UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Practical Work 1: An extended version of the k-means method for overlapping clustering

Unsupervised and Reinforcement Learning

João Valério

joao.agostinho@estudiantat.upc.edu

09/05/2023

An extended version of the k-means method or overlapping clustering

Index

1. INTRODUCTION

The main aim of the present research endeavour is to re-implement the algorithmic methodology put forth by Guillaume Cleuziou in his seminal work titled "An extended version of the k-means method for overlapping clustering," published in 2008.

The algorithmic framework, named Overlapping K-Means (OKM), is a comprehensive extension of the classical K-Means algorithm. The study seeks to empirically demonstrate that OKM is a viable and superior alternative to other existing methods, particularly in the domains of information retrieval and biology.

The proposed technique is a centroid-based approach that expands upon the k-means algorithm by introducing a novel objective function. This function is designed to minimize multiple assignments while adhering to specific constraints. Furthermore, the method emphasizes space coverage instead of space partitioning, a deviation from the traditional K-means approach.

The datasets employed in the study comprise authentic data samples sourced from the domains of Information Retrieval and Biology. The use of such datasets reinforces the notion that an overlapping clustering approach is well-suited for these domains. Specifically, the datasets under consideration are Reuters-21578[1] (Information Retrieval) and Yeast[2] (Biology). It is imperative to note that the latter dataset does not correspond to the original dataset used in the experiment, as access to the original dataset, through the provided links, was no longer available. Consequently, the veracity of obtaining the dataset through a distinct web source was questionable. Additionally, the report provides a comprehensive description of any preprocessing techniques employed on the datasets.

Furthermore, both the K-Means (KM) algorithm, which employs crisp partitioning, and the Fuzzy-K-Means (FKM) algorithm have been utilized for the purpose of conducting a comparative analysis of the results obtained from the OKM algorithm. With respect to evaluation, the external criterion employed in this research endeavour is F-Measure. This criterion is utilized to compare the clustering methodologies employed in this study by assessing the degree of correspondence between the clustering outcomes and the true labels.

Moreover, a comprehensive analysis of the research conducted will be presented, comprising a critical appraisal of the advantages and limitations of the algorithm proposed in [1]. Additionally, a critique of the techniques and methodologies employed in [1] will be provided.

Finally, the implementation of the proposed algorithm in this study will be elucidated in detail, along with all relevant sources and prerequisites necessary for replicating the study. However, it is important to note that the methodology employed in this study will be similar to that described in [1].

[1] The Reuters-21578 dataset can be found in the following link: https://www.kaggle.com/datasets/nltkdata/reuters

[2] The Yeast dataset can be found in the following link: https://www.uco.es/kdis/mllresources/#GenbaseDesc

2. DATA CHARACTERISTICS

a. Reuters-21578

The Reuters-21578 dataset is a widely used benchmark dataset in the field of information retrieval. It was released by Reuters in 1987 and consists of news articles from the Reuters newswire service. The dataset includes a total of 21,578 documents that are labeled with one or more of 114 different topics. These categories cover a wide range of topics, including business, finance, sports, and politics, among others.

b. Preprocessing: Reuters-21578

To perform the preprocessing of the Reuters-21578 dataset, 3 main subsets are formed, along with distinct characteristics:

- **Reuters-1**: contains 1156 articles having at least one tag among the set of 10 categories {gold, ipi, ship, yen, dlr, money-fx, acq, rice, grain, crude}.
- **Reuters-2**: contains 1308 articles having at least one tag among the set of 10 categories {coffee, sugar, trade, rubber, earn, cpi, cotton, alum, bop, jobs}.
- **Reuters-3**: contains 333 articles having at least one tag among the set of 10 categories {gnp, interest, veg-oil, oilseed, corn, nat-gas, carcass, live-stock, wheat, soybean}.

Subsequently, for every subset, a feature set is extracted by selecting tokens that appear in no less than three articles. The ultimate stage involves reparameterization with the Latent Semantic Analysis (LSA) method from the scikit-learn library, which provides semantic indexing for each document. It is imperative to note that the number of components used in this process has not been specified in reference [1]. However, given the available resources, this study has considered ten components, which facilitates a comprehensive understanding of the disparities between the implemented algorithms.

c. Yeast

The Yeast dataset is a widely used benchmark dataset in the field of bioinformatics for predicting the functional categories of genes in the yeast Saccharomyces cerevisiae. The dataset was first introduced by Elisseeff and Weston in 2001 and contains microarray gene expression data and phylogenetic profiles for 2,417 yeast genes. Each gene is annotated with a subset of 14 functional categories, representing the top level of the functional catalogue, such as Metabolism, Energy, and Cell Cycle.

d. Preprocessing: Yeast

To obtain a suitable dataset for conducting clustering analysis, it is imperative to undertake data preprocessing. This domain can be segmented into three core categories, namely: Data Upload (i.), Missing Values (ii.), and Dissimilar Ranges (iv.). The aforementioned categories have been listed in a sequential order that aligns with their respective codes. Owing to the fact that all the features in the dataset are numerical, there is no need to account for different types of features.

i. Data Upload

Concerning the data upload process, it is noteworthy that the preprocessing code possesses the capability to read files in the .arff format, as this is the file format in which the Yeast dataset is available.

ii. Missing Values

In the initial stage, it is imperative to address the issue of missing data, wherein two approaches may be considered: deletion or imputation. Given that the removal of observations leads to a loss of crucial information, imputation has been chosen as the preferred approach. This methodology is expected to retain the information of the observations without introducing any potential bias into the dataset.

Numerical Data:

For the numerical data, the considered metric is the K-Nearest Neighbours Imputer, in which each sample's missing values are imputed according to the mean of the k-nearest neighbours considered. As the function considered from the sckit-learn library is optimised to the general cases of numerical imputation, the best parameters among the tested are the default ones. According to that, k assumes a value of 5, with uniform weight distribution between the neighbours.

iii. Different Ranges

Then, the normalisation of all the numerical data is executed. Since different features have distinct numerical ranges, the weights between them are non-uniformly distributed, inserting biased pieces of information in the model. As there is no relevant information pointing out that certain features should have more weight than others, it is implemented a uniform weight distribution along the attributes.

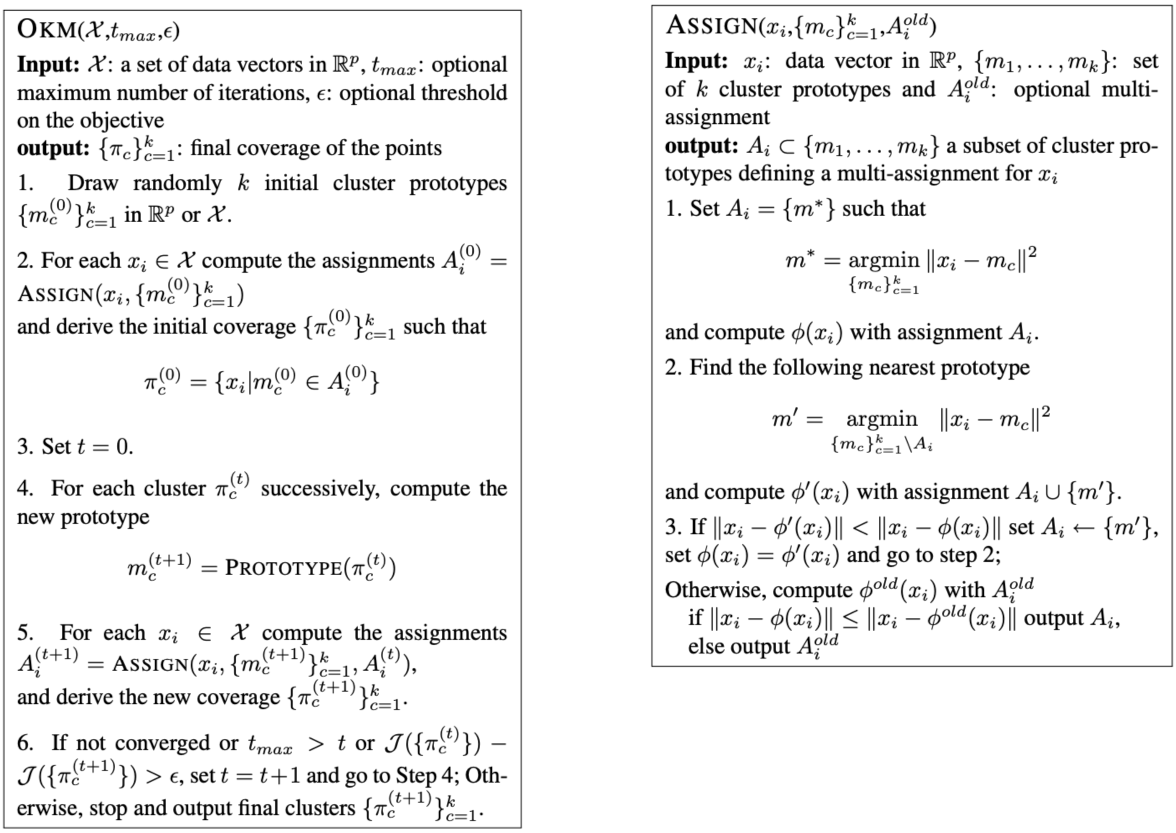
The method considered is Min-Max Scaling, in which each instance has a linear value attribution between 0 (minimum) and 1 (maximum).

3. OKM ALGORITHM

The algorithmic framework, named Overlapping K-Means (OKM), is a comprehensive extension of the classical K-Means algorithm. The proposed technique is a centroid-based approach that expands upon the k-means algorithm by introducing a novel objective function. This function is designed to minimize multiple assignments while adhering to specific constraints. Furthermore, the method emphasizes space coverage instead of space partitioning, a deviation from the traditional K-means approach.

In order to develop the algorithm, the descriptions presented in [1] are considered as the base of its development. Therefore, its correspondent pseudocode for clustering, with the correct order of execution, can be described as follows:

Illustration 3.1 - OKM pseudocode from [1].



```
OKM($\mathcal{X}$,$t_{max}$,$\epsilon$)
Input: $\mathcal{X}$: a set of data vectors in $\mathbb{R}^p$, $t_{max}$: optional maximum number of iterations, $\epsilon$: optional threshold on the objective
output: $\{\pi_c\}_{c=1}^k$: final coverage of the points
1. Draw randomly $k$ initial cluster prototypes $\{m_c^{(0)}\}_{c=1}^k$ in $\mathbb{R}^p$ or $\mathcal{X}$.
2. For each $x_i \in \mathcal{X}$ compute the assignments $A_i^{(0)} = \textsc{Assign}(x_i, \{m_c^{(0)}\}_{c=1}^k)$
   and derive the initial coverage $\{\pi_c^{(0)}\}_{c=1}^k$ such that
       $\pi_c^{(0)} = \{x_i | m_c^{(0)} \in A_i^{(0)}\}$
3. Set $t = 0$.
4. For each cluster $\pi_c^{(t)}$ successively, compute the new prototype
       $m_c^{(t+1)} = \textsc{Prototype}(\pi_c^{(t)})$
5. For each $x_i \in \mathcal{X}$ compute the assignments $A_i^{(t+1)} = \textsc{Assign}(x_i, \{m_c^{(t+1)}\}_{c=1}^k, A_i^{(t)})$,
   and derive the new coverage $\{\pi_c^{(t+1)}\}_{c=1}^k$.
6. If not converged or $t_{max} > t$ or $\mathcal{J}(\{\pi_c^{(t)}\}) - \mathcal{J}(\{\pi_c^{(t+1)}\}) > \epsilon$, set $t = t+1$ and go to Step 4; Otherwise, stop and output final clusters $\{\pi_c^{(t+1)}\}_{c=1}^k$.
```

```
ASSIGN($x_i$,$\{m_c\}_{c=1}^k$,$A_i^{old}$)
Input: $x_i$: data vector in $\mathbb{R}^p$, $\{m_1, \ldots, m_k\}$: set of $k$ cluster prototypes and $A_i^{old}$: optional multi-assignment
output: $A_i \subset \{m_1, \ldots, m_k\}$ a subset of cluster prototypes defining a multi-assignment for $x_i$
1. Set $A_i = \{m^*\}$ such that
       $m^* = \underset{\{m_c\}_{c=1}^k}{\operatorname{argmin}} \|x_i - m_c\|^2$
   and compute $\phi(x_i)$ with assignment $A_i$.
2. Find the following nearest prototype
       $m' = \underset{\{m_c\}_{c=1}^k \setminus A_i}{\operatorname{argmin}} \|x_i - m_c\|^2$
   and compute $\phi'(x_i)$ with assignment $A_i \cup \{m'\}$.
3. If $\|x_i - \phi'(x_i)\| < \|x_i - \phi(x_i)\|$ set $A_i \leftarrow \{m'\}$, set $\phi(x_i) = \phi'(x_i)$ and go to step 2;
   Otherwise, compute $\phi^{old}(x_i)$ with $A_i^{old}$
       if $\|x_i - \phi(x_i)\| \leq \|x_i - \phi^{old}(x_i)\|$ output $A_i$,
       else output $A_i^{old}$
```

In conclusion, for more comprehensive and detailed algorithmic explanations regarding the development of the code, it is recommended to see [1] to obtain the original description from the author. Rephrasing the description may potentially result in ambiguous interpretations that deviate from the intended meaning.

4. ANALYSIS

a. Evaluation Metrics

Firstly, previously to the analysis of the results, it is necessary to clarify the evaluation metric employed according to [2]. To assess the clustering outcomes, precision, recall, and F-measure have been computed with regard to pairs of points. These measures endeavour to determine the accuracy of the prediction that a given pair of points belong to the same cluster, based on the underlying true categories in the data, for all pairs of points that share at least one cluster in the overlapping clustering results. Precision is determined as the ratio of pairs that have been correctly classified as belonging to the same cluster, while recall is the proportion of actual pairs that have been identified. The F-measure, which is the harmonic mean of precision and recall, has also been calculated.

Regarding the previous description, the mathematical formulas provided by [2] are the following:

Illustration 4.a.1 - Mathematical descriptions of Precision, Recall and F-Measure from [2].

$$\text{Precision} = \frac{\text{Number of Correctly Identified Linked Pairs}}{\text{Number of Identified Linked Pairs}}$$

$$\text{Recall} = \frac{\text{Number of Correctly Identified Linked Pairs}}{\text{Number of True Linked Pairs}}$$

$$\text{F-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Ultimately, the results obtained are documented in the Results.txt file inside the Data folder.

b. Hyper-Parameters

Considering Illustration 3.1 it is understandable that the hyper-parameters of the OKM's implementation are the following:

- **Number of Iterations (tmax)**: pertains to the maximum allowable number of iterations during the clustering process. This parameter plays a crucial role in determining the convergence of the clustering algorithm. It is important to set an appropriate value for tmax that is sufficient to ensure convergence without incurring excessive computational costs.
- **Threshold (ε)**: this refers to the threshold that has been taken into account with respect to the discrepancy between the objective function's value from the previous iteration and that from the current iteration during the optimization problem.
- **Number of Clusters (k)**: denotes the number of clusters that have been considered for the algorithm to group the instances of the datasets. The selection of k is based on prior knowledge of the data.

Table 4.b.1 indicates the values considered for the hyper-parameters mentioned previously.

Table 4.b.1 – Hyper-Parameters.

	Hyper-Parameters		
	tmax	ε	k
Reuters-21578	100	1e-6	10
Yeast	100	1e-6	14

It should be noted that these parameters have been set as fixed values for both datasets in all runs. This has been done to ensure consistency in the experimental setup and to facilitate the comparison of results across different runs and datasets.

In relation to the KM algorithm, the default values from the scikit-learn library were utilized. As for FKM, the default values from the skfuzzy library were employed. In order to address the validity of the fuzzy matrices, specific thresholds were used. For the Reuters-1, Reuters-2, Reuters-3 and Genetic datasets, the corresponding thresholds are 10.03%, 15.00%, 10.03% and 7.14%, respectively.

c. Results

Through the execution of the code provided on the datasets selected, the F-Measures were obtained. In that perspective, the analysis of the outcomes regarding each dataset will be conducted individually in (i.), Reuters-21578, and (ii.), Yeast.

It is important to note that the reported results are based on the average of 50 runs for both datasets. Furthermore, the performance metrics presented in the following tables are consistent with the analysis conducted in [1]. It is worth mentioning that the initialization of the prototypes for all the algorithms involves selecting a set of k random points from the dataset space, which remains the same for all algorithms in each run. This has been done to ensure that the comparison of the results is fair and unbiased, as required in [1].

In conclusion, it is important to acknowledge that the results obtained for the Reuters-21578 dataset may not be directly comparable to the findings reported in [1]. This is due to several factors, such as the lack of clarifications in the original study, as well as the differences in the set of documents that are selected during the preprocessing stage. Nonetheless, the primary objective of this study is to evaluate the relative performance of the distinct clustering algorithms and to test the conclusions reached in [1]. By doing so, it is aimed to contribute to a better understanding of the clustering methods and their suitability for information retrieval and biology clustering tasks.

i. Reuters-21578

The F-Measures achieved for KM, FKM and OKM algorithms related to the Reuters-21578 dataset are presented in Table 4.c.i.1.

Table 4.c.i.1 – F-Measure in the Reuters-21578 dataset.

	F-Measure [%]		
	Reuters-1	**Reuters-2**	**Reuters-3**
KM	29.46	25.75	32.78
FKM	41.20	37.26	29.70
OKM	**43.40**	**53.00**	**39.55**

Regarding the results presented in Table 4.c.i.1, it can be concluded that they are consistent with the findings reported in [1]. In particular, for all three subsets of data, OKM exhibited superior performance compared to the other algorithms in terms of classification F-Measure, which highlights its potential in the field of information retrieval.

Furthermore, it is worth noting that OKM appeared to be more stable across runs than the other algorithms, as evidenced by the lower variation of the F-Measure. This implies that the clusters obtained by OKM are generally more reliable, which is an important characteristic in practical applications.

ii. Yeast

In the Yeast dataset the F-Measure, Precision and Recall values obtained are in Table 4.c.ii.1.

Table 4.c.ii.1 – F-Measure, Precision and Recall in the Yeast dataset.

	Precision [%]	Recall [%]	F-Measure [%]
KM	**81.30**	7.78	14.21
FKM	78.18	23.90	36.26
OKM	78.40	**68.73**	**73.19**

Based on the results obtained in this study, it can be concluded that KM is the most precise clustering model for the given dataset, achieving a value of 81.30%. However, it is noteworthy that OKM outperforms KM and FKM in terms of recall (68.73%) and F-measure (73.19%), by a substantial margin.

Therefore, the results obtained in this study suggest that the OKM algorithm is a highly suitable approach for biological applications, where the presence of overlapping clusters is common. This is supported by the superior performance of OKM in terms of recall and F-measure, which indicates its ability to effectively capture the overlapping nature of the clusters in the dataset. Furthermore, these findings and the numerical values presented in Table 4.c.ii.1 provide further support for the results reported in [1].

5. CONSIDERATIONS

In replicating any scientific study, there are inherent difficulties in interpreting and implementing the methods described by the original authors. In the case of [1], although the authors note that they have taken space constraints into account, some aspects of the methodology require more detailed clarification in order to facilitate successful replication. The lack of granularity in these areas can pose significant challenges for researchers attempting to reproduce the study's results.

It is crucial to note that the absence of the mathematical demonstration of the solution used to calculate the prototypes in [1] presents a significant challenge in replicating the algorithm developed. As a new algorithm, the demonstration of this solution is fundamental in establishing its veracity, and the lack thereof undermines the reliability of the algorithm. It is noteworthy that the algorithm heavily relies on this characteristic, further emphasizing the importance of its inclusion.

Moreover, the instructions for preprocessing the Reuters-21578 dataset are overly general and lack precision regarding the values used. Specifically, a critical error concerns the reparametrization performed with LSA. The paper does not mention the value of the parameter that controls the complexity of the dataset used in the clustering task. This parameter's value has a significant impact on the results, making it crucial to include this information.

Furthermore, the paper's algorithms were theoretically optimized, but the set of parameters allowing for this optimization is not mentioned. Additionally, the soft assignment step is not described in the report.

Finally, the genetic dataset source used and referenced in the paper is no longer accessible. Therefore, exact replication of the results is not entirely feasible, although this is a minor issue that lies beyond the scope of the author.

6. CODE

a. Organization

The code developed is organized into 4 main classes:

- **Reuters.py**: contains the Reuters class, which consists of the preprocessing of the Reuters-21578 dataset characterised in Chapters 2.a and 2.b.
- **Genetic.py**: contains the Genetic class, which consists of the preprocessing of the Yeast dataset characterised in Chapters 2.c and 2.d.
- **Models.py**: contain the implementation of the MODELS class with the KM, FKM and OKM algorithms.
- **main.py**: the main .py file, where the preprocessing and clustering stages are executed.

The description of each function can be accessed through the .py files, as well as the detailed description of each step of the code.

b. Modules

The necessary modules for the code development and the respective versions are the following: **beautifulsoup4** - version 4.12.2, **numpy** - version 1.24.2, **pandas** - version 1.5.3**, scikit_learn** - version 1.2.2, **scipy** - version 1.10.1, **tabulate** - version 0.9.0 and **scikit-fuzzy** - version 0.4.2. The Python version used is 3.8.8 through PyCharm CE software.

c. Execution

To execute the code through the terminal the following steps should be taken:

1. pip install beautifulsoup4
2. pip install numpy
3. pip install pandas
4. pip install scikit_learn
5. pip install scipy
6. pip install tabulate
7. pip install scikit-fuzzy
8. python3 /PATH_WHERE_THE_FILE_main.py_IS_INSERTED

   Ex: /Users/joaovalerio/Documents/"MAI UPC"/"2 Semester"/URL/W1/source/main.py

9. /PATH_WHERE_THE_FOLDER_DATA_IS_INSERTED

   Ex: /Users/joaovalerio/Documents/MAI UPC/2 Semester/URL/W1

From the execution of the code, the output is printed in the terminal. However, for management purposes, the same output can be found on Results.txt file inside the Data folder, as stated previously.

7. CONCLUSION

All the initially proposed goals have been successfully achieved, as will be elaborated upon in the conclusion.

To begin with, the OKM algorithm exhibited superior performance in terms of the F-Measure metric. Overall, OKM demonstrated its suitability in clustering tasks, especially when dealing with datasets where multiple instances can belong to more than one cluster, as observed in information retrieval and biology.

Additionally, OKM demonstrated greater stability compared to KM or FKM, suggesting that its results are more reliable and credible in real-time applications.

Nonetheless, the paper lacked detailed descriptions of certain aspects, hindering the reproducibility and development of this work.

In conclusion, the experiments conducted on real datasets exhibited a consistent behaviour of the OKM algorithm, demonstrating its capacity to generate overlapping clusters superior to that of other specialized or extended algorithms.

8. REFERENCES

[1] CLEUZIOU, G. (2008). *An extended version of the k-means method for overlapping clustering*. France: LIFO.

[2] BANERJEE, A.; *et al.* (2005). *Model-based overlapping clustering*. USA: ACM Press.