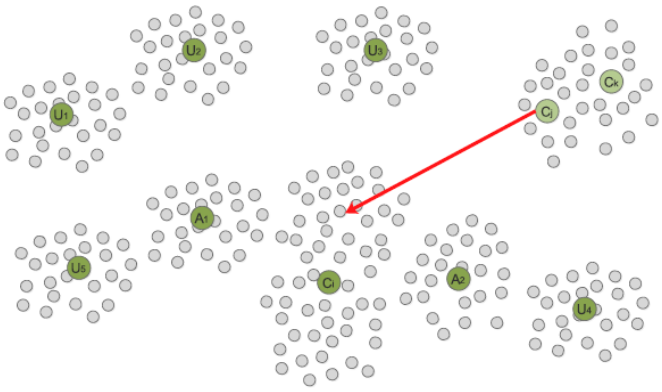
UPC —— UB —— URV

Master in Artificial Intelligence

Unsupervised Learning

Coursework Option B: I-K-Means-+



Authors:
Marc Asenjo

Jume 2019

Abstract

This report focuses on explaining the characteristics of the algorithm of I-K-Means-+, which has been implemented in Python, and compare the results obtained with it in several datasets to other clustering methods.

Contents

1 Introduction

The objective of this Coursework was to choose a state of the art algorithm without implementation, carefully read and understand the references that describe it, implement the algorithm itself, and finally compare the results obtained with it to other methods, in order to test the results from the initial publication. The algorithm chosen is I-k-Means-+, a clustering technique which will be described next. Its implementation has been done following the `scikit-learn` library API conventions, as well as making it as efficient as possible to be able to add to the comparison the execution time. These comparisons will be seen in the following section after. Finally, the experience of the work will be concluded with some last words.

2 I-k-Means-+

In this section the algorithm proposed by the chosen article [2] will be summarized. Then, a little note about the considerations or problems that have been encountered in its implementation will be mentioned.

2.1 Algorithm

The I-k-Means-+ algorithm stands for Iterative k-means Minus-Plus. It proposes an iterative method to improve k-means results so that the initial cluster centers chosen don't have an effect on the final results. This iterative method uses the called "Minus-Plus" stage which consists on the removal of a cluster (Minus) and then the division of another one into two (Plus), thus keeping the same amount of clusters but relocating them.

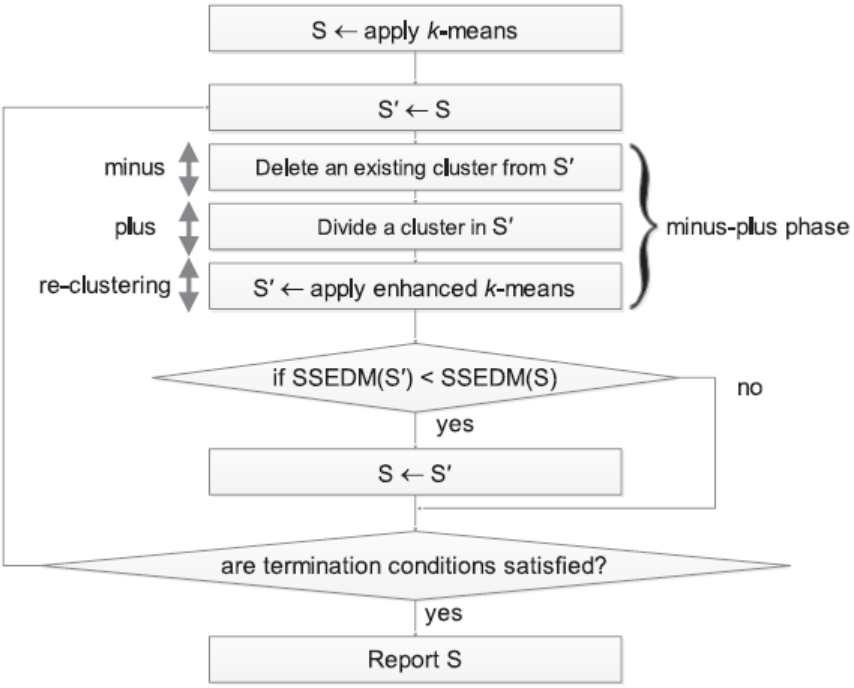


Figure 1: A view of the I-k-means-+ algorithm [2]

An image summarizing how the algorithm works can be seen in Figure 1. The general idea is to post-process the results obtained from a normal k-means algorithm using the mentioned mechanism. Consequently, it is necessary to first execute a normal k-means algorithm. In this case, the initialization used for this execution is the one proposed in another paper from the same author [1], which revolves around the idea of Useful Nearest Centers (UNC) of points. It starts by selecting a "random" center (the one with the smaller value in the first dimension) and then iteratively selects the others by the use of an heuristic value which depends on the distances to the so-called UNCs, which are also defined.

After this first k-means execution, the real iterative algorithm begins. In each iteration, a cluster to be removed and another one to be divided are chosen, the minus-plus operations are made and finally the obtained result is brought to an optimum by applying another k-means algorithm initialized with what we have. If the results obtained are better than

the ones we had before in terms of comparing the Sum of Squared Distances, the solution is updated. One thing to note here is that the k-means algorithm used to refine each new solution is not the basic one used before, but a more optimal one for the occasion. It is based on *Topical k-means* (t-k-means), which only updates and makes calculations for the points that are detected as "active", which means that they are affected by the changes that have been done. This is useful in this case as the changes consist on a removal and a division, which will keep most of the centers untouched in big databases with large amounts of clusters.

Getting a little bit more into detail, the main iteration of the algorithm works in a more complex way than it looks at first sight. It keeps track of indivisible and irremovable clusters for example, which is determined based on past modifications and the definitions of *cluster adjacency and strong adjacency*. Furthermore, to be able to choose adequate clusters for each of the operations of the minus-plus stage, two new heuristics are used, called *cost and gain*, which approximate the cost of removing a cluster and the gain of dividing one into two respectively and on terms of the Sum of Squared Distances. Once a pair of clusters is successfully selected, t-k-means can be applied and all sorts of restrictions for future selections can be drawn.

Finally, once the solution is thought to be impossible to improve anymore by several metrics, the final solution is returned by the algorithm, consisting for example of the cluster centers and the point's assignation to them.

2.2 Implementation

The following are elements that were taken into account during the implementation stage of the algorithm as it has been explained:

- For all development in general, the definitions and procedures presented by the paper [2] have been closely followed. These include pseudo-code of the different sub-algorithms used. In some cases, however, some little mistakes or inconsistencies were found on them, in which case the correct way to apply them was thought about and implemented separately.
- The first thing to be implemented was the basic k-means algorithm, but in a way thought out to make the future t-k-means algorithm able to reuse parts of it. Because of the use on many parts of the general algorithm of the second closest center to each point and separate Sum of Squared Distances for each of the clusters, these concepts where carried onto all k-means algorithms too for re-usability purposes.
- The implementation of the UNC initialization was a whole new algorithm on its own, which also follows the original description for it and is implemented efficiently.
- Many small functions are used like the calculation of the *cost* or the *gain* so that the main code is more readable and, to the maximum extent, easily modifiable and maintainable.
- All the implementation is done following the `scikit-learn` API conventions: having a main class that needs to be initialized and with the functions *fit, predict and* $fit_predict, which call to all main methods described above.$
- As we will see, for testing purposes the possibility of obtaining results using partial stages of the algorithm has been implemented.

3 Results

4 Conclusions

References

[1] Hassan Ismkhan. An initialization method for the k-means using the concept of useful nearest centers, 2017.

[2] Hassan Ismkhan. I-k-means+: An iterative clustering algorithm based on an enhanced version of the k-means. *Pattern Recognition*, 79:402 – 413, 2018.