BLG 527E - Machine Learning Term Project
2020 - Autumn

Re-implementation of
"Learning Pairwise Similarity for Data Clustering"[1]

Kübra Duran
504181521
Dept. of Computer Eng.
Istanbul Technical University
durank18@itu.edu.tr

Marcin Damek
922010001 (Erasmus student)
Dept. of Computer Eng.
Istanbul Technical University
damek20@itu.edu.tr

I. Introduction

The model ensemble chapter in [2] starts with a famous proverb "*Two heads are better than one*". It emphasizes that two minds can achieve better results. From the machine learning perspective, if we think of heads are features, then the proverb will be true. On the other hand, most succesive results are achieved not only combining the features but also the combination of models, known as model ensembles. Although the ensemble models require model and algorithm complexity, they are the most powerful techniques in machine learning applications [2].

In the case of clustering, combination of well-known clustering algorithms in other words clustering ensemble models try to find most sucessive and robust solutions to the clustering task [3], [4], [5], [6]. In these studies, different combination approaches are used to select the most qualified clusters, such as fixed combination rule, equal weights or voting based consistency measure etc. The main questions are answered below in order to give a summary about the examined study, [1]:

- **What this study says to us ?**
  In clustering task, traditional methods can show us the underlying structure of a given dataset. If we want to learn more about the given dataset we need to examine pairwise similarity between the objects. In this way, we can get more qualified clusters related to the dataset. Hence, in this study the authors propose a cluster ensemble approach in order to learn the pairwise similarity between the objects. By using the stability measure, they decide the qualified clusters and form a co-association matrix (C_M). By using the information in C_M, they calculate the average stability of each cluster in the final partition. As they use multiple algorithms in the application, they named this method as multi- EAC. In the ensemble method, they use k-means, SL (single link) and SC (spectral clustering) algorithms with different k and σ values They test Multi-EAC and EAC [3] model by using different datasets. As they use multiple algorithms by changing the parameters, they achieve better results with multi-EAC.
- **Why the topic is considered ?**
  Clustering algorithms in machine learning applications result in different partitions. As they use non-unique approaches and take into consideration different similarity measures we face large number of choices for the algorithm selection. Therefore, there arise several questions regarding the clustering task, such as "*Which similarity measure to use given a clustering problem ?*", and "*Are there any similar pattern on the underlying structure of a given dataset ?*" Thereby, we need additional methodologies in order to identify the data in a closer inspection.
- **How this study solves these problems ?**
  In order to get the underlying structure of a given dataset, the authors propose a clustering ensemble approach combined with cluster stability conditions to learn the pairwise similarity. They use multi-algorithm approach and take the performance of them for different regions on the dataset. After that, they combine the most qualified clusters taken from the results of different algorithms. In this way, only the clusters with the stability passing the threshold value contribute to the final partitioning.

In order to understand the work [1], we also revised the other studies [7],[8] of the authors related with this robust clustering ensemble method. In section II, we give the devel-

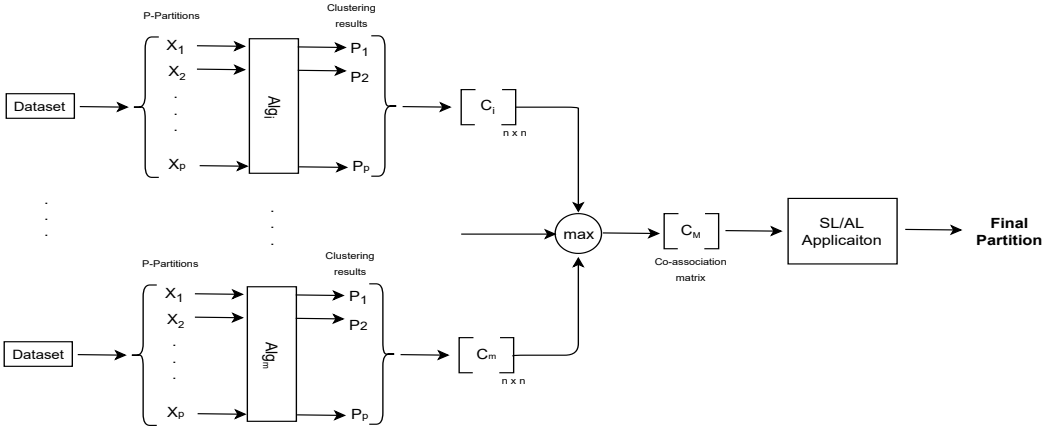


Fig. 1: Development steps of multi-EAC

opment steps of multi-EAC method. In Section III, we give the experimental results and discussion. Finally, in Section IV, we conclude this project.

II. Development Steps

1. Preparing Datasets

We took the data sets Iris, Breast-C, opdigits, and Yeast by using scikit learn library. For the Yeast dataset, we create logarithm and the standard versions.

2. Multi-EAC Application

In order to realize multi criteria clustering to learn pairwise similarity of objects, we start with creating q partitions from a given dataset (X) as given in (1). After that, we applied the clustering algorithm k-means, SL or SC, lets say Alg_i to the each of the partitions. As a result, we get q x (k - # of setted clusters) clusters.

$$X = \bigcup_{i=1}^{q} X_i,\ X = X_1 \cup\ X_2 \cup \dots X_q \tag{1}$$

For each algorithm, we calculate an nxn C matrix using (2), named as co-association matrix, where n is the number of objects in the dataset. $n_{i,j}$ in the nominator stands for the number of times the pair of objects with indices i and j are grouped together in a cluster. And $m_{i,j}$ represents the number of data sets that these objects are present at the same time.

$$C_{i,j} = \frac{n_{i,j}}{m_{i,j}}\ ,\ \forall_{i,j} \tag{2}$$

By using the C matrix, we calculate the $stab$ values for each cluster by using (3). After that, we use these $stab$ values in teh selection of the most significant clusters. We just check if the $stab$ value for a cluster is higher than the threshold value or not. Then we construct a new C_i matrix for the Alg_i, that includes the most significant clusters in the partition.

$$stab_{C_k} = \sum_{i,j \in C_k, j \neq i} \frac{C(i,j)}{(n_k)(n_k - 1)} \tag{3}$$

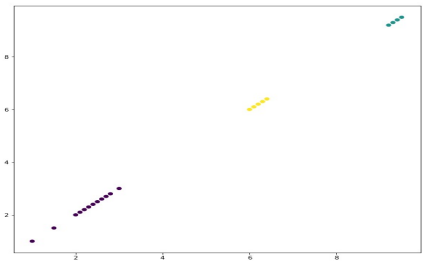
After getting C_i matrices for all algorithms, we use max rule in order to get the C_M co-association matrix. We run AL and SL algorithms on C_M in order to decide how many clusters should be placed in the final partition. In the deciding process, we calculate lifetime of the clusterings and choose the maximum one as the final number of clusters. We visualize the AL and SL applicaiton results in dengrograms. Our development steps can be seen in Fig. 1.

III. Experimental Results and Discussion

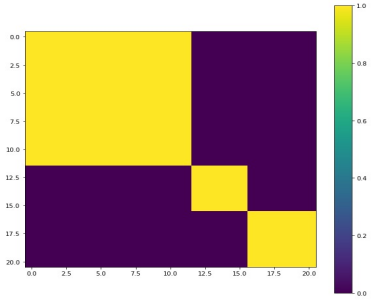
In our experiments for all data sets, we use 10 subsamples because of the computation time. We firstly examine *Synthetic* dataset. As it is seen from the scatter plot of this data which is given in Fig.2a, there are 3 clusters in this dataset. Moreover, calculated C_M matrix for this data is given in Fig.2b. The application of AL and SL methods in order to decide the final number of clusters are given in Fig.2c and Fig.2d respectively. In the dendrograms, we can see the constructed cluster sizes in the x axis and the calculated distances $d(i,j)$, with $d(i,j) = 1 - C(i,j)$ as calculated in [3]. As seen from the dendrogram given in Fig.1.c, the lifetime for 2-cluster case is l_2=4.07 - 3.0 = 1.07, and the lifetime for 3-cluster case is l_3=3.0. As l_3 is higher than l_2, we decide that 3 cluster will be in the final partition. Calculations for SL are done in the same way.

In order to test the developed model, we try K-means, SL and SC algorithms both one by one and the combination of them. As the comparison metric, we use C_i index value that is $C_i : C_i(P^*, P^o)$ by comparing the original label values that comes from the ground truth information and the calculated label values form the final partition. As seen from Table-I, we achieve 100.0 C_i value in the multi-EAC cases and 0.95 threshold for the synthetic dataset. It seems that we achieve better results for synthetic data compared to the original work. The reason for that may stems form the difference in dataset. In the original work, the authors use more complex synthetic dataset rather than ours.

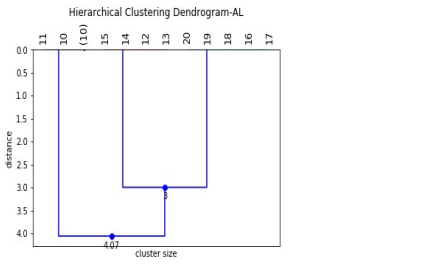
We continue to test our model with *Iris* dataset. The scatter plot of this dataset, calculated C_M, AL and SL application dendrograms are given in Fig.3a, Fig.3b, Fig.3c and Fig.3d,



(a) Synthetic dataset

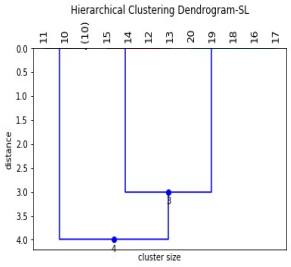


(b) C_M matrix for synthetic dataset



# of clusters for final partitioning in Pa according to lifetime criteria: 3

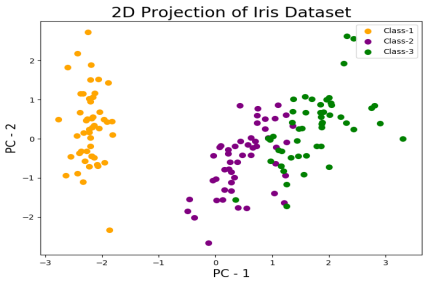
(c) AL application result to synthetic dataset



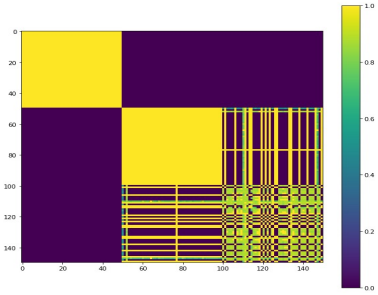
# of clusters for final partitioning in Pa according to lifetime criteria: 3

(d) SL application result to synthetic dataset

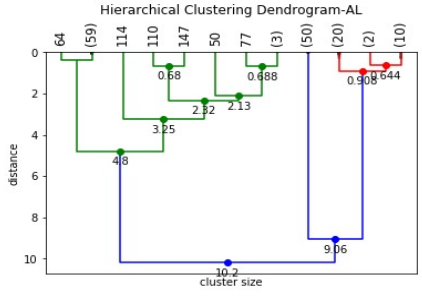
Fig. 2: Results for synthetic dataset



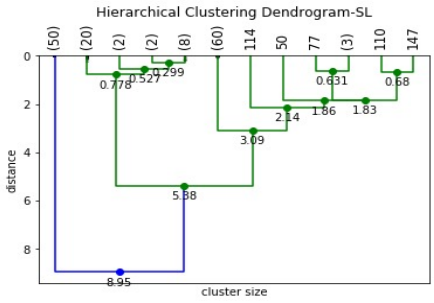
(a) Iris dataset



(b) C_M matrix for Iris dataset

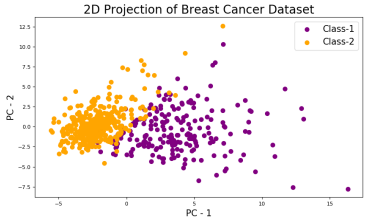


(c) AL application result to Iris dataset

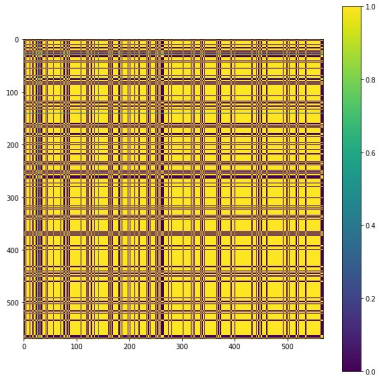


(d) SL application result to Iris dataset

Fig. 3: Results for Iris dataset

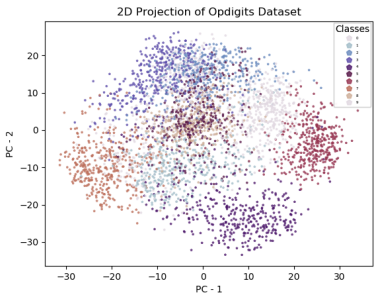


(a) breast cancer dataset

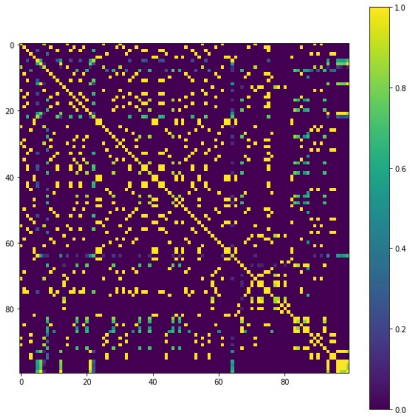


(b) C_M matrix for breast cancer dataset

Fig. 4: Results for breast cancer dataset



(a) opdigits dataset



(b) C_M matrix for opdigits dataset

Fig. 5: Results for opdigits dataset

TABLE I: Combination results

	Clustering Ensembles: K-means, SL		Clustering Ensembles: K-means, SL, SC	
	Multi-EAC		Multi-EAC	
	Ci (lifetime)	th	Ci (lifetime)	th
Synthetic	100.0	0.95	100.0	0.95
Iris	90.0	0.95	88.0	0.95
Breast-C.	85.41	0.95		
opdigits	81.0	0.75	81.0	0.80
log_yeast	45.33	0.90	46.6	0.90
std_yeast	46.6	0.90	46.6	0.90

respectively. We choose the total number of clusters in the final partition making lifetime calculations as in synthetic dataset. Furthermore, results for *Breast cancer*, *opdigits*, *logyeast*, and *stdyeast* datasets are given in Fig.4, Fig.5, Fig.6 and Fig.7 respectively. For these data sets, we did not use lifetime criteria, we manually set the number of clusters in the final partition.

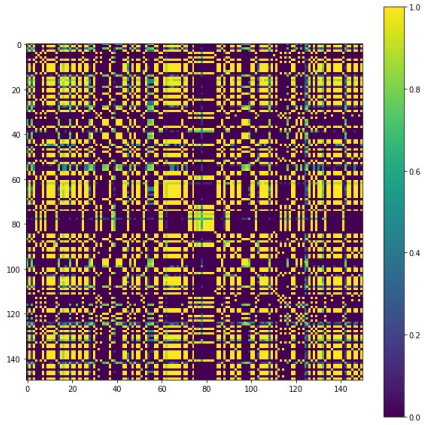
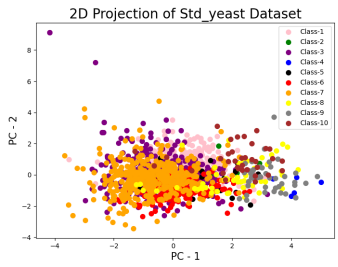


Fig. 6: C_M matrix for logyeast dataset

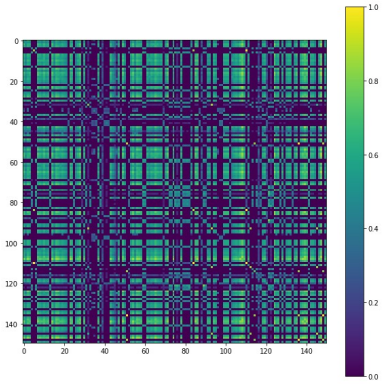
We run our method on 5 different data sets. And take into consider two models, first one is K-means and SL combined model and the latter one is K-means, SL and also SC combined model. Also, we applied the k and σ parameters as defined in paper [1] for each of the dataset. Our results can be seen in Table-I.

For the first ensemble model (K-means and SL combination), for *synthetic* data we can say that we achieve better results in multi-EAC case with th= 0.95 and C_i= 100 compared to the original work. This difference may stems form the difference in the dataset. They use more complex synthetic data compared to us. For all the other datasets except *breast cancer*, we also achieved better results than the original work.

For the second ensemble model (K-means, SL and SC combination), we could not get better results except *logyeast* dataset. Actually when we add SC to our method, we took warnings for big datasets. Furhermore, for *breast cancer* dataset we took an error saying that the graph is not complete. As it is a graph partitioning algorithm, maybe it requires additional works for the specific data sets. However our results are similar to the original work.



(a) stdyeast dataset



(b) C_M matrix for stdyeast dataset

Fig. 7: Results for stdyeast dataset

As we do not get successive results with SC, we tried to use supervised learning algorithm k-nn (k-nearest neighbors), because others are unsupervised. In our applicaiton 70% of data went for learning and 30 % was divided to subsamples (with respect 90% of the remaining 30% data). We used threshold 0.99. k-nn was learned before all subset. At the end, we achieved 90% accuracy. The C_M matrix calculated with this new ensemble is given in Fig.8. Unfortunately, using the supervised learning algorithm, we did not manage to significantly improve the performance of the ensemble model. We can focus on this algorithm in the future.

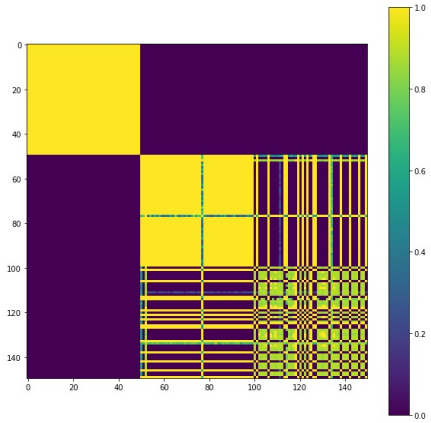


Fig. 8: C_M matrix for iris dataset by adding k-nn algorithm

IV. Conclusion

In this project, we try to re-implement the paper "*Learning Pairwise Similarity for Data Clustering*"[1]. Within this work, we use multi-EAC ensemble model for data clustering task. In order to select the significant clusters, we use stability criteria that takes into consider the co-occurance of the objects. Hence, we achieve learning the pairwise similarity based on the stable clusters. As the algorithms, we use k-means, SL and SC in combination and apply max rule in order to get the most qualified clusters from a given dataset. Moreover, we use AL and SL to compute the lifetime of the k-class cases, and using dendrograms we decide how many clusters should we have in the final partition. As the extraction method, we use AL as in the original work. Our results show that, we achieve better results in multi-EAC case except one dataset.

This was the first time for both of us getting to know and apply ensemble models. We have learned a lot within the concept of this study. In the future, we plan to use the knowledge that we gain while developing our thesis.

References

[1] A. L. Fred and A. K. Jain, "Learning pairwise similarity for data clustering," in *18th International Conference on Pattern Recognition (ICPR'06)*, vol. 1. IEEE, 2006, pp. 925–928.

[2] P. Flach, *Machine learning: the art and science of algorithms that make sense of data*. Cambridge University Press, 2012.

[3] A. L. Fred and A. K. Jain, "Combining multiple clusterings using evidence accumulation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, no. 6, pp. 835–850, 2005.

[4] H. G. Ayad and M. S. Kamel, "Cluster-based cumulative ensembles," in *International Workshop on Multiple Classifier Systems*. Springer, 2005, pp. 236–245.

[5] L. I. Kuncheva and S. T. Hadjitodorov, "Using diversity in cluster ensembles," in *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583)*, vol. 2. IEEE, 2004, pp. 1214–1219.

[6] A. Strehl and J. Ghosh, "Cluster ensembles—a knowledge reuse framework for combining multiple partitions," *Journal of machine learning research*, vol. 3, no. Dec, pp. 583–617, 2002.

[7] A. Lourenço and A. L. Fred, "Selectively learning clusters in multi-eac." in *KDIR*, 2010, pp. 491–499.

[8] L. Ana and A. K. Jain, "Robust data clustering," in *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, vol. 2. IEEE, 2003, pp. II–II.