K-MEANS CLUSTERING, AN IMPROVED SEEDING PROCESS

by

Yaser Alkayale

Submitted in partial fulfillment of the requirements
for the degree of Bachelor's of Computer Science, Honours

at

Dalhousie University
Halifax, Nova Scotia
April 2018

This thesis is in dedication to my grandfather whom I was named after.

Table of Contents

List of Figures

Abstract

Clustering is a well-known task that has been studied and used for decades. The idea is to take a set of items and group them into a number of clusters based on a similarity measure. K-means proposed in 1957 by Stuart Lloyd is one of the most widely used clustering algorithm and is still used today for its reasonably fast heuristic to find the clusters based on the Lloyd algorithm and more recent developments in that area. K-means has two main parts to clustering, the initial seeding process and the iteration process. The seeding process picks k initial seeds as cluster centres, and highly affects the accuracy of the final result in the algorithm. The iteration process dominates running time to move the centres around until it converges to an optimum. In this paper, we discuss a new method of the seeding process that gives us more accurate seeds to start the algorithm. We also discuss a novel approach to find an approximation of the correct number of clusters for a given dataset.

Acknowledgements

Big thank you to my supervisor Dr. Norbert Zeh. Without his assistance, this project would not have seen the day of light. Thank you to Dr. Vlado Keselj who made himself available when we needed to consult with him. Also big thank you to Arazoo who was with me from the beginning, and went through my ideas with me.

Chapter 1

Introduction

Clustering problems arise in many domains like natural language processing[5], crash report analysis[7], and vehicle navigation[4]. The notion of what is a good cluster highly depends on the domain and application at hand. Many clustering techniques like hierarchical clustering[2] and graph-based techniques[6], each serving a different purpose. K-means clustering continues to be one of the most clustering algorithms for it's simplicity of implementation and relative efficiency[3].

Formally, K-means is the problem where we are given a set of n points in d-dimensional space and a number k, where we are asked to split n points into k clusters while minimizing the total sum of distances from points to their cluster centres. K-means does especially well on convex shaped clusters as it minimizes the sum of distances of all points to their belonging cluster centres.

While K-means is widely used for its efficiency and simplicity of implementation, it is not perfect for many use cases. The algorithm can give highly inaccurate clusters if the number k is not known in advance. This is due to the nature of the algorithm forcing all the points into k clusters. Another problem with the algorithm is that it works best on convex shaped clusters, and struggles when there is a lot of noise[1] in the data. Having noisy data or the incorrect number of clusters k, the K-means algorithm struggles with running time, as it requires a large number of iterations and sometimes never converges to an optimum.

In this paper, we will introduce a new way to seed the K-means algorithm that gives us better initial seeds to help converge the algorithm with fewer iterations and a lower objective function.

[1]Note: Noise in this context is referring to the overlap of multiple clusters, where it becomes difficult to separate points if they are at the edges of two or more clusters.

Chapter 2

Background

2.1 Seeding Process

The seeding process of the k-means algorithm is crucial to the accuracy of the result because while the iterations converge the centres to an optimum, it is mostly localized to the regions constrained by the initial seeds[1]. Many new approaches have been introduced like K-means++[1], and Kmeans||. It has been experimentally proven that having a better seeding process improves the algorithm by both lowering the objective function of the algorithm, and allowing it to converge faster with less iterations and running time.

2.1.1 K-means++

K-means++ uses an intuitive way to seed the initial clusters of the algorithm by trying to pick seeds that are as far apart as possible. This is done by giving a higher probability for points that are further away from the ones already picked. The way it works is that we pick a random initial point, and then given a probability of picking a point proportional to the seeds already picked, until k seeds are picked.

K-means++ works very well because it about evenly distributes the initial seeds in the dataset. Also, because a probablistic model is used, then a higher concentration of points get a high probability of getting at least one point in them.

2.1.2 K-means|| (parallel)

2.2 Iterations

The iteration process of the k-means clustering algorithm dominates the running time of the algorithm

2.3 Objective Function

The objective function of the clustering task determines what we are trying to group points based on. In most tasks, including in everything discussed in this paper, the objective function is the sum of squared distances[1] of all the points to their cluster centres. More formally expressed as:

$$\underset{r}{\operatorname{argmin}} \sum_{i=1}^{k} \sum_{x \in S_i} \|x - u_i\|^2$$

2.4 Difficulties

K-means clustering is very good at clustering well-separated convex clustered where k the number of clusters in known or accurately measured in advance. However, this is very rarely the case.

[1]Note: Here we used squared distance because we are merely comparing the distances to each other and do not care what the actual values are. Keeping them squared allows us to same on the computational cost of taking the square root of all the distances before summing.

Chapter 3

Minimum Weight Perfect Matching on Bipartite Graphs

Perfect matching is the problem where a graph is partitioned into pairs based on a heuristic. Minimum weight perfect matching is where the total costs of the edges connecting the pairs in the graph are minimized. The is an NP-hard problem on a general graph; however, it can be solved in $O(n^3)$. The Hungarian algorithm is one of the best known problems to solve the assignment problem, the more widely used name of the minimum weight perfect matching problem.

The assignment problem is where given a set of w workers and t tasks, we are asked to find the

Chapter 4

Datasets

To test how effective our new techniques are, we

4.1 Generated Artificial Datasets

The datasets were generated using

4.2 Real World Datasets

Chapter 5

Future work

5.1 Moving On

While k-means is a great clustering method for it's efficiency and simplicity, it is definitely not the greatest algorithm to be used for every single clustering task.

Chapter 6

Conclusion

Chapter 7

Extras

7.1 Getting Ready

Get all the parts that I need. I can throw in a whole pile of terms like preparation, methodology, forethought, andd analysis as examples for me to use in the future.

7.2 Next Step

Do it!

Of course, you have to have pictures to show how you did it to make peoplee understand things better. Get it done! Use reference material by Limpet [3] or Gooses, Mittelback, and Samarin .

This following line should be exactly 5cm long. It can be used to check the typesetting process.

Did it!!

Bibliography

[1] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

[2] Florence Corpet. Multiple sequence alignment with hierarchical clustering. *Nucleic acids research*, 16(22):10881–10890, 1988.

[3] Anil K Jain. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666, 2010.

[4] Dario Maio, Davide Maltoni, and Stefano Rizzi. Dynamic clustering of maps in autonomous agents. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(11):1080–1091, 1996.

[5] Deepak Ravichandran, Patrick Pantel, and Eduard Hovy. Randomized algorithms and nlp: using locality sensitive hash function for high speed noun clustering. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 622–629. Association for Computational Linguistics, 2005.

[6] Satu Elisa Schaeffer. Graph clustering. *Computer science review*, 1(1):27–64, 2007.

[7] Axel J Soto, Ryan Kiros, Vlado Keselj, and Evangelos Milios. Machine learning meets visualization for extracting insights from text data. *AI Matters*, 2(2):15–17, 2016.