Scientific Journal of Impact Factor (SJIF): 4.72

e-ISSN (O): 2348-4470
p-ISSN (P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 4, Issue 10, October -2017

# A Survey on k-means

Pranjali Shashikant Kothawade

*Department of Computer Engineering, Bharti Vidhyapeeth College Of Engineering Pune.*

**Abstract** —*K-means clustering is a variety of unsupervised learning, which is utilized when you have unlabeled information (i.e., information with no defined classes or groups). The objective of this algorithm is to discover groups in the information, with the amount of groups represented by the variable K. The algorithm performs iteratively to assign every single information level to one particular of K groups primarily based on the attributes that are offered. Information factors are clustered based mostly on attribute similarity. The outcomes of the K-means [1] clustering algorithm are:1. The centroids of the K clusters, which can be utilized to label new information. 2. Labels for the instruction information (each information level is assigned to a single cluster). Rather than defining groups prior to seeking at the information, clustering enables you to discover and analyze the groups that have formed organically. The "Choosing K" segment below describes how the variety of groups can be established. Every single centroid of a cluster is a assortment of attribute values which define the resulting groups. Examining the centroid attribute weights can be utilized to qualitatively interpret what variety of group every single cluster represents.*

## I. INTRODUCTION

*K*-means clustering is a type of unsupervised learning, which is used when you have unlabeled data (i.e., data without defined categories or groups). The goal of this algorithm is to find groups in the data, with the number of groups represented by the variable *K*. The algorithm works iteratively to assign each data point to one of *K* groups based on the features that are provided. Data points are clustered based on feature similarity. The results of the *K*-means clustering algorithm are:
1. The centroids of the *K* clusters, which can be used to label new data.

2. Labels for the training data (each data point is assigned to a single cluster)

Rather than defining groups before looking at the data, clustering allows you to find and analyze the groups that have formed organically. The "Choosing K" section below describes how the number of groups can be determined.
Each centroid of a cluster is a collection of feature values which define the resulting groups. Examining the centroid feature weights can be used to qualitatively interpret what kind of group each cluster represents.

This introduction to the *K*-means clustering algorithm covers:

- Common business cases where *K*-means is used
- The steps involved in running the algorithm
- A Python example using delivery fleet data

The *K*-means clustering algorithm is used to find groups which have not been explicitly labeled in the data. This can be used to confirm business assumptions about what types of groups exist or to identify unknown groups in complex data sets. Once the algorithm has been run and the groups are defined, any new data can be easily assigned to the correct group.
The *K*-means clustering algorithm uses iterative refinement to produce a final result. The algorithm inputs are the number of clusters *K* and the data set. The data set is a collection of features for each data point. The algorithms starts with initial estimates for the *K* centroids, which can either be randomly generated or randomly selected from the data set. The algorithm then iterates between two steps:

1. Data assigment step:
Each centroid defines one of the clusters. In this step, each data point is assigned to its nearest centroid, based on the squared Euclidean distance. More formally, if $c_i$ is the collection of centroids in set $C$, then each data point $x$ is assigned to a cluster based onwhere $dist(\cdot)$ is the standard ($L_2$) Euclidean distance. Let the set of data point assignments for each $i^{th}$ cluster centroid be $S_i$.

 

2. Centroid update step:
In this step, the centroids are recomputed. This is done by taking the mean of all data points assigned to that centroid's cluster.

The algorithm iterates between steps one and two until a stopping criteria is met (i.e., no data points change clusters, the sum of the distances is minimized, or some maximum number of iterations is reached).
This algorithm is guaranteed to converge to a result. The result may be a local optimum (i.e. not necessarily the best possible outcome), meaning that assessing more than one run of the algorithm with randomized starting centroids may give a better outcome.Choosing *K*-The algorithm described above finds the clusters and data set labels for a particular pre-chosen *K*. To find the number of clusters in the data, the user needs to run the *K*-means clustering algorithm for a range of *K* values and compare the results. In general, there is no method for determining exact value of *K*, but an accurate estimate can be obtained using the following techniques.
One of the metrics that is commonly used to compare results across different values of *K* is the mean distance between data points and their cluster centroid. Since increasing the number of clusters will always reduce the distance to data points, increasing *K* will *always* decrease this metric, to the extreme of reaching zero when *K* is the same as the number of data points. Thus, this metric cannot be used as the sole target. Instead, mean distance to the centroid as a function of *K* is plotted and the "elbow point," where the rate of decrease sharply shifts, can be used to roughly determine *K*.A number of other techniques exist for validating *K*, including cross-validation, information criteria, the information theoretic jump method, the silhouette method, and the G-means algorithm. In addition, monitoring the distribution of data points across groups provides insight into how the algorithm is splitting the data for each *K*.

## II. LITERATURE SURVEY

**Paper Name:Clustering Short Texts using Wikipedia**
**Authors:**Somnath Banerjee, Krishnan Ramanathan, Ajay Gupta
**Description:** A method of improving theaccuracy of clustering short texts by enriching theirrepresentation with additional features from Wikipedia.Empirical results indicate that this enriched representation oftext items can substantially improve the clustering accuracywhen compared to the conventional bag of words representation.

**Paper Name: Data clustering: 50 years beyond K-means**
**Authors:** Anil K. Jain
**Description:** A brief overview of clustering, summarize well known clustering methods, discuss the majorchallenges and key issues in designing clustering algorithms, and point out some of the emerging anduseful research directions, including semi-supervised clustering, ensemble clustering, simultaneous featureselection during data clustering, and large scale data clustering.

**Paper name: Semi-supervised graph clustering: a kernel approach**
**Authors:**Brian Kulis, SugatoBasu,InderjitDhillon, Raymond Mooney
**Description:**We unify vector-based and graph-based approaches. We first show that a recently-proposed objective function for semi-supervised clustering based on Hidden Markov Random Fields, with squared Euclidean distance and a certain class of constraint penalty functions, can be expressed as a special case of theweighted kernel *k*-means objective.

## III. CONCLUSION AND FUTURE SCOPE

We conclude that the highlights statistical enhancements on the k-means algorithm and proves their efficiency via a statistical examine on a recognized database. In this paper we talked about the ideas of information mining, cluster evaluation and mainly we centered on presenting extensively the k-means algorithm, the two from the mathematical point of view and from the level of view of its implementation. Despite the issues presented in Segment three, k-means stays the most broadly utilized partitioned clustering algorithm in practice. The algorithm is easy, simply understandable and fairly scalable, and can be effortlessly modified to deal with streaming data. To deal with extremely huge datasets, significant hard work has also gone into additional speeding up k-means, most notably by making use of kd-trees or exploiting the triangular inequality to stay away from evaluating each and every information level with all the centroids throughout the assignment stage. Continual enhancements and generalizations of the fundamental algorithm have ensured its continued relevance and progressively increased its effectiveness as nicely.
For long term we aim to target on the comparison of statistically viewpoint of the clustering technique which involves k-means algorithm with the classification technique which involves k-nearest neighbors' algorithm. We also started out investigating a method to applying the k-means algorithm in the domain of the topographic engineering, especially for building topographical maps.

 

**REFERENCES**

[1] Jain, Anil K. "Data clustering: 50 years beyond K-means." *Pattern recognition letters* 31.8 (2010): 651-666.

[2] Kulis, Brian, et al. "Semi-supervised graph clustering: a kernel approach." *Machine learning* 74.1 (2009): 1-22.

[3] Aggarwal, Charu C., and Chandan K. Reddy, eds. *Data clustering: algorithms and applications*. CRC press, 2013.

[4] Basu, Sugato, Ian Davidson, and KiriWagstaff, eds. *Constrained clustering: Advances in algorithms, theory, and applications*. CRC Press, 2008.

[5] Wolkowicz, Henry, RomeshSaigal, and LievenVandenberghe, eds. *Handbook of semidefinite programming: theory, algorithms, and applications*. Vol. 27. Springer Science & Business Media, 2012.

[6] Banerjee, S., Ramanathan, K., Gupta, A., *Clustering short texts using Wikipedia.* In: Proc. *SIGIR.* 2007b.

[7] Dhillon, Inderjit S., Guan, Yuqiang, Kulis, Brian, *Kernelk-means: Spectral clustering and normalized cuts.* In: Proc. 10th KDD, pp. 551–556.2004.

[8]Har-peled, Sariel, Mazumdar, Soham, *Coresets for k-means and k-medianclustering and their applications.* In: Proc. 36th Annu. ACM Sympos. TheoryComput., pp. 291–300. 2004.

[9] E. Gabrilovich. *Feature Generation for Textual InformationRetrieval Using World Knowledge*. PhD Thesis, Departmentof Computer Science, Technion – Israel Institute ofTechnology, Haifa, Israel, 2006.

[10] A. Hotho, S. Staab, and G. Stumme. *Ontologies ImproveText Document Clustering*, In the Proc of the Third IEEEInternational Conference on Data Mining (ICDM'03),Melbourne, Florida, USA, 2003.

 