K-Means-Based Consensus Clustering

Josep Famadas
jfamadas95@gmail.com

April 26th, 2018

Contents

1 Introduction

In this document it is presented the implementation of the consensus clustering technique proposed in [1].
First, the algorithm is explained and then some experiments are performed in order to test its limitations and its performances compared to other clustering algorithms.

2 Algorithm

Consensus clustering consists of finding a single partitioning of data from multiple existing basic partitionings. This is done by finding the partition π that maximizes Eq. (1), being π_i all the basic partitions contained in Π, ω_i the weight associated to each basic partition, r the number of basic partitions and U a utility function.

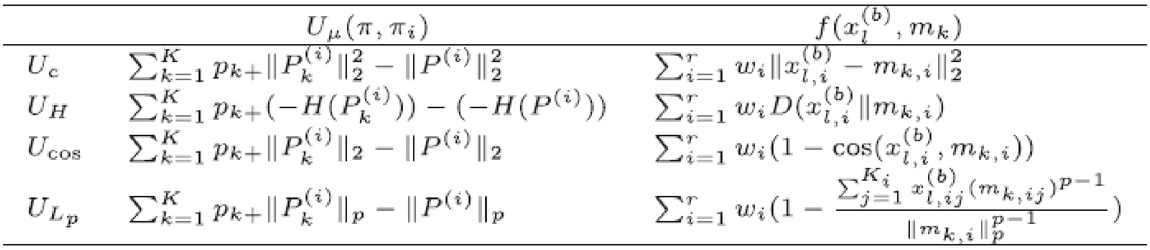
$$\Gamma(\pi, \Pi) = \sum_{i=1}^{r} \omega_i U(\pi, \pi_i) \tag{1}$$

In the paper, by mathematical demonstrations, some utility functions are derived, which allow the consensus clustering problem to be transformed into a K-Means problem and then solved by the 2-phase algorithm used in K-Means.
The K-Means problem is now solved following this procedure:

1. A binary matrix $\mathcal{X}^{(b)}$ is constructed with the partitions of Π by concatenating the labels of each π_i in one-hot encoding. Therefore, the shape of $\mathcal{X}^{(b)}$ is (n, $\sum_{i=1}^{r} K_i$), being n the number of data instances and K_i the number of clusters in each basic partition i. Each row of the binary matrix contains r ones.

2. The binary matrix is clustered with K-Means but using as sample-centroid distance instead of the classical euclidean distance, the $f(x_l^{(b)}, m_k)$ (being $x_l^{(b)}$ the sample and m_k the centroid) associated with each utility function (see Figure 1).

3. The original dataset i labeled with the partition of $\mathcal{X}^{(b)}$ obtained.



	$U_\mu(\pi, \pi_i)$	$f(x_l^{(b)}, m_k)$
U_c	$\sum_{k=1}^{K} p_{k+} \Vert P_k^{(i)} \Vert_2^2 - \Vert P^{(i)} \Vert_2^2$	$\sum_{i=1}^{r} w_i \Vert x_{l,i}^{(b)} - m_{k,i} \Vert_2^2$
U_H	$\sum_{k=1}^{K} p_{k+}(-H(P_k^{(i)})) - (-H(P^{(i)}))$	$\sum_{i=1}^{r} w_i D(x_{l,i}^{(b)} \Vert m_{k,i})$
$U_{\cos}$	$\sum_{k=1}^{K} p_{k+} \Vert P_k^{(i)} \Vert_2 - \Vert P^{(i)} \Vert_2$	$\sum_{i=1}^{r} w_i (1 - \cos(x_{l,i}^{(b)}, m_{k,i}))$
U_{L_p}	$\sum_{k=1}^{K} p_{k+} \Vert P_k^{(i)} \Vert_p - \Vert P^{(i)} \Vert_p$	$\sum_{i=1}^{r} w_i (1 - \frac{\sum_{j=1}^{K_i} x_{l,ij}^{(b)} (m_{k,ij})^{p-1}}{\Vert m_{k,i} \Vert_p^{p-1}})$

Figure 1: Utility functions used in the implementation and their respective distance function. D–KL-Divergence; H–Shannon entropy; L$_p$ – $L_p norm$.

3 Experimental Results

In this section are presented the experimental results of the K-Means-based Consensus Clustering applied to real world datasets described in Table 1.

The experimental results have been performed with the following setup;

- *Validation measure* : Given the fact that there is the ground truth for every dataset the selected validation measure is the adjusted Rand index. It is used from *sklearn*.
- *Basic Partitions* : 100 basic partitions are generated using *sklearn* K-Means being the number of clusters a random integer from K (the actual number of classes) to $\sqrt{n}$ (being n the number of objects) for each partition.
- *KCC run* : The input number of clusters is the actual number of classes and the algorithm is run 100 times (because it is non deterministic) taking the best partition out of the 10 (evaluated with the $calinski - harabaz$ score).

Data Set	#Objects	#Attributes	#Classes
iris	150	4	3
wine	178	13	3
breastcancer	569	30	2

Table 1: Datasets used in the experimental results.

3.1 Clustering Quality

In this section, the KCC is performed for each dataset described in Table 1 using each of the utility functions described in Figure 1. As said in this section, the metric quality of each partition is measured using the adjusted Rand index. The results can be seen in Table 2

Dataset	U_c	U_H	U_{cos}	U_{L5}	U_{L8}
iris	0.7455	0.7455	0.7455	0.7455	0.7455
wine	0.3711	0.3711	0.3711	0.3749	0.3711
breastcancer	0.5296	0.5019	0.5557	0.5502	0.5557
score	1.6462	1.6185	1.6723	1.6706	1.6668

Table 2: Rand score using different distance functions.

As it can be seen in the table, the highest Rand Score is achieved with the *cosine* distance. However, the difference is not enough to say that it is the best distance to use.

3.2 Methods comparison

In this section, it is selected the KCC distance that gave better results (U_{cos}) and compared against different clustering and consensus clustering methods. In Table 3 (ordered):

- K-Means (sklearn)
- Gaussian Mixture Model (sklearn)
- Hierarchical Clustering - Single Link (scikit)
- Hierarchical Clustering - Complete Link (scikit)
- Simple Consensus Clustering (kemlglearn)

Dataset	KCC	$K-Means$	GMM	$HC-SL$	$HC-CL$	SCC
$iris$	0.7455	0.7302	0.7302	0.5637	0.6423	0.7455
$wine$	0.3711	0.3711	0.3941	0.0054	0.3708	0.4029
$breastcancer$	0.5557	0.4914	0.4169	0.0024	0.0523	0.4149
$score$	1.6723	1.5927	1.5412	0.5715	1.0654	1.5633

Table 3: Different consensus clustering and clustering algorithms compared to the KCC.

As it can be appreciated, the KCC is the method that obtains better results followed nearly by K-Means.

References

[1] J. Wu, H. Liu, H. Xiong, J. Cao, and J. Chen. K-means-based consensus clustering: A unified view. 2015.