Analysis of Fuzzy K-means Clustering

By Jonathan Koller

CS 608. Fall 2018

***Definition of k-means***

K-means clustering is a type of unsupervised learning, which is a machine learning method. In machine learning, the algorithms used are typically called models. For this model, the data is unlabeled, which means there are no pre-defined classifications; the data is truly “raw.” Consequently it is the task of the model to define these classes [1].

The k-means model iterates over every member of the data set, assigning it to one of k clusters based on the features that are provided based on total similarity. The output for this includes both center points of the clusters and labels for training data [1].

***Soft/fuzzy k-means***

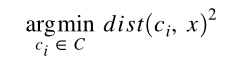
This is a variant of the k-means algorithm in which we take into account the susceptibility of the model to define points sharply according to local minima. It also attempts to ease the fact that k-means are highly sensitive to initialization and may need to restart multiple times and use results that best fit the purpose of its implementation [2].

As such, fuzzy membership in each class is used. This means that each data point doesn't belong to one class solely; memberships lie along spectra. For example, a data point could feasibly land with 60% dog, 20% cat, and 10% fish classification weights [2].

***Theory***

Step 1: Data assignment

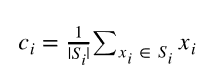
If ci is the collection of centroids in C, each data point x is assigned to a cluster based on:



This is the min of dist(), which is the squared Euclidean distance.

Step 2: Centroid update

Centroids are recomputed using mean of all data points assigned to a given centroid cluster:



The model iterates over the data set until either no data points change clusters or the sum of all distances is minimized [1].

***Fuzzy K-Means Pseudocode and Explanation***

1) Initialize m1,…mk to random points in X

2) Loop:

1. Cluster responsibilities is a fraction (0 or 1 in hard or regular k means)

-Depends on distance between data point and cluster center

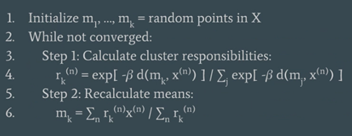
-Fuzziness affords confidence of each points’ memberships

-Numerator resembles Gaussian variance which measures influence of cluster on a point

2. Weighted mean: Recalculate means based on responsibilities

-Generalization of traditional mean but you consider that the weight of each data point

3) Purpose: assigns confidence in a cluster assignment

[2]

***Cost function: Linear Regression for Minimizing Cost***, J, is squared distance weighted by the responsibilities of X.

In this last step, the model compares the difference between hypothetical values and real values by using weights. This is called coordinate descent.

In this process, the model is moving in direction of a smaller J function with respect to only one variable at a time. As it iterates, this function will decrease in cost, which guarantees a convergence, or the point at which valuable data has been inferred. The rapidity of convergence in k-means clustering makes it a key tool in any classification task where a handful of models are used to test hypotheses.

[2]

(Sum of n of Sum of k of responsibilities times squared distance between means and X)

***Analysis***

Please see comments in attached Python file for an implementation and detailed asymptotic analysis. A notebook can also be accessed at [bit.ly/2A8mg3P](https://bit.ly/2A8mg3P).

***Conclusion***

Constraints on input data abound with the current implementation. It is only viable for using 2- or 3D responsibility matrices in terms of K clusters times N samples. In this way, it hinges on a limited analysis of 2 or 3 clusters. Furthermore, the amount of iterations, or epochs, which defines the range of the very inefficient outer loop in the model, is also worth noting as the algorithm can be immensely complex. Though it isn’t unreasonable to consider a lower sample size to increase K by a few integers, this is a fairly specific use-case.

Other limitations include the fact that K must be pre-defined; the model itself will not generate this value. Theoretically, the dimensions can be higher than 2 or 3 but if the input data itself has too many dimensions, the plot can look very unintuitive [2]. However it is worth recalling that computational difficulties are present as matrix multiplication is currently nested in two loops.

Optimizations in terms of raw runtime could include parallelization of the first and second steps involved in this model. Subgroups of the sample collection could be fed through the classification step of the algorithm, collected, and then run again in parallel batches to the means calculation.

**Reference**

[1] Oracle: *Introduction to K-Means Clustering* ­­­by Andrea Trevino. Retrieved from:

[datascience.com/blog/k-means-clustering](https://www.datascience.com/blog/k-means-clustering%20)

[2] Lazy Programmer Inc./Udemy.com: *Cluster Analysis and Unsupervised Machine Learning in Python*. Retrieved from:

[udemy.com/cluster-analysis-unsupervised-machine-learning-python](https://www.udemy.com/cluster-analysis-unsupervised-machine-learning-python)

**Reference in Code and Analysis**

[1] Python.org: *TimeComplexity*. Retrieved from: [wiki.python.org/moin/TimeComplexity](https://wiki.python.org/moin/TimeComplexity)

[2] SciPy.org: *numpy.ndarray.shape*. Retrieved from:

[docs.scipy.org/doc/numpy-1.13.0/reference/generated/numpy.ndarray.shape.html](https://docs.scipy.org/doc/numpy-1.13.0/reference/generated/numpy.ndarray.shape.html)

[3] Github.com/numpy/: *core.py.* Retrieved from:

[github.com/numpy/numpy/blob/master/numpy/ma/core.py](https://github.com/numpy/numpy/blob/master/numpy/ma/core.py)

[4] Sanjiv K. Bhatia/University of Missouri-St. Louis: *Matrix-Vector* Multiplication. Retrieved from:

[cs.umsl.edu/~sanjiv/classes/cs5740/lectures/mvm.pdf](http://www.cs.umsl.edu/~sanjiv/classes/cs5740/lectures/mvm.pdf%20)