Sparse Subspace K-means

1st Abdoul Wahab Diallo
African Institute for Mathematical Sciences (AIMS)
Mbour, Sénégal
abdoul.w.m.diallo@aims-senegal.org

2nd Ndèye Niang
CEDRIC LAB
CNAM
Paris, France
ndeye.niang_keita@cnam.fr

3rd Mory Ouattara
UFR SFA
Université NANGUI ABROGOUA
Abidjan, Côte d'Ivoire
ouattaramory.sfa@univ-na.ci

***Abstract*—We consider the problem of clustering observations described by a large set of features. The full set of features may not be relevant to determine the true underlying clusters present in the data which may differ only with respect to a small number of features. We propose Sparse subspace K-means (SSKM) a new subspace clustering method that performs simultaneously a clustering of the observations and the selection of relevant features for each cluster. The method is based on a single criterion with a lasso-type penalty for both the relevant features selection and the clustering as in the sparse K-means. The proposed method associates a set of relevant features to each cluster rather than to the whole partition as it is done in sparse K-means. The method is demonstrated on simulated and real data. In comparison with K-means, Sparse K-means and Entropy Weighting K-Means (EWKM) the SSKM method has better performances both in terms of partition quality indices and detection of relevant features.**

***Index Terms*—K-means, sparse, subspace clustering, high dimensional data**

I. Introduction

Clustering is a process of partitioning a set of objects into clusters such that objects in the same cluster are more similar to each other than objects in different clusters according to some defined criteria [1], [2].
In high dimensional data, clusters are often described by subspaces of features. Thus, not all of the features are relevant for cluster determination. The irrelevant features can confuse clustering algorithms by hiding clusters in noisy data [3]; so it becomes difficult for these algorithms to return a good partition. To overcome this inefficiency of classical clustering methods, subspace clustering algorithms have been proposed. The objective of subspace clustering methods is to find clusters from subspaces of data, instead of the entire data space [3], [4].

There are two main types of subspace clustering. The fisrt one is Hard Subspace Clustering (HSC) with methods such as CLustering In QUEst (CLIQUE) [5] and ENtropy-based CLUStering (ENCLUS) [6], Fast and INtelligent subspace clustering algorithm using DImension voTing (FINDIT) [7] that determine the exact subspaces where the clusters are found.
The second type is Soft Subspace Clustering (SSC) [8] with methods such as Weighting K-Means (W K-Means) [9], Entropy Weighting K-Means [10], Feature Group K-Means (FGKM) [11] and Entropy-based subspace clustering for mining numerical data [12], that are based on assignments of weights to each feature for each cluster and then discovers clusters from subspaces of the features with large weights.
When the number of features is very large, the process of assigning weights in the soft subspace clustering methods such as the EWKM [10] becomes complex ; thus leading to a difficulty in the clusters discovery and also making the interpretation of the partition complicated. The issue of high dimensional data clustering has been addressed by many authors and several methods have been proposed. Among them we are interested in sparse methods [13], [14], [15], [16], [17], [18].

We restrict ourselves to distanced-based methods and we consider the Sparse K-means method [19]. This method is inspired by sparse regression methods [20]. Using a criterion with a lasso-type penalty, Sparse K-means set to zero the weights of the features not relevant for the partition.

In this paper, we propose a new sparse subspace clustering method called SSKM (Sparse Subspace K-Means) that performs simultaneously a clustering of the observations and the selection of relevant features for each cluster that facilitates the interpretation of the partition. The method is based on a single criterion with a lasso-type penalty for both the relevant features selection and the clustering as in the sparse K-means.The proposed method sets the weights of the irrelevant features for each cluster to zero while assigning large weights to features characterizing the clusters. The results of a series of experiments conducted on both synthetic and real data show that the new method outperforms the litterature methods such as K-means, Sparse K-means and EWKM.

The rest of this paper is organized as follows. Section 2 is a brief description of two related methods for variable selection in clustering. In section 3, the new SSKM method and the different steps of the associated algorithm are presented. Section 4, presents the comparison of the SSKM method with the classic K-means method, the EWKM and Sparse K-means methods through an application on simulated and real data. Section 6 gives the conclusion.

II. Related Work

Feature selection is an important issue in high dimensional data cluster analysis addressed by several authors. Among proposed methods are EWKM and sparse K-means based on a modification of the k-means criterion.

2375-9259/21/$31.00 ©2021 IEEE
DOI 10.1109/ICDMW53433.2021.00089

A. Entropy Weighting K-Means

In [10], the authors propose EWKM, a K-means type algorithm for soft subspace clustering of high-dimensional. The method is based on the determination of a set of weights introduced in the cost function of the K-Means algorithm. The EWKM method simultaneously maximizes between-cluster inertia and maximize a negative entropy term in the learning process.

In the proposed algorithm, the authors consider that the weight of a feature in a cluster represents the probability that the feature contributes to form the cluster. Thus the entropy of the feature weights represents the certainty of features in the identification of a cluster.

Therefore, they modify the objective function of K-means by adding the weight entropy term to it so that they can simultaneously minimize the within cluster dispersion and maximize the negative weight entropy to stimulate more features to contribute to the identification of clusters.
For each feature, weights inversely proportional to their variance in each cluster are computed.

B. Sparse K-means

In [19], inspired by sparse methods in regression, the authors propose Sparse K-means a method which clusters the observations using an adaptively chosen subset of the features.

Let $x^j (j = 1, \ldots, p)$ be p features describing n objects associated to the data matrix X. x_i^j denotes the value of feature j for object i, K is the number of clusters.

The classical K-means clustering seeks to partition the n observations into K sets, or clusters, such that the within-cluster sum of squares

$$\sum_{k=1}^{K} \frac{1}{n_k} \sum_{i,i' \in C_k} \sum_{j} d(x_i^j, x_{i'}^j) \tag{1}$$

is minimal, where n_k is the number of observations in cluster k and C_k contains the indices of the observations in cluster k. In general, $d(x_i^j, x_{i'}^j)$ can denote any dissimilarity measure between observations i and i' along feature j.

Then minimizing the within-cluster sum of squares is equivalent to maximizing the between cluster sum of squares. Then the authors define the between-cluster sum of squares as with constraints on the weights. Then the criterion associated with the proposed Sparse K-means algorithm is as follows:

$$\max_{C_1,\ldots,C_K,w} \left\{ \sum_{j=1}^{p} w_j \left(\frac{1}{n} \sum_{i=1}^{n} \sum_{i'=1}^{n} d(x_i^j, x_{i'}^j) - \sum_{k=1}^{K} \frac{1}{n_k} \sum_{x_i, x_i' \in C_k} d(x_i^j, x_{i'}^j) \right) \right\} \tag{2}$$

subject to $\|w\|_2 \leq 1, \|w\|_1 \leq s, w_j \geq 0, \forall j$

The optimization of the Sparse K-means objective function [19] provides the sparse weights $w_j (j = 1, ..., p)$ for the partition.

The criterion (2) assigns a weight to each feature, based on the increase in between-cluster inertia that the feature can contribute. First, consider the criterion with the weights $w_1, ..., w_p$ fixed. It reduces to a clustering problem, using a weighted dissimilarity measure. Second, consider the criterion with the clusters $C_1, ..., C_K$ fixed. Then a weight will be assigned to each feature based on the between-cluster inertia of that feature; features with larger between-cluster inertia will be given larger weights.

The process of optimizing the features weights in (2) is done using the formula below.

$$w = \frac{S(a_+, \Delta)}{||S(a_+, \Delta)||_2}$$

where

$$a_j = \frac{1}{n} \sum_{i=1}^{n} \sum_{i'=1}^{n} d(x_i^j, x_{i'}^j) - \sum_{k=1}^{K} \frac{1}{n_k} \sum_{x_i, x_i' \in C_k} d(x_i^j, x_{i'}^j) \tag{3}$$

and $\Delta = 0$ if this results in $||w||_1 < s$, otherwise, $\Delta > 0$ is chosen so that $||w||_1 = s$. Note that a_+ denotes the positive part of a and S is a soft-thresholding operator defined by

$$S(x, c) = sign(x)(|x| - c)_+.$$

The feature relevance on sparse K-means method is relative to the whole partition and not to individual clusters unlike for subspace clustering methods in which weights are specifically associated with clusters.

We propose SSKM, a extension of the Sparse K-means method to subspace clustering. This new method determines the clusters of the partition and simultaneaously assigns zero weights to irrelevant features for each cluster avoiding the tedious weights analysis step.

III. Sparse Subspace K-means

A. Sparse Subspace K-means criterion

The SSKM method is based on the Sparse K-means criterion. We propose a modification of the objective function of (2) by considering the sparsity of the weights at the clusters of the partition. In (2), when the weight of the feature w_j is different to 0 it means that the feature is relevant for the partition. As it is for all the clusters we don't know for which cluster the feature is relevant. This is not necessarily the case in subspace clustering.

Then the choice of sparsity at cluster level is motivated by the fact that each cluster will be characterized by a specific set of features.

More precisely, the SSKM optimization criterion is obtained by modifying (4) to take into account the relevance of the feature in the determination of the cluster through the replacement of w_j by w_j^k.

This gives the following objective function :

$$\max_{c_1,\ldots,c_K,w^k} \left\{ \sum_{j=1}^{p} \sum_{k=1}^{K} w_j^k \left(\frac{1}{nK} \sum_{i=1}^{n} \sum_{i'=1}^{n} d(x_i^j, x_{i'}^j) - \frac{1}{n_k} \sum_{x_i, x_{i'} \in c_k} d(x_i^j, x_{i'}^j) \right) \right\} \quad (4)$$

subject to :

$\|w^k\|_2 \leq 1$, $\|w^k\|_1 \leq s$ and $w_j^k \geq 0, \forall j$ and k. where w^k is the weight vector of the features for cluster k. The first term of (4) is related to the total inertia and the second term to the within-cluster inertia. Thus, maximizing (4) is equivalent to minimizing the weighted within-cluster inertia.

In the same way as Sparse K-means, the process of optimizing the features weights in (4) is done using the following formula:

$$w^k = \frac{S((a^k)_+, \Delta)}{||S((a^k)_+, \Delta)||_2} \quad (5)$$

where the coordinates of vector a^k are

$$a_j^k = \frac{1}{n} \sum_{i=1}^{n} \sum_{i'=1}^{n} d(x_i^j, x_{i'}^j) - \frac{1}{n_k} \sum_{x_i, x_{i'} \in c_k} d(x_i^j, x_{i'}^j) \quad (6)$$

and $\Delta = 0$ if this results in $||w^k||_1 < s$, otherwise, $\Delta > 0$ is chosen so that $||w^k||_1 = s$.

The iterative algorithm for maximizing (4) is presented in the following.

B. Sparse Subspace K-means algorithm

1) Initialize w^k with $w_1^k = ... = w_p^k = \frac{1}{\sqrt{p}}$, $1 \leq k \leq K$.
2) Iterate until convergence :
   - Fix w^k and optimize (4) with respect to $C_1, ..., C_K$:

     $$\min_{C_1,\ldots,C_K} \left\{ \sum_{k=1}^{K} \frac{1}{n_k} \sum_{x_i, x_{i'} \in c_k} \sum_{j=1}^{p} w_j^k d(x_i^j, x_{i'}^j) \right\} \quad (7)$$

     by applying the standard K-means algorithm to the weighted data.
   - Fix $C_1, ..., C_K$ and optimize (4) with respect to $w^1, ..., w^k$ by applying formula (5).

When implementing the algorithm, step 2 is iterated until the condition below is satisfied.

$$\frac{\sum_{j=1}^{p} |(w_j^k)^r - (w_j^k)^{r-1}|}{\sum_{j=1}^{p} |(w_j^k)^{r-1}|} < 10^{-4} \quad (8)$$

Here $(w^k)^r$ indicates the weight vector obtained at iteration r.

The SSKM algorithm relies on a proper choice of the sparsity parameter s. Thus the algorithm given below allows to select the optimal s parameter.

C. Tuning parameter s selection for SSKM

1) Obtain permuted data sets $X_1, ..., X_B$ by independently permuting the observations within each feature of the dataset X.
2) For each candidate tuning parameter value s:
   - Compute

     $$O(s) = \sum_{j=1}^{p} \sum_{k=1}^{K} w_j^k \left(\frac{1}{nK} \sum_{i=1}^{n} \sum_{i'=1}^{n} d(x_i^j, x_{i'}^j) - \frac{1}{n_k} \sum_{x_i, x_{i'} \in c_k} d(x_i^j, x_{i'}^j) \right) \quad (9)$$

     the objective obtained by performing SSKM with tuning parameter value s on the data X.
   - For $b = 1, 2, .., B$, compute $O_b(s)$, the objective function obtained by performing SSKM with tuning parameter value s on the data X_b.
   - Compute

     $$Gap(s) = log(O(s)) - \frac{1}{B} \sum_{b=1}^{B} log(O_b(s)) \quad (10)$$

3) Choose $s^\star$ corresponding to the largest value of Gap(s).

IV. Experiments

A. Data

Two data sets X_1 and X_2 were simulated. The two data sets contain 200 objects described by 60 features. The objects are divided into 4 clusters of 50 objects each. The data set X_1 has been simulated in such a way that all clusters are well separated and described by the whole set of features. Hence data set X_1 does not have a subspace structure. On the other hand, for the data set X_2 each cluster is described by a subspace of the set of features. The clusters c_1, c_2, c_3, and c_4 are described by the features grouped from 1 *to* 15, 16 *to* 30, 31 *to* 45 and 46 *to* 60, respectively. Naturally for each cluster the features describing the other clusters are noisy features. Fig. 1 and Fig. 2 present the cluster structure in the first factorial plan of a principal component analysis on the data sets X_1 and X_2. We observe well separated clusters on X_1 and overlapping clusters on X_2.

Moreover, to investigate the performance of the SSKM algorithm on real data we selected two datasets from the UCI Machine Learning Repository [22]: the Multiple Features data set (Dutch utility maps, DMU) and the Image Segmentation (IS) data set.

The Multiple Features data set contains 2000 handwritten digits grouped in 10 clusters $c_k (k = 0, \ldots, 9)$, each with 200 objects. Each digit is described by 649 features that are divided into the following six feature groups:

$G1 = \{A\_i(i = 0, \ldots, 75)\}$: mfeat-fou group, contains 76 Fourier coefficients of the character shapes.
$G2 = \{B\_i(i = 0, \ldots, 215)\}$: mfeat-fac group, contains 216 profile correlations.
$G3 = \{C\_i(i = 0, \ldots, 63)\}$: mfeat-kar group, contains 64

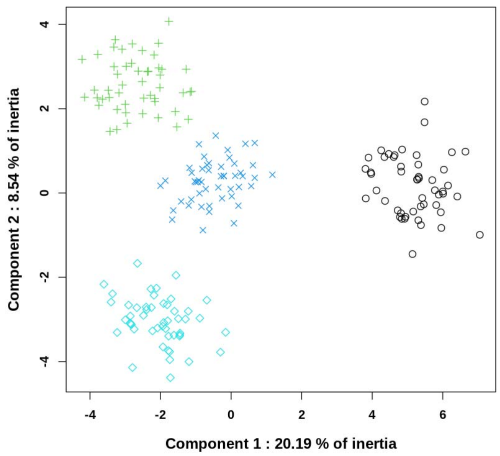

Fig. 1. Projection of X_1 clusters

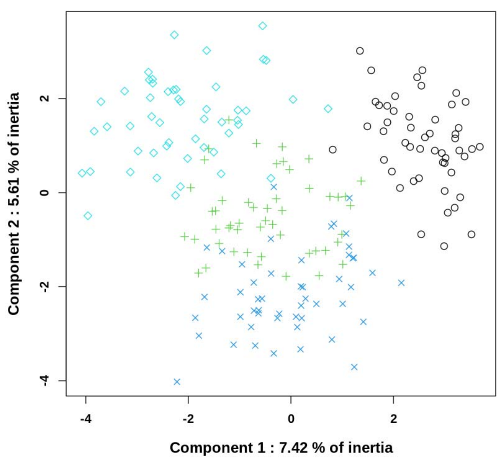

Fig. 2. Projection of X_2 clusters

Karhunen-Loeve coefficients.
$G4 = \{D\_i(i = 0, \ldots, 239)\}$: mfeat-pix group, contains 240 pixel averages in 2×3 windows.
$G5 = \{E\_i(i = 0, \ldots, 46)\}$: mfeat-zer group, contains 47 Zernike moments.
$G6 = \{F\_i(i = 0, \ldots, 5)\}$: mfeat-mor group, contains six morphological features.

The Image Segmentation data set consists of 2310 objects grouped in 7 clusters, drawn randomly from a database of out door images. The data set contains 18 features which can be naturally divided into two feature groups:

$g1 = \{V\_i(i = 1, \ldots, 8)\}$: Shape group, it contains the first eight features about the shape information of the images.

$g2 = \{V\_i(i = 9, \ldots, 18)\}$: RGB group, it contains the last 10 features about the RGB values of the images.

Fig. 3 and Fig. 4 present the cluster structure in the first factorial plan of a principal component analysis on the DMU and IS data sets.

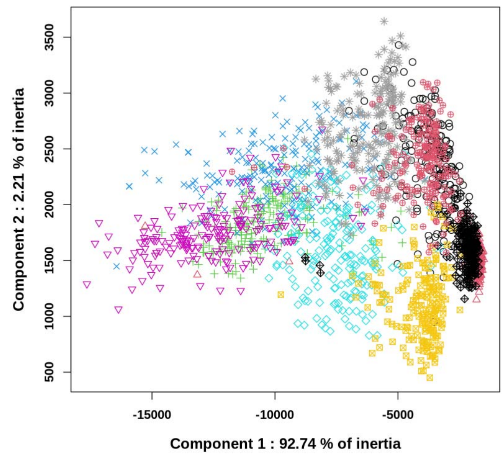

Fig. 3. Projection of DMU clusters

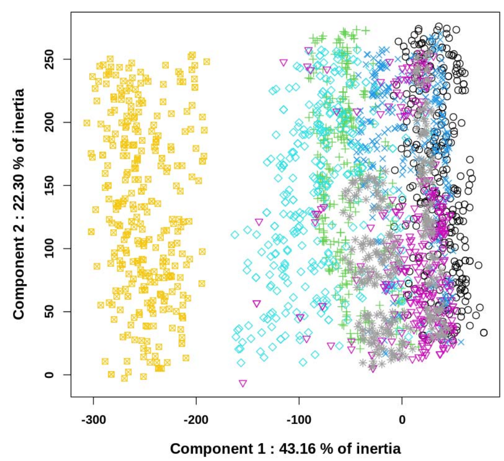

Fig. 4. Projection of IS clusters

B. Comparison of performances

Table I contains the average performances (and standard deviation) in terms of Normalized Mutual Information (NMI) and Adjusted Rand Index (ARI) for 30 repetitions of the K-means, EWKM, Sparse K-means and SSKM algorithms on the data sets X_1, X_2, DMU and IS. The first and second rows represent the NMI and ARI indices respectively.

Comparisons of the results show that the SSKM and Sparse K-means algorithms outperform the others algorithms on X_1. Indeed, as presented in section (IV-A), the clusters are well separated on the X_1 data set and without any subspace structure; this explains the very good performances of SSKM, Sparse K-means and Kmeans. For the X_2 data set, SSKM

TABLE I
PERFORMANCES OF K-MEANS, EWKM, SPARSE K-MEANS AND SSKM ON X_1, X_2, DMU AND IS DATA SETS

Data	K-means	EWKM	Sparse K-means	SSKM
X_1	0.95 (0.1)	0.52 (0.1)	**1** (0)	**1** (0)
	0.93 (0.13)	0.45 (0.21)	**1** (0)	**1** (0)
X_2	0.76 (0.08)	0.47 (0.17)	0.89 (0)	**0.92** (0.02)
	0.77 (0.12)	0.45 (0.17)	0.92 (0)	**0.92** (0.03)
DMU	0.73 (0.04)	0.49 (0.05)	0.78 (0)	**0.81** (0.02)
	0.63 (0.06)	0.38 (0.07)	0.71 (0)	**0.75** (0.05)
IS	**0.57** (0.02)	0.47 (0.08)	0.56 (0)	**0.57** (0.01)
	0.47 (0.03)	0.34 (0.09)	0.46 (0)	**0.46** (0.01)

performs better than the other methods. This is due to the fact that Sparse K-means considers that all selected features are relevant for all clusters, but this is not the case for the X_2 data set which has a subspace structure. We also observe better performances of SSKM for DMU data. On the IS data set the SSKM and the K-means perform better than Sparse K-means and EWKM algorithms.

C. Evaluation of the feature relevance

Simulated data : Fig. 5, Fig. 6 and Fig. 7 represent the heatmaps of the features weights in the 4 clusters of the X_2 data set returned by the SSKM, EWKM and Sparse K-means algorithms. We note that the partitions whose heatmaps are shown below are those with the highest NMI (SSKM : 0.92, EWKM : 0.91 and Sparse KM : 0.92).

We see through the 4 most colored blocks of Fig. 5 and Fig. 6 that the highest weights assigned by SSKM and EWKM are indeed for the features defining the 4 clusters of the X_2 data set as presented in the (IV-A) section. We also observe that the majority of the noisy features weights in the clusters are characterized by the blocks in black showing the irrelevance of these features.

Fig. 7 represents the weights assigned by Sparse K-means to the different features of X_2. We observe that the majority of the weights of the features describing the c_2, c_3 and c_4 clusters are either close to zero or equal to zero. This may explain the poor performance of Sparse K-means. We emphasize that the two features ($V34$ and $V49$) with the highest weights are features describing respectively the clusters c_3 and c_4 according to the data simulation process.

Fig. 8, Fig. 9 and Fig. 10, they represent the heatmaps of the features weights in the 4 clusters of the X_1 data set returned by the SSKM, EWKM and Sparse K-means algorithms. Thus we observe on Fig. 8 and Fig. 10 that practically for each cluster a large number of features are relevant. This is justified by the fact that the data set X_1 has been simulated in such a way that all features are relevant for all clusters. Contrary to Fig. 8 and Fig. 10, we observe in Fig. 9 that a significant number of features are irrelevant for each cluster.

IS data : Fig. 11, Fig. 12 and Fig. 13 represent the heatmaps of the features weights in the 7 clusters of the IS data

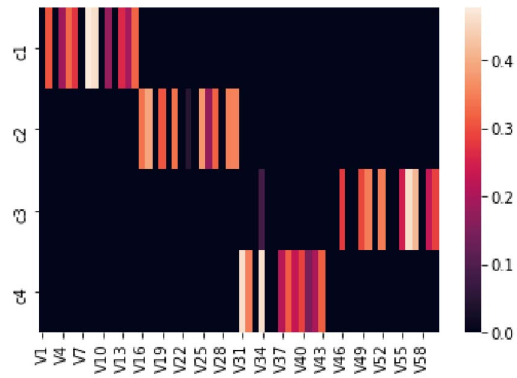

Fig. 5. SSKM

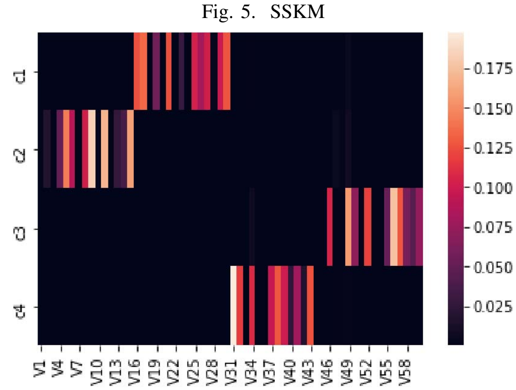

Fig. 6. EWKM

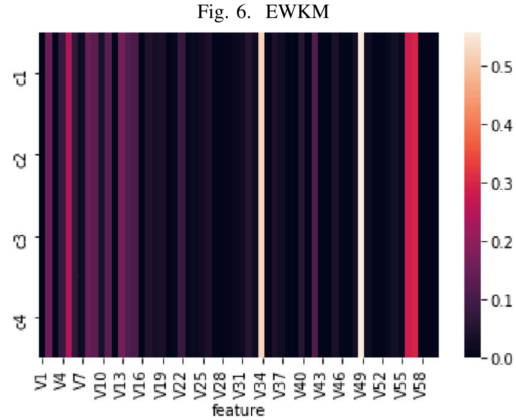

Fig. 7. Sparse KM

set returned by the SSKM, EWKM and Sparse K-means algorithms. Same as on the simulated data, the partitions whose heatmaps are shown below are those with the highest NMI (SSKM : 0.63, EWKM : 0.57 and Sparse KM : 0.58).

On the heatmaps, we observe for SSKM, that the clusters from c_1 to c_4 are more described by the features of the group $g2$. While the clusters c_5, c_6 and c_7 are described by a large part of the features of the groups $g1$ and $g2$.

For sparse K-means, we observe that all clusters are more described by the features in the $g2$ group that display the

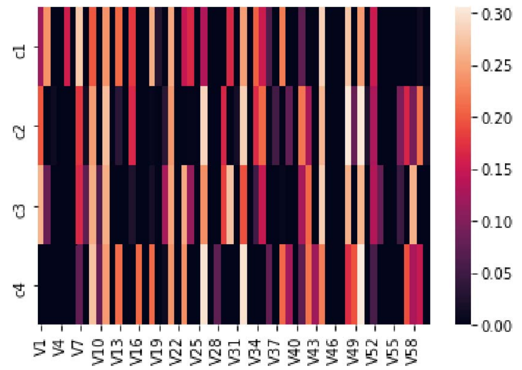

Fig. 8. SSKM

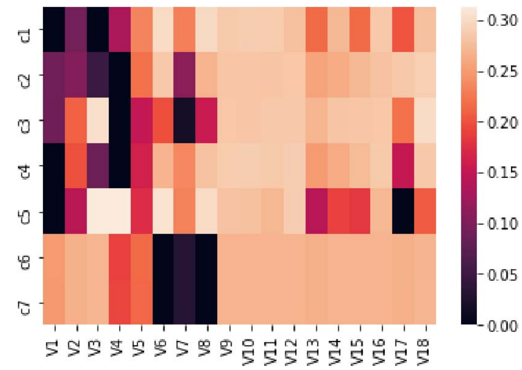

Fig. 11. SSKM

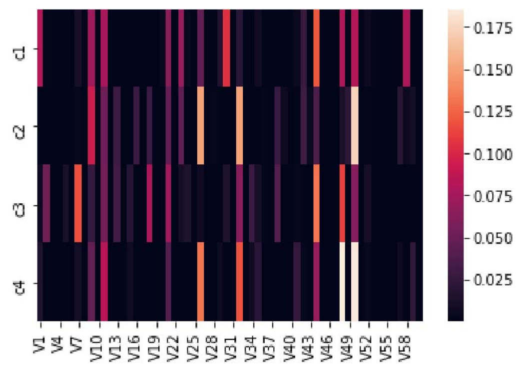

Fig. 9. EWKM

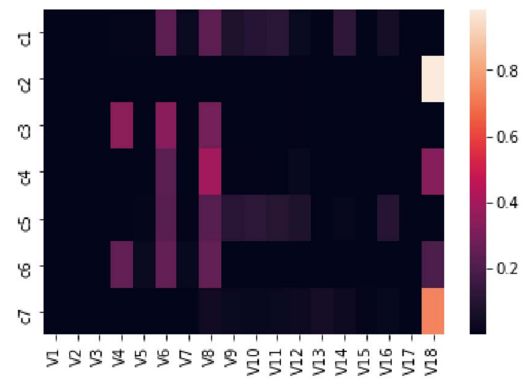

Fig. 12. EWKM

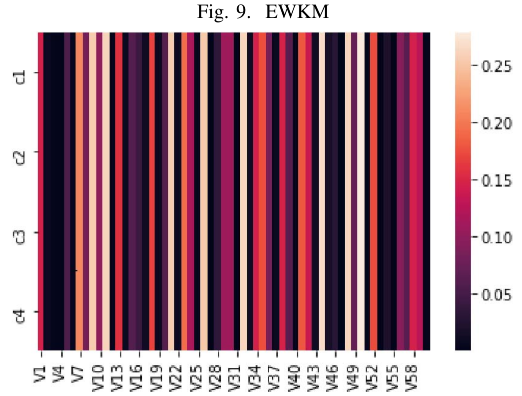

Fig. 10. Sparse KM

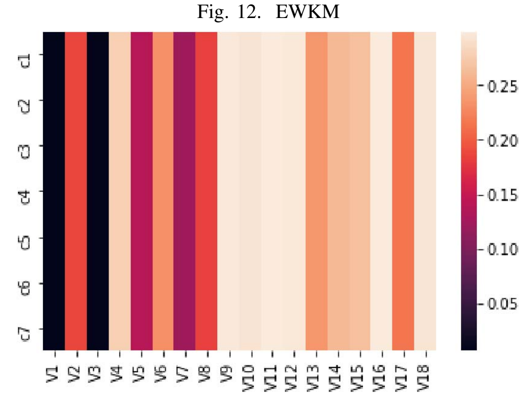

Fig. 13. Sparse KM

highest weights.

As for EWKM, we observe that the features of the group $g2$ display too low weights for the clusters c_2, c_3, c_4 and c_6 showing their low participation in the discovery of these clusters. We also observe that the clusters c_1, c_5 and c_7 are more characterized by the features of the $g2$ group.

DMU data : Fig. 14, Fig. 15 and Fig. 16 below represent the heatmaps of the features weights in the 10 clusters of the DMU data set returned by the SSKM, EWKM and Sparse K-means algorithms. The partitions whose heatmaps are shown above are those with the high NMI (SSKM : 0.83, EWKM : 0.64 and Sparse KM : 0.81).

On the heatmaps given above, we observe that for SSKM, except for c_1, the clusters are characterized essentially by $G4$ feature group. More precisely, these clusters are more described by the features of the $G4$ group and the features weights of the other groups are very low. The cluster c_1,is more described by the features of the groups $G1$ and $G2$.
For the EWKM algorithm we observe that a large part of the features weights of the different groups are very low.

For the Sparse K-means method we observe that the clusters are more described by the features of the groups $G1$, $G2$ and $G4$ contrary to the features of the groups $G3$ and $G5$ which

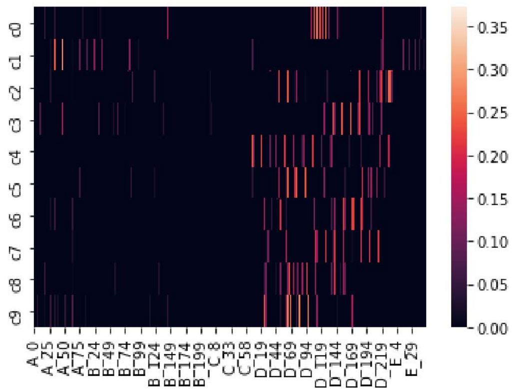

Fig. 14. SSKM

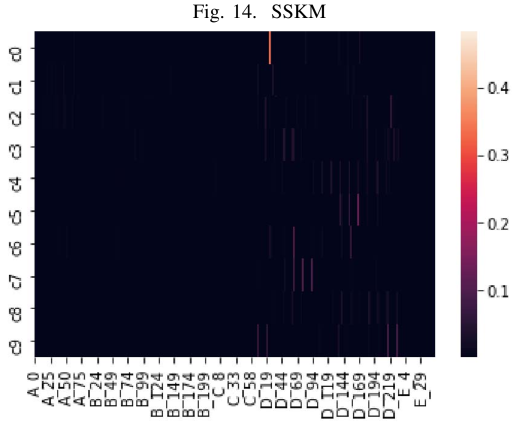

Fig. 15. EWKM

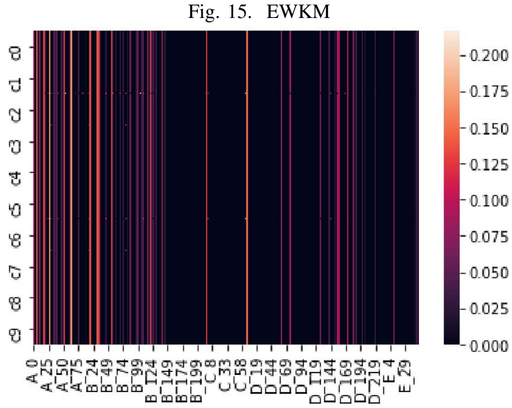

Fig. 16. Sparse KM

are in majority irrelevant for the clusters.

We can therefore conclude that the proposed SSKM method performs better in terms of the quality indices of the obtained partition. Moreover, SSKM determines subsets of features relevant to the clusters. More precisely, it allows to retain a small or relatively small percentage (between 7% and 13%) of the features considered as relevant while keeping good performances,thus facilitating the interpretation.

V. Conclusion

We have presented the Sparse Subspace K-means (SSKM) method which is an extension of the Sparse K-means method. This new method selects the relevant features for each cluster by setting the weights of irrelevant features to zero. The application of the SSKM method on simulated data and on real data yields very satisfying results in terms of clustering quality and cluster subspaces identification. Work is on progress to evaluate more intensively these performances. In some applications, features can be previously structured in blocks as in the case of DMU data. Future work, will be dedicated to the extention of SSKM to methods such as FGKM [11] taking into account the block structure and getting sparsity both for features and blocks of features.

References

[1] N. El Malki and F. Ravat and O. Teste, ”KD-means: Clustering Method for Massive Data based on KD-tree”, DOLAP 2020: 26-35.

[2] N. El Malki and R. Cugny and O. Teste and F. Ravat, ”Density-Based Clustering using Wasserstein Distance”, CIKM 2020: 2005-2008.

[3] L. Parsons and E. Haque and H. Lui, ”Subspace clustering for high dimensional data : a review”, SIGKDD Explor. Newsl, 90-105, 2004.

[4] H. Kriegel and P. Kröger and A. Zimek ”Clustering high-dimensional data: a survey on subspace clustering, pattern based clustering, and correlation clustering”, ACM Transactions on Knowledge Discovery from Data, 1–58, 2009.

[5] R. Agrawal and J. Gehrk and D. Gunopulos, and P. Raghavan, ”Automatic subspace clustering of high dimensional data for data mining applications”, In Proceedings of the 1998 ACM SIGMOD international conference on Management of data. ACM Press, 94–105, 1998.

[6] S. Goil and H. Nagesh, and A. Choudhary, ”Entropy-based subspace clustering for mining numerical data”, In Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM Press, 84–93, 1999.

[7] K.-G. Woo and J.-H. Lee, ”Finding generalized projected clusters in high dimensional spaces”, Korea Advanced Institute of Science and Technology, 2002.

[8] Z. Deng and K. Choi and Y. Jiang and J. Wang and S. Wang, ”A survey on soft subspace clustering”, Information Sciences, 84–106, 2016.

[9] Z. Huang and M. K. Ng and H. Rong and Z. Li, ”Automated variable weighting in k-means type clustering”, IEEE Transactions on pattern analysis and machine intelligence, 657-668, 2005.

[10] L. Jing and M. Ng and J. Huang, ”An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data”, Knowledge and Data Engineering, IEEE Transactions, 1026-1041, 2007.

[11] X. Chen and Y. Ye and X. Xu and J. Z. Huang, ”A feature group weighting method for subspace clustering of high-dimensional data”, Pattern Recognition, 434-446, 2012.

[12] C. H. Cheng and A. W. Fu, and Y. Zhang, ”Entropy-based subspace clustering for mining numerical data”, In Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM Press, 84–93, 1999.

[13] Ś. Brodinová and P. Filzmoser and T. Ortner and C. Breiteneder and M. Rohm, ”Robust and sparse k-means clustering for high-dimensional data”, Advances in Data Analysis and Classification, 905–932, 2019.

[14] X. Chang and Y. Wang and R. Li and Z. Xu, ”Sparse K-Means with l_{infty} / l_0 Penalty for High-Dimensional Data Clustering”, arXiv:1403.7890v1 [stat.ML], 2014.

[15] B. Xin, Y. Wang, W. Gao, and D. Wipf, ”Building Invariances Into Sparse Subspace Clustering”, IEEE Transactions on signal processing, vol. 66, No. 2, 2018.

[16] S. Tierneya, Y. Guob, J. Gaoa, ”Efficient Sparse Subspace Clustering by Nearest Neighbour Filtering”, arXiv:1704.03958v1 [cs.CV], 2017.

[17] M. Brbić and I. Kopriva, ”Multi-view Low-rank Sparse Subspace Clustering”, arXiv:1708.08732v1 [cs.CV], 2017.

[18] M. Abdolali, N. Gillis, M. Rahmati, ”Scalable and robust sparse subspace clustering using randomized clustering and multilayer graphs”, Signal Processing, 166–180, 2019.

[19] D. M. Witten and R. Tibshirani. "A framework for feature selection in clustering", Journal of the American Statistical Association, 2010.
[20] R. Tibshirani, "Regression Shrinkage and Selection via the Lasso", Journal of the Royal Statistical Society. Series B (Methodological), Volume 58, Issue 1 (1996), 267-288.
[21] H. Cheng and K. A. Hua and K. Vu, "Constrained locally weighted clustering", In Proceedings of the VLDB Endowment, vol.1, 90–101, 2008.
[22] A. Frank and A. Asuncion, UCI Machine Learning Repository. 2010.