clusters is also large. When this situation arises, one or more clusters produced by K-Means algorithm often start to converge and the result is that the clusters produced are either empty or generate very few data points (i.e. one data point). [2] Equal size of cluster will ensure that the user gets more, and accurate number of results close to the user when it queries for wi-fi hotspots near it. The algorithm, in a reasonable amount of time, possess ability to do its best to regard the equal cardinality constraint, but one needs to be very careful when this option is enabled because sometimes the algorithm might generate clusters that are not as spatially cohesive and thus equally efficient like the original K-means algorithm (NP-hard problem). [3]

K-means clustering model is the best possible way to deal with the large dataset of wi-fi hotspots [4] in a city and generate an accurate navigation application that can provide intelligent solution to basic navigation queries of the people like

1. Where can the user find hotspot nearby?

2. How many wi-fi hotspots are present close to user?

3. How far does the user need to travel to get to nearest wi-fi?

This application can further be scaled to many other datasets like restaurants, grocery outlets, etc.

# Evolution of K-means Clustering

Edwin Diday, in his early phase of career, chose cluster analysis as prime research topic. He wrote a monograph (Diday et al. 1979), alongside twenty-two co-writers, that traces noticeable level of generalization of the crux of k-means and rooted its utility in model-based clustering. ’Principles of numerical taxonomy’, a monograph that Sokal and Sneath wrote in 1963 gained attention across the globe and there was a curiosity in clustering methods among researchers. It became catalyst for the publication of books like ‘Automatische Klassifikation’ written by Bock in 1974, ‘Les bases de la classification automatique’ which was written by Lerman in 1970, and ’Cluster analysis for applications’ which was written by Anderberg in 1973. Consequently, the basic ideas and functions of clustering became familiar over broad scientific communities of data analysis, decision analysis, statistics and particularly, applications. [8] K-means clustering algorithm is the most famous clustering approach being used today and it is built on sum of squares criteria. [2] If you trace back the origins of the algorithm, several scientists proposed in various forms and under diverse presumptions the same algorithm. Then many scholars like Cox in 1957, Fisher in 1958 and Bock in 1974 explored analytical and functional aspects of the method while looking for ’continuous’ analogues of the SSQ criteria. Hartigan in 1975, Pollard in 1982 and Bock in 1985 investigated the algorithm by exploring asymptotic behavior under arbitrary sampling strategies and stretching the algorithm’s sphere to probabilistic models and new data types. [8]

The k-means clustering algorithm is famous because of its simplicity. K-means clustering (MacQueen, 1967) is a machine learning algorithm usually used when we need to automatically partition data set into clusters. It starts its iteration after selecting an initial cluster center. [3] It proceeds by selecting k initial cluster centers and then iteratively refining them as follows:

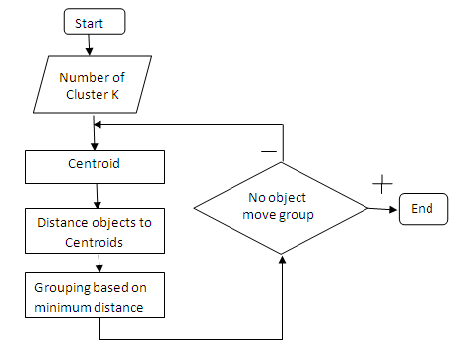
1. Each instance d, data point is allocated to its nearby cluster center.
2. Each cluster center, C is reorganized to be the mean of its integral data points. [3]

The algorithm begins to converge when there occurs no additional modification in assignment of data points to clusters. The issue is how we can choose the number of clusters k. We can use wrapper search to identify the best value for k for data-sets where optimal value of k is already known (i.e. in wi-fi hotspot dataset of city). [5] The center is chosen randomly, so the result generated varies based on what center you have chosen i.e. it will never provide us with a unique solution. It is imperative to choose initial center optimally. [6]

# K-means Algorithm and its Result

Algorithm 1: K-Means Clustering Algorithm

1. We are given a data-base D having m points in Rn and
2. Group centers are C1,t, C2,t, C3,t… Ck,t at iteration t,
3. We then evaluate C1,t+1, C2,t+1, C3,t+1… Ck,t+1 at iteration t + 1 in the following 2 steps:
   1. Cluster Assignment: For each data element xi D, allocate xi to cluster h(i) such that center Ch(i)’t is closest to xi in the 2-norm.
   2. Cluster Update: Compute Ch(i)’t+1 as the mean of all points assigned to cluster h and stop only when Ch(i)’t+1 = Ch(i)’t; where h = 1…k, else increase t by 1 and go again to step 1. [2]



Flowchart of the process of k-means clustering Algorithm.

The results of the K-means clustering algorithm are:

1. The centroids of the K groups, which can be used to mark new information
2. Tags for the training statistics (i.e. each data point is allocated to a single cluster)

Instead of defining clusters before observing the data-set, k-means clustering permits you to search and examine the clusters that have formed organically. Each centroid of a group is an assortment of feature values which identifies the resulting clusters. [5,7]

There may be some cluster h which are empty when Algorithm terminates. The solution computed by Algorithm in this case satisfies the Karush-Kuhn-Tucker or KKT conditions [7] for (2). Hence, it is reasonable to trust that the standard K-Means algorithm might converge with empty clusters. In real life, we witness this phenomenon when we cluster high-dimensional datasets with huge number of clusters. In order to avoid clustering results with empty clusters, we can explicitly add constraints requiring that cluster should contain at least some data points. [2,8]

The basic algorithm to implement K-means clustering with Special Equal Option is as below:

* Find the desired cluster size using wrapper search.
* Initialize the mean elements for the clusters and select the equal option.
* Order data points by their distance to their nearest mean element and minus distance to the farthest mean element
* Assign data points to the ideal cluster until the cluster is full to the maximum allowed elements, then again sort outstanding data points, this time not taking the full cluster into account anymore. [2,5]

# Benefits of using K-means clustering

K-means algorithm is highly user-friendly and effortlessly comprehensible. If data-sets are defined well and they are classified properly, then this algorithm provides excellent clustering. [5,6] The k-means clustering algorithm is famous because of its ease and it is usually used when we need to automatically partition data set into clusters. It starts its iteration after selecting an initial cluster center. [3] K-means clustering with special equal option will not have the convergence property i.e. clusters are equal and uniform to each other and they are also not hierarchical in nature. [5,6] When variables are large in number and the dataset is huge, then K-Means clustering is computationally quicker than hierarchical clustering, if we identify the right value for k. Also, if we have globular clusters, K-Means produces tighter clusters when compared to hierarchical clustering. [2]

# Points to be careful about when using K-means

The equal option is investigational. One ought to know that if you select this option the produced clusters may not be spatially cohesive as the original k-means would generate. However, the algorithm will do its finest, in sufficient time, to respect the equal cardinality constraint. [5]

It is oversensitive to primary cluster centroid selection, because of which it delivers the outcome that is local optimal. The most popular clustering technique k-means algorithm, exhibits local minima problem due to initial center selection. [6]

The k-means algorithm gets more complicated than other algorithms because it requires the number of clusters to be known in advance. Kiri Wagstaff et.al [5] showcased a way in which wrapper search can be used to identify the optimal value for k for data-sets where optimal value of k is already known (i.e. in wi-fi hotspot dataset of city). K-means algorithm can also suffer from empty cluster issue while executing if there are no data points that are assigned to a cluster in the assignment stage. [6] In order to avoid clustering results with empty clusters, we can explicitly add constraints requiring that cluster should contain at least some data points.

# Conclusion

One can easily create clusters of wi-fi hotspots in the city using the data available publicly using the k-means algorithm. However, it is vital that the user gets optimal results and k-means clustering implementation is efficient. When special equal option is enabled for k-means, user will not face a denial of service. However, equal option is hyper-sensitive and may produce results that are not optimal or spatially cohesive. The main area to work upon is to be able to achieve a variation of k-means that will ensure that each k-means cluster will have uniform cluster size and each cluster has a uniform number of points.

The next area of focus will be that the k-means algorithm gets more complicated than other algorithms because it requires the number of clusters to be known in advance. Implementing wrapper search to fetch the best possible number of clusters will be key. K-means algorithm can also suffer from empty cluster issue while executing if there are no data points that are assigned to a cluster in the assignment stage. It is oversensitive to primary cluster centroid selection, because of which it delivers the outcome that is local optimal.

# References

1. R. O. Duda and P. E. Hart. Pattern Classication and Scene Analysis. John Wiley & Sons, New York, 1973
2. Bradley, P. S., K. P. Bennett, and Ayhan Demiriz. "Constrained k-means clustering." Microsoft Research, Redmond (2000): 1-8.
3. Raykov YP, Boukouvalas A, Baig F, Little MA (2016) What to Do When K-Means Clustering Fails: A Simple yet Principled Alternative Algorithm. PLoS ONE11(9): e0162259. <https://doi.org/10.1371/journal.pone.0162259>
4. City of New York (2017) New York City wi-fi Hotspots Available: https://www.kaggle.com/new-york-city/nyc-public-wifi/version/1/data
5. Kiri Wagstaff, Claire Cardie, Seth Rogers, and Stefan Schrödl. 2001. Constrained K-means Clustering with Background Knowledge. In Proceedings of the Eighteenth International Conference on Machine Learning (ICML '01), Carla E. Brodley and Andrea Pohoreckyj Danyluk (Eds.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 577-584
6. V. S. Chandrawanshi, R. K. Tripathi and N. U. Khan, "A comprehensive study on k-means algorithms initialization techniques for wireless sensor network," 2016 International Conference on Signal Processing and Communication (ICSC), Noida, 2016, pp. 154-159. doi: 10.1109/ICSPCom.2016.7980567
7. O. L. Mangasarian. Nonlinear Programming. McGraw{Hill, New York, 1969. Reprint: SIAM Classic in Applied Mathematics 10, 1994, Philadelphia
8. H. H. Bock (2007) Clustering Methods: A History of k-Means Algorithms. In: P. Brito, G. Cucumel, P. Bertrand, F. de Carvalho Selected Contributions in Data Analysis and Classification. Studies in Classification, Data Analysis, and Knowledge Organization. Springer, Berlin, Heidelberg