

An Initialization for Supervised K-means

Vincent Lemaire, Oumaima Alaoui Ismaili, Antoine Cornuéjols

Abstract—Over the last years, researchers have focused their attention on a new approach, supervised clustering, that combines the main characteristics of both traditional clustering and supervised classification tasks. Motivated by the importance of the initialization in the traditional clustering context, this paper explores to what extent supervised initialization step could help traditional clustering to obtain better performances on supervised clustering tasks. This paper reports experiments which show that the simple proposed approach allows obtaining faster a good solution when considering the number of replicates to use.

I. INTRODUCTION

To discover the internal structure of huge collections of data, clustering algorithms have shown significant results over the past few years. These algorithms aim to identify heterogeneous groups (or clusters) in a manner that instances inside each one share the same characteristics (see Figure 1. a). This clustering problem has motivated a huge body of work and has resulted in a large number of algorithms (see e.g. [1, 2]).

On the other hand, to learn the link between a set of input features and an output feature (target class), classification algorithms are efficient. The main purpose of these algorithms is to construct a learning model which is able to predict class membership for new instances (see Figure 1. b).

Over the last years, researchers have focused their attention on the combination of characteristics of both clustering and classification tasks. The goal behind this combination is to find the internal structure of the target classes. This advanced research gives birth to a new approach called *Supervised clustering* (e.g. see [3, 4]). It aims to elaborate or modify clustering algorithms to find clusters where instances inside each cluster share the same characteristics and they are likely to have the same class label. The generated clusters are then labeled with the majority class of their instances (see Figure 1. c).

Among clustering methods (see e.g [1, 2]), some partitioning approaches such as K-means [5], K-medians [6], etc require an initialization step. The choice of an appropriate method of initialization is then important. Indeed, such step could have an impact on the quality of the obtained solution (intra-similarity) and on the computational efficiency [7]. In addition to this, by using a good method of centers initialization, clustering algorithms need just a lowest number of iterations (or replicates) to yield an optimal result (partition having a lowest intra-inertia value). Inspired by this importance, it's natural to ask: *Could supervised initialization*

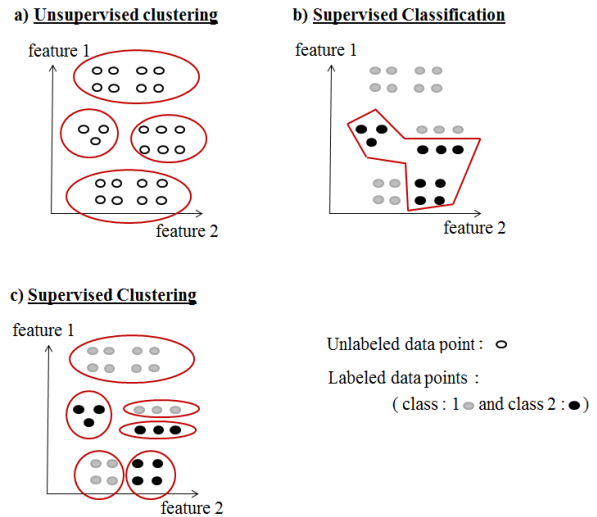


Fig. 1: Classification processes

method help traditional clustering algorithms to reach a good predictive performance in a supervised clustering context? In other words, does introducing the information given by the target class in a centers initialization step produce a good supervised clustering, meaning exhibiting high value of the Adjusted Rand Index (ARI)¹ [8] while at the same time uncovering interesting clusters in the data set.

To be able to answer the question above, we present firstly a supervised initialization method for the traditional K-means algorithm. This method uses the additional information given by the target class to get an appropriate initial vector of centers. Secondly, we compare this method with others unsupervised initialization methods from the literature using both supervised and unsupervised criteria such as the ARI and the Mean Squared error (MSE) (see Section V).

The remainder of this paper is organized as follows. Section 2 presents the main idea of the K-means clustering algorithm. Section 3 describes briefly related works about supervised K-means algorithm. Section 4 presents the proposed method of initialization for the traditional K-means algorithm with a didactic presentation using a toy problem. Section 5 compares the performance of the generated partitions using supervised and unsupervised initialization step with ARI, Accuracy (ACC) and MSE. Finally, a conclusion with future works is presented in the last section.

¹The ARI criterion is a member of the family of the external criterion which estimates the quality in reference to an additional variable, here is the target variable. It is computed by comparing the partition of the target class labels with the partition of the k-means algorithm.

II. K-MEANS-ALGORITHMS

The purpose of the partitioning algorithms is to construct a partition of N objects into a set of K clusters. These algorithms require an initialization phase of centers and differ by the way of how to get the centers. The most common algorithms of this category of clustering are: K -means [5], K medians [6], K -modes [9] and K medoids [10]. Each of these algorithms depends on the number of clusters K (fixed a priori), the input features, the initial vector of centers (k_1, \dots, k_K) , the similarity measure and the criterion used to evaluate the quality of the partition.

The choice of one of these algorithms depends on: (i) the nature of datasets, (ii) the desired result (mean, medoid ...), and (iii) the complexity of the algorithm. In this paper, we are focused on one of the most widely used algorithm: K -means algorithm.

A. K -means algorithm

The procedure of the K -means algorithm (KM) follows an iteratively way to improve the quality of the partition:

```

K-means(K, input features, number of replicates R)
for 1 to R (Replicates) do
    Initialize  $K$  cluster centroids (see Section II-B).
    while The centroid vectors change do
        • Associate each data instance to the closest centroid
          using the Euclidean distance (L2 norm)
        • Update each centroid vector by computing the
          average of its associated instances
    end
end

```

Output: Return the best solution among the R results in the sense of the Mean Squared Error (MSE).

Algorithm 1: K -means algorithm.

B. Unsupervised initialization

The cluster initialization centers have an impact on the quality of the generated partition and also on the computational efficiency [7]. In addition to this, the choice of an inappropriate initialization method could generate adverse effects such as: (i) empty clusters, (ii) slower convergence, and (iii) a higher chance of getting stuck in a bad local minimum and thus, the need of running the algorithm for another time. For all of these reasons, the choice of an initialization method has been well studied in the literature (e.g. for K -medians in [11], for K -medoids in [12] and for k -means in [13, 14, 15]).

In this section, we present a brief overview of the most common initialization methods for the k -means algorithm, with an emphasis on their computational efficiency (these methods are described in [13], [14] or [15]).

- **Random [16] (RAND):** The cluster centers are chosen randomly from the data set; The complexity of this scheme is $\mathcal{O}(K)$.

- **Splitting [17] (SPLIT):** The first center k_1 is chosen as the centroid of the data set. At iteration i ($i \in \{1, 2, \dots, \log_2 K\}$), each of the existing 2^{i-1} centers is split into two new centers by subtracting and adding a fixed perturbation vector ϵ , i.e. $k_j - \epsilon$ and $k_j + \epsilon$, ($j \in \{1, 2, \dots, 2^{i-1}\}$). These 2^i new centers are then refined using the KM algorithm. The complexity of this scheme is $\mathcal{O}(NK)$.
- **Minmax [18] (MINMAX):** The first center k_1 is chosen randomly and the i -th ($i \in \{2, 3, \dots, K\}$) center k_i is chosen to be the point that has the largest minimum distance to the previously selected centers, i.e. k_1, k_2, \dots, k_{i-1} . The complexity of this scheme is $\mathcal{O}(NK)$.
- **Density-based [19] (DENS):** The data space is partitioned uniformly into M cells. From each of these cells, a number (that is proportional to the number of points in this cell) of centers is chosen randomly until K centers are obtained. The complexity of this scheme is $\mathcal{O}(N)$.
- **Sample (SAMPLE):** This method consists in taking a sample, N' , of the data set (often 10%) and in applying the k -mean algorithm to this sample. Then to take the centers found as initial centers. The complexity of this scheme is $\mathcal{O}(N'kq)$, where q is the number of iteration.
- **Kmeans++ [20] (K++):** The first center k_1 is chosen randomly and the i -th ($i \in \{2, 3, \dots, K\}$) center k_i is chosen to be $x \in X$ with a probability of $\frac{Q(x)^2}{\sum_{i=1}^n Q(x_i)^2}$, where $Q(x)$ denotes the minimum distance from a point x to the previously selected centers. The complexity of this scheme is $\mathcal{O}(NK)$.

III. RELATED WORK : SUPERVISED K -MEANS

In the last decade, many researchers focused their attention to build or modify standard clustering to identify class-uniform clusters where instances within each cluster are homogeneous. Several algorithms are developed to achieve that objective (e.g. [21, 22, 23, 24, 25, 26, 27]).

In this section, we present two of the most cited methods for supervised K -means algorithms. These algorithms incorporate the additional information given by the target class in middle of the algorithm (i.e. in the “while loop” of the Algorithm 1). However, to our knowledge, there are no methods which used a *supervised* initialization step to choose the adequate centers.

Al-Harbi et al. [3] developed a K -means algorithm in such a way to use it as a classifier algorithm. First of all, they replaced the Euclidean metric used in a standard K -means by a weighted Euclidean metric. This modification is carried out in order to be able to estimate the distance between any two instances that have the same class label. The vector of weights is chosen in such a way to maximize the confidence of the partitions generated by the k -means algorithm. This confidence is determined by calculating the percentage of correctly classified objects with respect to the total number of objects in the data set. To solve this problem of optimization, they used Simulated Annealing (a generic probabilistic meta-

heuristic for the global optimization problem). This iterative process is repeated until an optimal confidence is obtained. In this algorithm, the number of clusters is an input.

Eick et al. [4] introduced four representative-based algorithms for supervised clustering: *SRIDHCR*, *SPAM*, *TDS* and *SCEC*. In their experimentation, they used the first one (i.e. *SRIDHCR*). The greedy algorithm *SRIDHCR* (or Single Representative Insertion/Deletion Steepest Decent Hill Climbing with Randomized Start) is mainly based on three phases. The first one is the initialization of a set of representatives that is randomly selected from the dataset. The second is the primary cluster creation phase, where instances are assigned to the cluster of their closest representative. The third one is the iteration phase where the algorithm is run r times: In each time 'r', the algorithm tries to improve the quality of clustering, for instance, by adding a non-representative instance or by deleting a representative instance. To measure this quality, they use a supervised criterion. It takes into account two points: (i) The impurity of the clustering which defined as a percentage of misclassified observations in the different clusters and (ii) a penalty condition which used in a manner to keep a lowest number of clusters. In this greedy algorithm, the number of clusters is an output.

IV. PROPOSED METHOD : SUPERVISED INITIALIZATION

In this paper, we suggest that on way to help the traditional K -means in a supervised context is to integrate the target class information into the initialization process. That is, we suggest that an efficient supervised initialization approach allows obtaining a good performance in terms of (i) computational efficiency, and (ii) prediction quality. In addition to this, with such a supervised method, (i) the chance of falling in a bad local minima **“in the sense of supervised clustering”** is lower and (ii) the number of replicates (number of iterations) is minimized (i.e. R in Algorithm 1). Therefore, we can obtain a good solution with only a small number of replicates. To test the validity of all these points, we propose a simple supervised initialization method.

The proposed method called **K++R** is an exploit and an explore mechanism: the information given by the class label is firstly exploited and then the density of the data distribution is explored. The main idea of this method is to dedicate one center per class (as a *“Rochio”* [28] solution). Each center is defined as the average vector of instances which have the same class label. If the number of clusters (K) exceeds the number of classes (C), the initialization continues using the K -means++ [20] algorithm for the $K - C$ remaining centers in such a way to add diversity. This initialization method could be performed just in the case² where $K \geq C$ since it takes into account the cardinality of data classes (C). The complexity of this scheme is $\mathcal{O}((N+(K-C)N) < \mathcal{O}(NK)$.

Clearly, this initialization approach is not suitable for a traditional clustering: it would deteriorate the quality of the

²In the context of supervised clustering there is no sense to cluster instances in K clusters where $K < C$

generated partitions in terms of reconstruction error (the Mean Squared Error (**MSE**)). But, let us recall that the objective of a supervised clustering is to make a trade-off between similarity and prediction. At this stage

- the intra similarity “quality” is guaranteed by the standard K -means algorithm and measured by the MSE criterion.
- the supervised “quality” is guaranteed by the supervised initialization step (our suggestion) and measured by a suitable supervised criterion (to be defined).

To be consistent with the definition of a supervised clustering, we select a criterion that allows to choose the closest partition to the one given by the target class. For this, we use the Adjusted Rand Index (**ARI**) [8] to measure the quality of the obtained partition [29]. It measures the agreement between the generated partition for the K -means and the partition given by the target class.

To highlight the strong points of our proposed approach, we illustrate it on a toy dataset (see Figure 2) using the standard K -means. This dataset contains 510 instances, distributed on five classes ($C = 5$) : 'ear left' (blue), 'ear right' (green), 'head' (red), 'shoulder' (cyan) and 'noise' (purple); Each class contains respectively 290, 100, 100, 10 and 10 instances. Here, the standard K -means is applied in the following conditions: L2 as a norm, mean as a centroid, Mean Squared Error (MSE) as a criterion to select the best replicates and statistical normalization (SN)³ as a preprocessing step.

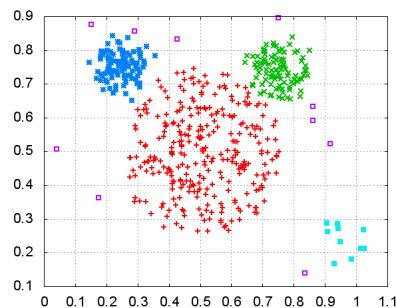


Fig. 2: Mouse dataset.

MSE criterion : Using the standard K -means, the emplacement of the initial center has often an impact on the quality of the generated partition.

Our supervised initialization approach requires at least an initial center per class ($K \geq C$). However, for this dataset, the initialization of a center in the 'cyan' class could deteriorate the quality of the final solution in terms of MSE (the initial center couldn't move outside the cyan points).

Figure 3 illustrates this point: the generated partitions using the usual initialization approaches (Rand, Sample and $K++$) have a better performance in terms of MSE than

³This approach transforms data derived from any normal distribution into a standard normal distribution $N(0,1)$. The formula that allows the transformation of feature X_u is: $X'_u = \frac{X_u - \mu}{\sigma}$ where μ is the mean of the feature u , σ is its standard deviation.

the K++R approach where $K=C$. Note that, the difference between the MSE of K++ and K++R is only 8.9% (0.326 against 0.355) when the number of replicates is equal to 1000 and 1 respectively. The last point shows that the K++R is a faster algorithm than the others.

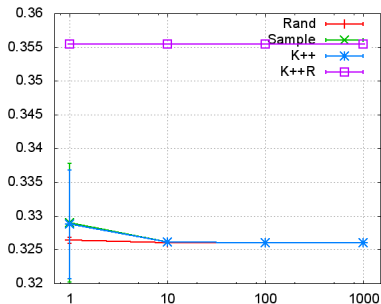


Fig. 3: Mouse dataset: Reconstruction Error versus Number of replicates ($K=5$). The points represent the mean result obtained over 10 tries and the error bars the standard deviation ($\nu \pm \sigma$).

ARI criterion: For this example, when the number of clusters is equal to the cardinality of the target class, using an unsupervised initialization approach, the chance to select more than one seed in the same class is high (especially in the red class). Likewise, the chance to select one center in the magenta class is lower. Consequently, the predictive performance of the generated partition could be deteriorated (e.g lower value of ARI).

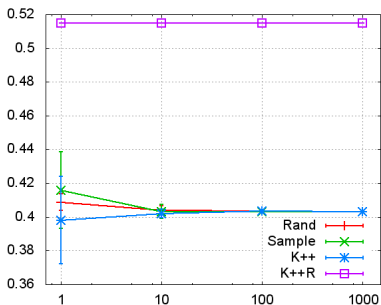


Fig. 4: Mouse dataset: ARI versus Number of replicates ($K=5$). The points represent the mean result obtained over 10 tries and the error bars the standard deviation ($\nu \pm \sigma$).

Figure 4 describes the evolution of the ARI versus the number of replicates (in the case where $K = C$) using both K++R and the usual initialization approaches. In this case, our approach is deterministic (i.e. it gives one unique value of ARI whatever the number of replicates). This result shows that, using the ARI criterion, the K++R approach is better than the other unsupervised approaches.

In the case where $K \geq C$, Figure 5 presents the evolution of the ARI versus the number clusters. This result shows that, using the proposed approach, the optimal generated partition in terms of ARI is reached where $K = C$. Likewise, when

$K > C$, the K++R has the same predictive performance 'quality' as the unsupervised approaches.

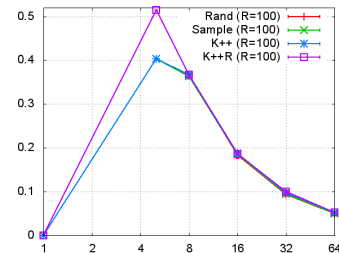


Fig. 5: Mouse dataset: ARI versus Number of clusters (K). The points represent the mean result obtained over 10 tries. The small error bars corresponding to the standard deviation ($\nu \pm \sigma$) is not visible.

The preliminary experiment conducted on this dataset has shown the following result: (1) the predictive performance of the generated partition using the proposed initialization method is better than the obtained performance using the usual initialization approaches (when $K = C = 5$), (2) K++R is faster than the others approaches (reach a good performance with a few number of replicates), and (3) unsupervised initialization approaches are better than K++R in terms of MSE.

To further evaluate the efficiency of our proposed approach in a supervised context, we propose an extensive comparison with the usual initialization approaches for several benchmark datasets.

V. EXPERIMENTS

In this section, we present and compare the average performance of the traditional K -means algorithm using both supervised and unsupervised initialization approaches.

A. Protocol

1) *Initialization method:* To test the validity of our suggestion, we compare our supervised approach of initialization (K++R) to the three most popular unsupervised approaches (see Section II-B) such as Rand (as a baseline), Sample (which exhibits interesting performance as described in [14]) and K++ (which has theoretical foundations and a good performance while keeping a reasonable complexity).

2) *Datasets:* To evaluate and compare the behavior of different initialization approaches in terms of their capacity to help the traditional clustering in a supervised context, some tests are performed on different datasets of the UCI repository [30]. Table I presents the used datasets in this study. We chose these datasets in a manner to get some diversity in terms of number of clusters, number of features (categorical and continuous) and number of instances.

3) *Number of clusters:* In this paper⁴, we deal the case where K is an input (as in [3]): In this study, K is equal to the cardinality of the target class (C).

⁴For place consideration, the case where K is an output (as in [4]) will be presented in a future publication

TABLE I: The used datasets from UCI: M_n : Number of numerical variables; M_c : Number of categorical variables; N : Number of instances; C : number of Classes; Maj. acc: Baseline performance indication using a classifier based only on the majority class.

Id	Name	M_n	M_c	N	C	Maj. acc.
1	Iris	4	0	150	3	33.33
2	Hepatitis	6	13	155	2	79.35
3	Glass	10	0	214	6	35.51
4	Heart	10	3	270	2	55.56
5	Horsecolic	7	20	368	2	63.04
6	Soybean	0	35	376	19	13.83
7	Pima	8	0	768	2	65.10
8	Vehicle	18	0	846	4	25.77
9	Tictactoe	0	9	958	2	65.34
10	LED	7	0	1000	10	11.40
11	Phoneme	256	0	2254	5	25.95
12	Segmentation	19	0	2310	7	14.29
13	Abalone	7	1	4177	28	16.50
14	Waveform	21	0	5000	3	33.92
15	PenDigits	16	0	10992	10	10.41

4) *Preprocessing*:: In a previous paper [31], we prove that one way to help the standard K-means algorithm to reach a good predictive performance is to incorporate the additional information given by the target class in a pre-processing step. In this study, we used one of these supervised preprocessing approaches, called "Conditional Info".

The following notation is used below: Let $D = \{(X_i, Y_i)\}_1^N$ denote a training dataset of size N , where $X_i = \{X_{i1}, \dots, X_{id}\}$ is a vector of d features and $Y_i \in \{1, \dots, N\} \in \{Class_1, \dots, Class_C\}$ is the target class of size C . Let K denote the number of clusters.

This preprocessing method is based on two steps: (1) supervised representation and (2) recoding. The first one aims at giving information about features distribution conditionally to a target class. To achieve this objective, the *MODL* (a bayes optimal pre-processing method for continuous and categorical features) approach is used. It seeks to estimate the univariate conditional density (i.e. $P(X_{im}|Class_j)$, where $m \in \{1, \dots, d\}$ and $j \in \{1, \dots, C\}$). To obtain this estimation a supervised discretization method is used for continuous features [32] and a supervised grouping method is used for categorical ones [33]. To exploit the information given by the first step, a recoding phase is then required.

Each feature from the instance X_i is recoded in a qualitative attribute containing I_C recoding values. The resulting vector for this instance is $X_i = X_{i1_1}, \dots, X_{i1_C}, \dots, X_{id_1}, \dots, X_{id_C}$. Where $X_{id_1}, \dots, X_{id_C}$ represent the recoding values for the feature d with respect to the number of a class label ($X_{id_C} = \log(P(X_{id}|Class_C))$). As a result, the initial vector containing d features (continuous and categorical) becomes a vector (XR) containing $d \times C$ real components: $\log(P(X_{im}|Class_j)), j \in \{1, \dots, C\}, m \in \{1, \dots, d\}$.

To give an idea of the good impact of this preprocessing step, using the previous dataset (Figure 2), the obtained performance 'quality' (using ARI criterion) is equal to 0.792 compared to 0.514 (using SN preprocessing step).

5) *Cross validation*: In order to compare the obtained results, a '10 × 5' fold cross validation has been performed on all datasets. Thus, the results are presented as an average of 50 tests (e.g. see table III).

B. Results

1) *Evolution of the error reconstruction*: The generated partition contains K centers (k_1, \dots, k_K). Each center is a vector of Z features, where Z is the number of features after the supervised preprocessing process. So, for a given dataset, the value of Z is then equal to $d \times C = (M_n + M_c) \times C$ (see Table I).

To evaluate the quality of the generated partition in terms of instances similarity in each cluster, the Mean Squared Error is used (see equation 1).

$$MSE = \frac{1}{N} \frac{1}{Z} \frac{1}{K} \sum_{i=1}^N \sum_{z=1}^Z \sum_{t=1}^K \sum_{i \in k_t} (XR_{i,z} - k_{t,z})^2 \quad (1)$$

Table II presents the comparison results (in percentage) between the MSE of K++ and the others initialization approaches (i.e. Rand, Sample and K++R) when $R=1$ (see Algorithm 1). In this table, a positive percentage indicates that the MSE of the K++ is lower (better) than it corresponding initialization method. This results show that, with a single replicate, K++ is better than the two other unsupervised initialization methods and it is competitive to K++R.

TABLE II: Comparison in percentage between the MSE of K++ and the others methods when using a single replicate(R)

Database	K++/Rand	K++/Sample	K++/K++R
Iris	56%	44%	0%
Hepatitis	0%	0%	0%
Glass	4%	2%	2%
Heart	4%	4%	0%
Horsecolic	0%	0%	-3%
Soybean	27%	27%	-12%
Pima	-2%	-2%	0%
Vehicle	8%	7%	2%
Tictactoe	0%	0%	-14%
LED	14%	11%	-17%
Phoneme	0%	0%	-1%
Segmentation	4%	4%	-3%
Abalone	19%	18%	30%
Waveform	0%	0%	0%
PenDigits	1%	1%	-2%

When the value of replicates increase (i.e. $R \in \{1, 10, 100, 1000\}$), the average value of MSE for the three unsupervised initialization methods decrease; especially for Random and Samples methods (see Table III). This improvement is important when going from one to 10 replicates, less important from 10 to 100 replicates and it remains nearly unchanged when going from 100 to 1000 for all datasets except for Soybean, LED and Abalone datasets.

Among the three unsupervised initialization method, K++ is the best one. However, it is recommended to use it with a number of replicates at least equal to 10 or greater for certain datasets.

For the K++R method, its average value of MSE remains unchanged for all datasets whatever the number of replicates

TABLE III: MSE versus the number of replicates (R)

Dataset	Rand	Sample	K++	K++R
R=1				
Iris	4.74±3.62	4.37±3.61	3.03±1.59	3.04±1.43
Hepatitis	0.22±0.08	0.22±0.08	0.22±0.08	0.22±0.08
Glass	13.37±2.40	13.10±2.31	12.80±2.52	13.03±2.45
Heart	0.26±0.05	0.26±0.05	0.25±0.04	0.25±0.04
Horsecolic	0.30±0.05	0.30±0.05	0.30±0.05	0.29±0.05
Soybean	35.23±4.50	35.16±4.68	27.71±3.40	24.27±2.16
Pima	0.42±0.11	0.42±0.11	0.43±0.12	0.43±0.11
Vehicle	7.79±1.32	7.78±1.31	7.23±1.35	7.35±1.24
Tictactoe	0.07±0.02	0.07±0.02	0.07±0.02	0.06±0.02
LED	4.41±0.54	4.29±0.52	3.86±0.43	3.21±0.35
Phoneme	22.98±0.73	22.97±0.72	22.97±0.68	22.85±0.69
Segmentation	36.82±4.15	36.79±4.16	35.40±3.17	34.19±1.32
Abalone	33.95±3.65	33.67±3.64	28.59±2.51	37.08±4.03
Waveform	2.11±0.51	2.11±0.51	2.12±0.54	2.13±0.46
PenDigits	58.44±2.27	58.33±2.30	57.94±1.94	56.66±1.59
R=10				
Iris	2.81±1.43	2.81±1.43	2.80±1.43	3.04±1.43
Hepatitis	0.21±0.08	0.21±0.08	0.21±0.08	0.22±0.08
Glass	12.22±2.39	12.25±2.41	12.15±2.37	13.03±2.45
Heart	0.25±0.04	0.25±0.04	0.25±0.04	0.25±0.04
Horsecolic	0.29±0.05	0.29±0.05	0.29±0.05	0.29±0.05
Soybean	29.43±2.32	29.16±2.27	25.06±1.95	24.27±2.16
Pima	0.40±0.10	0.40±0.10	0.39±0.07	0.43±0.11
Vehicle	6.46±0.83	6.48±0.82	6.41±0.83	7.35±1.24
Tictactoe	0.06±0.02	0.06±0.02	0.06±0.02	0.06±0.02
LED	3.68±0.42	3.64±0.39	3.50±0.34	3.21±0.35
Phoneme	22.85±0.69	22.85±0.69	22.85±0.69	22.85±0.69
Segmentation	33.95±1.33	33.93±1.33	33.71±1.26	34.19±1.32
Abalone	30.75±2.58	30.41±2.55	26.72±1.82	35.42±4.07
Waveform	2.06±0.45	2.06±0.45	2.06±0.45	2.13±0.46
PenDigits	56.41±1.62	56.43±1.63	56.39±1.61	56.66±1.59
R=100				
Iris	2.80±1.43	2.80±1.43	2.80±1.43	3.04±1.43
Hepatitis	0.21±0.08	0.21±0.08	0.21±0.08	0.22±0.08
Glass	12.05±2.28	12.02±2.31	12.09±2.26	13.03±2.45
Heart	0.25±0.04	0.25±0.04	0.25±0.04	0.25±0.04
Horsecolic	0.29±0.05	0.29±0.05	0.29±0.05	0.29±0.05
Soybean	27.12±2.45	27.02±2.37	23.88±2.21	24.27±2.16
Pima	0.39±0.07	0.39±0.07	0.39±0.07	0.43±0.11
Vehicle	6.41±0.83	6.41±0.83	6.41±0.83	7.35±1.24
Tictactoe	0.06±0.02	0.06±0.02	0.06±0.02	0.06±0.02
LED	3.48±0.35	3.45±0.34	3.35±0.34	3.21±0.35
Phoneme	22.85±0.69	22.85±0.69	22.85±0.69	22.85±0.69
Segmentation	33.60±1.25	33.60±1.25	33.61±1.25	34.19±1.32
Abalone	28.92±2.32	28.65±2.27	25.76±1.78	34.94±4.27
Waveform	2.06±0.45	2.06±0.45	2.06±0.45	2.13±0.46
PenDigits	56.23±1.55	56.23±1.55	56.23±1.55	56.66±1.59
R=1000				
Iris	2.80±1.43	2.80±1.43	2.80±1.43	3.04±1.43
Hepatitis	0.21±0.08	0.21±0.08	0.21±0.08	0.22±0.08
Glass	12.06±2.27	12.05±2.26	12.05±2.26	13.03±2.45
Heart	0.25±0.04	0.25±0.04	0.25±0.04	0.25±0.04
Horsecolic	0.29±0.05	0.29±0.05	0.29±0.05	0.29±0.05
Soybean	25.56±2.18	25.30±2.22	23.21±2.38	24.27±2.16
Pima	0.39±0.07	0.39±0.07	0.39±0.07	0.43±0.11
Vehicle	6.41±0.83	6.41±0.83	6.41±0.83	7.35±1.24
Tictactoe	0.06±0.02	0.06±0.02	0.06±0.02	0.06±0.02
LED	3.37±0.34	3.37±0.34	3.25±0.34	3.21±0.35
Phoneme	22.85±0.69	22.85±0.69	22.85±0.69	22.85±0.69
Segmentation	33.61±1.25	33.61±1.25	33.61±1.25	34.19±1.32
Abalone	28.07±1.80	27.72±1.87	25.32±1.90	34.79±4.40
Waveform	2.06±0.45	2.06±0.45	2.06±0.45	2.13±0.46
PenDigits	56.23±1.55	56.23±1.55	56.23±1.55	56.66±1.59

(see Table III) Indeed, when the number of clusters is equal to the number of classes, the proposed method is a determinist approach (i.e. it follows the Rochio solution (see Section IV)).

From these results, we compare now the quality of the generated partition in terms of MSE, using in each time K++ and K++R. Table IV presents the comparison results

TABLE IV: Comparaisn in percentage between the MSE of K++ and K++R for R=1, 10, 100, 1000.

Database	R=1	R=10	R=100	R=1000
Iris	0%	9%	9%	9%
Hepatitis	0%	5%	5%	5%
Glass	2%	7%	8%	8%
Heart	0%	0%	0%	0%
Horsecolic	-3%	0%	0%	0%
Soybean	-12%	-3%	2%	5%
Pima	0%	10%	10%	10%
Vehicle	2%	15%	15%	15%
Tictactoe	-14%	0%	0%	0%
LED	-17%	-8%	-4%	-1%
Phoneme	-1%	0%	0%	0%
Segmentation	-3%	1%	2%	2%
Abalone	30%	33%	36%	38%
Waveform	0%	3%	3%	3%
PenDigits	-2%	0%	1%	1%

(in percentage) for the MSE of K++R and K++R, for $R \in \{1, 10, 100, 1000\}$. This table show that when we combine the proposed initialization method with the supervised preprocessing approach (see Section V-A), K++R shows competitive to K++ method in terms of MSE when R=1 (i.e. it exhibits better or similar MSE than K++). When R increases, K++R approach exhibits MSE not exceeding 5% for 10 datasets.

2) *Evolution of the ARI*: The above experiments show that the proposed method can reach a good performance in terms of MSE comparing to the other popular unsupervised initialization methods. Now, to achieve the goal of the supervised clustering, we have to prove now that the traditional K-means with the proposed method could also reach a good predictive performance (using a supervised criterion).

Table V presents the average predictive performance (using ARI criterion) of the partition of the traditional K-means using supervised and unsupervised initialization method. From this results, we can see that, whatever the number of replicates R, the proposed method remains the best one for 9 datasets and similar to the others but with one replicate (i.e the traditional K-means with the proposed method is 10 times (or more) faster than using K++).

Now, if we choose another supervised criterion to measure the quality of the generated partitions, the obtained conclusion remains the same. Figures 6 and 7 present respectively, for all used datasets, the quality of the generated partitions using Accuracy⁵ (ACC) and Balanced Accuracy⁶ (BACC)criterion, in the case R=1000 for 'Rand', 'Sample' and 'K++' and where R=1 for 'K++R'.

VI. CONCLUSION

This paper has presented the influence of the supervised initialization step on the performance of the traditional K-means algorithm in terms of predictions (using ARI and ACC criteria). The experimental results show that the proposed

⁵the accuracy criterion is computed by comparing the true class and the predicted class which is the majority class of the examples belonging to a cluster

⁶This criterion avoids inflated performance estimates on imbalanced datasets.

TABLE V: ARI versus the number of replicates (R)

Dataset	Rand	Sample	K++	K++R
R=1				
Iris	0.59±0.17	0.61±0.20	0.65±0.16	0.78±0.12
Hepatitis	0.15±0.16	0.15±0.16	0.17±0.18	0.21±0.15
Glass	0.47±0.15	0.48±0.14	0.48±0.15	0.78±0.10
Heart	0.32±0.18	0.32±0.18	0.29±0.18	0.36±0.15
Horsecolic	0.33±0.17	0.33±0.17	0.37±0.12	0.39±0.10
Soybean	0.45±0.08	0.44±0.07	0.50±0.07	0.63±0.07
Pima	0.04±0.08	0.04±0.08	0.05±0.08	0.10±0.12
Vehicle	0.16±0.04	0.16±0.04	0.16±0.03	0.20±0.09
Tictactoe	0.09±0.07	0.09±0.07	0.08±0.06	0.14±0.05
LED	0.37±0.06	0.37±0.06	0.40±0.05	0.52±0.05
Phoneme	0.71±0.04	0.71±0.04	0.71±0.04	0.72±0.02
Segmentation	0.59±0.08	0.59±0.08	0.60±0.07	0.70±0.04
Abalone	0.05±0.01	0.05±0.01	0.05±0.01	0.06±0.00
Waveform	0.20±0.06	0.20±0.06	0.20±0.06	0.22±0.08
PenDigits	0.53±0.05	0.53±0.05	0.53±0.04	0.62±0.02
R=10				
Iris	0.68±0.15	0.68±0.15	0.67±0.15	0.78±0.12
Hepatitis	0.10±0.12	0.10±0.12	0.10±0.12	0.21±0.15
Glass	0.47±0.15	0.47±0.15	0.47±0.14	0.78±0.10
Heart	0.29±0.18	0.29±0.18	0.29±0.18	0.36±0.15
Horsecolic	0.39±0.10	0.39±0.10	0.39±0.10	0.39±0.10
Soybean	0.49±0.07	0.48±0.07	0.51±0.08	0.63±0.07
Pima	0.01±0.02	0.01±0.02	0.00±0.01	0.10±0.12
Vehicle	0.17±0.03	0.17±0.03	0.17±0.03	0.20±0.09
Tictactoe	0.13±0.06	0.13±0.06	0.13±0.06	0.14±0.05
LED	0.44±0.05	0.43±0.05	0.45±0.04	0.52±0.05
Phoneme	0.72±0.02	0.72±0.02	0.72±0.02	0.73±0.02
Segmentation	0.66±0.06	0.66±0.07	0.69±0.06	0.70±0.04
Abalone	0.05±0.01	0.05±0.01	0.05±0.01	0.06±0.01
Waveform	0.21±0.04	0.21±0.04	0.21±0.04	0.22±0.08
PenDigits	0.56±0.03	0.56±0.03	0.56±0.02	0.62±0.02
R=100				
Iris	0.67±0.15	0.67±0.15	0.67±0.15	0.78±0.12
Hepatitis	0.10±0.13	0.10±0.13	0.10±0.13	0.21±0.15
Glass	0.48±0.14	0.49±0.13	0.47±0.14	0.78±0.10
Heart	0.29±0.18	0.29±0.18	0.29±0.18	0.36±0.15
Horsecolic	0.39±0.10	0.39±0.10	0.39±0.10	0.39±0.10
Soybean	0.52±0.09	0.51±0.08	0.55±0.07	0.63±0.07
Pima	0.00±0.01	0.00±0.01	0.00±0.01	0.10±0.12
Vehicle	0.17±0.03	0.17±0.03	0.17±0.03	0.20±0.09
Tictactoe	0.13±0.06	0.13±0.06	0.13±0.06	0.14±0.05
LED	0.46±0.04	0.46±0.04	0.48±0.04	0.52±0.05
Phoneme	0.72±0.02	0.72±0.02	0.72±0.02	0.73±0.02
Segmentation	0.70±0.05	0.70±0.05	0.70±0.05	0.70±0.04
Abalone	0.05±0.01	0.05±0.01	0.05±0.00	0.06±0.00
Waveform	0.21±0.04	0.21±0.04	0.21±0.04	0.22±0.08
PenDigits	0.55±0.01	0.55±0.01	0.55±0.01	0.62±0.02
R=1000				
Iris	0.67±0.15	0.67±0.15	0.67±0.15	0.78±0.12
Hepatitis	0.10±0.13	0.10±0.13	0.10±0.13	0.21±0.15
Glass	0.48±0.14	0.48±0.14	0.48±0.14	0.78±0.10
Heart	0.29±0.18	0.29±0.18	0.29±0.18	0.36±0.15
Horsecolic	0.39±0.10	0.39±0.10	0.39±0.10	0.39±0.10
Soybean	0.53±0.08	0.53±0.08	0.55±0.08	0.63±0.07
Pima	0.00±0.01	0.00±0.01	0.00±0.01	0.10±0.12
Vehicle	0.17±0.03	0.17±0.03	0.17±0.03	0.20±0.09
Tictactoe	0.13±0.06	0.13±0.06	0.13±0.06	0.14±0.05
LED	0.48±0.04	0.48±0.04	0.50±0.05	0.52±0.05
Phoneme	0.72±0.02	0.72±0.02	0.72±0.02	0.72±0.02
Segmentation	0.70±0.05	0.70±0.05	0.70±0.05	0.70±0.04
Abalone	0.05±0.01	0.05±0.01	0.05±0.00	0.06±0.00
Waveform	0.21±0.04	0.21±0.04	0.21±0.04	0.22±0.08
PenDigits	0.55±0.01	0.55±0.01	0.55±0.01	0.62±0.02

method of initialization (K++R) with a supervised preprocessing step allow the standard K-means algorithm to reach a certain tradeoff between similarity (using MSE criterion) and prediction (that is the objective of the supervised clustering). Future works will be done: (i) to improve the method in the case where the number of cluster exceeds to the number of classes and therefore the case where K is an output, (ii) to

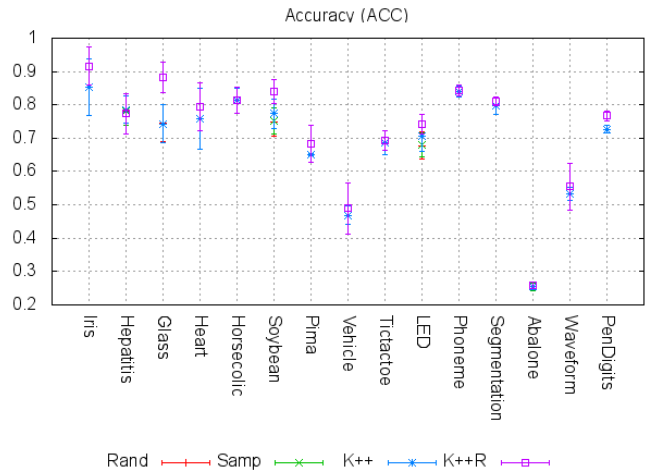


Fig. 6: ACC on the different databases (R=1000 for 'Rand', 'Sample' and 'K++', R=1 for K++R).

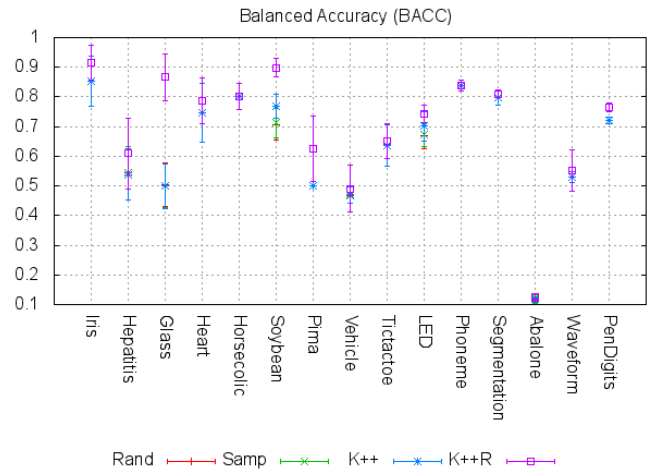


Fig. 7: BACC on the different databases (R=1000 for 'Rand', 'Sample' and 'K++', R=1 for K++R).

compare K++R to others unsupervised initialization methods, and (iii) to combine the K++R method with a supervised K -means algorithm.

REFERENCES

- [1] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley, 1990.
- [2] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, 1999.
- [3] S. H. Al-Harbi and V. J. Rayward-Smith, "Adapting k-means for supervised clustering," *Applied Intelligence*, vol. 24, no. 3, pp. 219–226, Jun. 2006.
- [4] C. F. Eick, N. Zeidat, and Z. Zhao, "Supervised clustering algorithms and benefits," in *In proceedings of*

the 16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI04), Boca, 2004, pp. 774–776.

- [5] J. MacQueen, “Some methods for classification and analysis of multivariate observations,” in *5th Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, 1967, pp. 281–297.
- [6] P. S. Bradley, O. L. Mangasarian, and W. N. Street, “Clustering via concave minimization,” in *Advances in Neural Information Processing Systems -9*. MIT Press, 1997, pp. 368–374.
- [7] M. Meilă and D. Heckerman, “An experimental comparison of several clustering and initialization methods,” in *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers Inc., 1998, pp. 386–395.
- [8] L. Hubert and P. Arabie, “Comparing partitions,” *Journal of Classification*, vol. 2, no. 1, pp. 193–218, 1985.
- [9] Z. Huang, “Extensions to the k-means algorithm for clustering large data sets with categorical values,” *Data Min. Knowl. Discov.*, vol. 2, pp. 283–304, 1998.
- [10] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley, 1990.
- [11] A. Juan and E. Vidal, “Comparison of Four Initialization Techniques for the k -medians Clustering Algorithm,” in *Proc. Joint IAPR International Workshops on Advances in Pattern Recognition*. London, UK: Springer-Verlag, 2000, pp. 842–852.
- [12] H.-S. Park and C.-H. Jun, “A simple and fast algorithm for k-medoids clustering,” *Expert Syst. Appl.*, vol. 36, no. 2, pp. 3336–3341, 2009.
- [13] M. E. Celebi, H. A. Kingravi, and P. A. Vela, “A comparative study of efficient initialization methods for the k-means clustering algorithm,” *Expert Syst. Appl.*, vol. 40, no. 1, pp. 200–210, 2013.
- [14] R. Maitra, A. D. Peterson, and A. P. Ghosh, “A systematic evaluation of different methods for initializing the k-means clustering algorithm,” *IEEE Transactions on Knowledge and Data Engineering*, pp. 522–537, 2010.
- [15] M. E. Celebi, “Improving the performance of k-means for color quantization,” *Image Vision Comput.*, vol. 29, no. 4, pp. 260–271, 2011.
- [16] E. Forgy, “Cluster analysis of multivariate data: Efficiency vs. interpretability of classification,” *Biometrics*, vol. 21, no. 78, 1965.
- [17] Y. Linde, A. Buzo, and R. Gray, “An Algorithm for Vector Quantizer Design,” *Communications, IEEE Transactions on*, vol. 28, no. 1, pp. 84–95, 1980.
- [18] I. Katsavounidis, C. C. Jay Kuo, and Z. Zhang, “A new initialization technique for generalized Lloyd iteration,” *IEEE Signal Processing Letters*, vol. 1, no. 10, pp. 144–146, 1994.
- [19] M. B. Al-Daoud and S. A. Roberts, “New methods for the initialisation of clusters,” *Pattern Recognition Letters*, vol. 17, no. 5, pp. 451 – 455, 1996.
- [20] D. Arthur and S. Vassilvitskii, “K-means++: The advantages of careful seeding,” in *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, ser. SODA ’07. USA: Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.
- [21] J. Sinkkonen, S. Kaski, and J. Nikkil, “Discriminative clustering: Optimal contingency tables by learning metrics,” in *ECML*, ser. Lecture Notes in Computer Science, vol. 2430. Springer, 2002, pp. 418–430.
- [22] T. Finley and T. Joachims, “Supervised clustering with support vector machines,” ser. ICML ’05. USA: ACM, 2005, pp. 217–224.
- [23] J. S. Aguilar-Ruiz, R. Ruiz, J. C. R. Santos, and R. Girddez, “Snn: A supervised clustering algorithm,” in *IEA/AIE*, ser. Lecture Notes in Computer Science, vol. 2070. Springer, 2001, pp. 207–216.
- [24] Y. Qu and S. Xu, “Supervised cluster analysis for microarray data based on multivariate gaussian mixture,” *Bioinformatics*, vol. 20, no. 12, pp. 1905–1913, 2004.
- [25] N. Slonim and N. Tishby, “Agglomerative information bottleneck,” in *Neural Information Processing Systems*, 2000, pp. 617–623.
- [26] N. Tishby, F. C. Pereira, and W. Bialek, “The information bottleneck method,” in *37-th Annual Allerton Conference on Communication, Control and Computing*, 1999, pp. 368–377.
- [27] P. Bungkomkhun and S. Auwatanamongkol, “Grid-based supervised clustering algorithm using greedy and gradient descent methods to build clusters,” *IJCSI International Journal of Computer Science*, 2012.
- [28] C. D. Manning, P. Raghavan, and H. Schtze, *Introduction to Information Retrieval*. New York: Cambridge University Press, 2008.
- [29] J. M. Santos and M. Embrechts, “On the use of the adjusted rand index as a metric for evaluating supervised classification,” ser. ICANN ’09. Springer-Verlag, 2009, pp. 175–184.
- [30] D. N. A. Asuncion, “UCI machine learning repository,” 2007. [Online]. Available: <http://www.ics.uci.edu/~sim5mlearn/{MLR}epository.html>
- [31] O. Alaoui Ismaili, V. Lemaire, and A. Cornuéjols, “Supervised pre-processings are useful for supervised clustering,” *Springer Series Studies in Classification, Data Analysis, and Knowledge Organization*, 2015.
- [32] M. Boullé, “MODL: a Bayes optimal discretization method for continuous attributes,” *Machine Learning*, vol. 65, no. 1, pp. 131–165, 2006.
- [33] —, “A Bayes optimal approach for partitioning the values of categorical attributes,” *Journal of Machine Learning Research*, vol. 6, pp. 1431–1452, 2005.