Active Learning for Semi-Supervised K-Means Clustering

Viet-Vu Vu, Nicolas Labroche, Bernadette Bouchon-Meunier

Active Learning for Semi-Supervised K-Means Clustering

Viet-Vu Vu, Nicolas Labroche, and Bernadette Bouchon-Meunier
Université Pierre et Marie Curie - Paris 6, CNRS UMR 7606, LIP6
4 Place Jussieu, 75005 Paris, France
{*viet-vu.vu, nicolas.labroche, bernadette.bouchon-meunier*}*@lip6.fr*

***Abstract*—K-Means algorithm is one of the most used clustering algorithm for Knowledge Discovery in Data Mining. Seed-based K-Means is the integration of a small set of labeled data (called seeds) to the K-Means algorithm to improve its performances and overcome its sensitivity to initial centers, that are, most of the time, generated at random or the authors assume that the seeds are available for each cluster. This paper introduces a new efficient algorithm for active seeds selection which relies on a Min-Max approach that favors the coverage of the whole dataset. Experiments conducted on artificial and real datasets show that, using our active seeds selection algorithm, our algorithm can collect the seeds such that, for each data set, each cluster has at least one seed after a very small number of queries, and using the collected seeds, the number of convergence iteration of K-Means clustering will be reduced, which is crucial in many KDD applications.**

***Keywords*-Semi-supervied clustering; active learning; seed;**

I. Introduction

Clustering is an important task in the process of Knowledge Discovery in Data Mining. In the past ten years, the problem of clustering with side information (known as Semi-Supervised Clustering) has become an active research direction to improve the quality of the results by integrating knowledge to the algorithms. Semi-Supervised clustering uses a small set of labeled data (called seeds) to guide the clustering process. Some works have been already proposed in the literature like seed-based K-Means [1], seed based Fuzzy-CMeans [5], and semi-supervised Density-Based Clustering (SSDBSCAN) [3].

The main problem when working with seeds is to determine how to choose the most useful for the algorithm. One of the solution to this problem is to adopt an active learning strategy which asks the users (called oracle or teacher in this case) to label potentially interesting data selected by the algorithm.

The objective of our work is to propose a new active learning algorithm adapted to semi-supervised clustering algorithms like Seed K-Means. To the best of our knowledge, this is the first paper dealing with active learning for seed-based clustering algorithms. Our method selects useful user queries according to a Min-Max approach to determine the set of labeled data. The underlying idea is that labeled data may not be available in real world applications while expert users can be solicited for some questions. Experiments conducted on artificial and real datasets show that, using our active seeds selection algorithm, our algorithm can collect the seeds such that, for each data set, each cluster has at least one seed after a very small number of queries, and using the collected seeds, the number of convergence iteration of K-Means clustering will be reduced, which is crucial in many KDD applications.

The rest of the paper is organized as follows: Section II discusses the related work. Section III presents our new framework for active constraint selection, while section IV describes the experiments that have been conducted on benchmark datasets. Finally, section V concludes and discusses future research.

II. K-Means Clustering by Seeding

The seed based K-Means algorithm has been proposed by Basu et al. [1]. This method uses a small set of labeled data, the seeds, to help the clustering of the unlabeled data. Two variants of semi-supervised K-Means clustering are introduced: Seed K-Means and Constraint K-Means. In both methods the seeds are supposed to be representative of all the clusters. In Seed K-Means, the labeled data are used to compute an initial center for each cluster. Then a traditional K-Means is applied on the dataset without any further use of the labeled data while in Constraint K-Means the information is used as constraints so that the labeled data can not be removed from the cluster they have been affected to by the user. The seed based K-Means is presented in the algorithms 1.

III. Proposed Algorithm

The novelty of our algorithm lies in the use of a Min-Max approach that selects a set of points that are far from each other, to collect a suitable set of seeds.

A. The Min-Max Approach

The idea of the Min-Max Approach (MMA) presented in [2] is to build a set of points Y from a dataset X such that the points in Y are far from each other and ensures a good coverage of the dataset. The main principles of MMA are described hereafter.

First a starting point y_1 is randomly chosen from the dataset X. Then, all the other points in Y are chosen among the points of X that maximize their minimal distance from the points already in Y. Thus, when t points already belong

Algorithm 1 Seed-KMeans

Input: *Data set* $X = \{x_i\}_{i=1}^{N}$, *number of clusters K, set* $S = \cup_{l=1}^{K} S_l$ *of initial seeds*

Output: *Disjoint K partitioning of* $\{X_l\}_{l=1}^{K}$ *such that KMeans objective funtion is optimized*

Process:

1: Initialize: $\mu_h^{(0)} \leftarrow \frac{1}{|S_h|} \sum_{x \in S_h} x$, for $h = 1, ..., K; t \leftarrow 0$
2: **repeat**
3: Assign_cluster: Assign each data point x to the cluster h^* (i.e. set $X_{h^*}^{(t+1)}$), for $h^* = argmin \|x - \mu_h^{(t)}\|^2$
4: estimate_means: $\mu_h^{(t+1)} \leftarrow \frac{1}{|X_h^{(t+1)}|} \sum_{x \in X_h^{(t+1)}} x$
5: $t \leftarrow (t+1)$
6: **until** (Convergence)

to Y, the process that selects the point y_{t+1} from X can be formalized as shown in equation 1:

$$y_{t+1} = argmax_{x \in X} \left(min_{y \in Y} d(x, y) \right) \quad (1)$$

where $d(.,.)$ denotes the distance defined in the space of the objects (for example an Euclidean distance, a Mahanalobis distance ...).

The underlying idea of the MMA in the context of active learning, is to select the point that is the farthest from the points that have already been used to formulate a query to the user. In other words, at each iteration, this method selects the point that exhibits the largest label uncertainty according to the previous answers of the user.

Figure 1 illustrates an example of MMA. It can be observed from this figure that MMA collects points that cover all the dataset X. For this reason, MMA has been used to initialize the k centers for the original K-Means in [7]. Contrary to this work, our objective is to use the MMA to develop an active learning system in which the points provided by MMA are used as candidates to generate queries to the user as explained in the following section.

B. New active learning algorithm

The active seed selection process is expressed as a loop, in which, at each iteration, the algorithm selects a point u from the $Candidate\_Set$ according to the Min-Max approach and formulate a query about this point to the user. As in all active learning system, we assume that the user can answer all the questions asked by the system. The loop stops when the user decides to or when all the points in the dataset have been explored (i.e. when the $Candidate\_Set$ is empty). The main steps of our algorithm for active seed selection are summed up in Algorithm 2 and Algorithm 3.

C. A variant of the proposed algorithm

As shown in Figure 1, although MMA allows a good coverage of the dataset, it does not ensure that the selected points are near the centers of each group while it could help K-Means like algorithm to converge faster. We propose hereafter a variant of our previous algorithm that initializes the points near the centers of the clusters, in the dense region of the dataset.

To this aim, we use a k-Nearest Neighbors Graph (k-

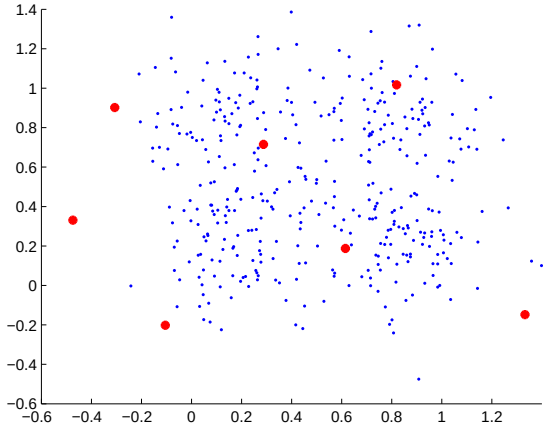

Figure 1. An example of illustration for 7 detected queries points (red points) by the Min-Max Strategy for the data set which consist 4 clusters of Gaussian distribution.

Algorithm 2 Active Seed Selection

Input: *Data set* $X = \{x_i\}_{i=1}^{N}$,

Output: *Set of seed* Y

Process:

1: $Y = \emptyset$
2: $Candidate\_Set = X$
3: Chose randomly a point r from $Candidate\_Set$
4: Query the user about the $Label$ of r ?
5: $Y = Y \cup \{Label(r)\}$
6: $Candidate\_Set = Candidate\_Set - \{r\}$
7: **repeat**
8: u = **Min_Max_Approach**$(Y, Candidate\_Set)$
9: **if** (exists candidate u) **then**
10: Query the user about the $Label$ of u ?
11: $Y = Y \cup \{Label(u)\}$
12: $Candidate\_Set = Candidate\_Set - \{u\}$
13: **end if**
14: **until** (*User_Stop = True*) **or** $(Candidate\_Set = \emptyset)$

Algorithm 3 Min_Max_Approach$(Y, Candidat_Set)$

Process:

1: **for all** $v_i \in Candidat\_Set$ **do**
2: $q_i = min\{distance(v_i, y_j) : \forall y_j \in Y\}$
3: **end for**
4: $u = u_k \in Candidate\_Set : k = max_k\{q_k\}$
5: Return u;

NNG), from which it is possible to estimate a density degree for each point in the dataset by using the notion of *Local Density Score* proposed by Le et al. in [6].

The k-NNG is defined as a weighted undirected graph, in which each vertex represents a data point, and that possesses at most k edges to its k-nearest neighbors. An edge is created between a pair of points, x_i and x_j, if and only if x_i and x_j have each other in their k-nearest neighbors set. The weight $\omega(x_i, x_j)$ of the edge (the similarity) between two points x_i and x_j is defined as the number of common nearest neighbors the two points share as shown in equation 2 [8].

$$\omega(x_i, x_j) = \mid NN(x_i) \cap NN(x_j) \mid \tag{2}$$

where *NN(.)* denotes the set of k-nearest neighbors of the specified point. The important property of this similarity measure is its own *built-in automatic scaling*, which makes it adapted to treat datasets with distinct cluster densities.

The notion of Local Density Score (*LDS*) of a vertex $x_i \in$ k-NNG is defined by equation 3 [6]:

$$LDS(x_i) = \frac{\sum_{q \in NN(x_i)} \omega(x_i, q)}{k} \tag{3}$$

The *LDS* of a point in $[0, k-1]$ is the average distance to its k-nearest neighbors. *LDS* is defined in such a way that a high value indicates a strong association between the point x_i and its neighbors, i.e. x_i belongs to a dense region. In contrast, a low value of *LDS* means that x_i belongs to a sparse region or transition region between clusters.

In our approach, where we are only interested in points that are in dense regions of the dataset, we keep the points that have a *LDS* score higher than a fixed threshold ϵ. In algorithm 2, the $Candidat\_Set$ will be chosen as shown in equation 4:

$$Candidate\_Set = \{p \in X : LDS(p) \geq \epsilon\} \tag{4}$$

Figure 2 illustrates the $Candidate\_Set$ obtained on the same dataset used in the Figure 1. It can be seen that, as expected, the candidates are selected in the dense region of the dataset, i.e. near the center of the clusters.

Finally, our *LDS*-variant uses two parameters: the number of nearest neighbors k and the threshold ϵ. Preliminary experiments conducted on the datasets introduced in the next section show that the value of k cannot be generalized for all datasets because it depends on the structure and the size of the datasets. However, we have observed experimentally that, for the datasets introduced in the next section, the value of ϵ can be set in the interval $[\frac{k}{2}-2, \frac{k}{2}+2]$.

D. Complexity of our new methods

The complexity of our algorithm (algorithm 2 and algorithm 3) is O((number of point in Y)^2 $\times$ number of point in the $Candidate\_Set$) = O((number of queries)$^2 \times n$). In fact, the number of queries is very small.

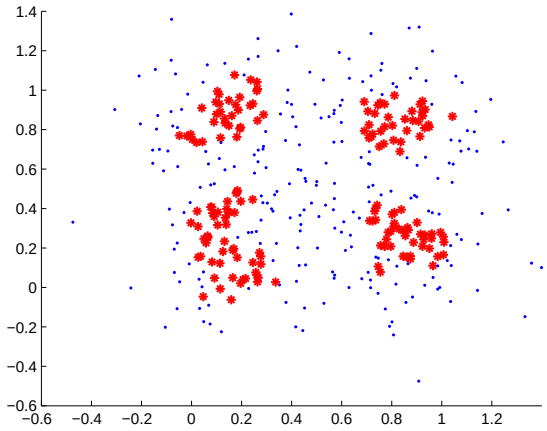

Figure 2. An example of visualization LDS, the $Candidate\_Set$ consists the red points for the data set which consists 4 clusters of Gaussian distribution. In this example, we use $k = 30$ and the threshold $\epsilon = 17$

The complexity of our variant algorithm is $O(n)$, $O(n \times logn)$ or $O(n^2)$ when the dimension of data is respectively low, medium or extremely high [9].

IV. Experiments and Results

A. Experimental Protocol

We use 5 real datasets from the Machine Learning Repository [11] named: Iris, Soybean, Thyroid, Harbermam, and Pima, and use 3 artificial data sets named Art_1, Art_2, and Art_3 [10], that have been generated according to Gaussian or uniform laws with distinct difficulties. The detail of the datasets can be found in Table I.

We first compares our new active learning approach based on Min-Max to a random selection of seeds each time with two distinct semi-supervised clustering algorithms. The second experiment concerns our variant of the Min-Max algorithm coupled with the *LDS* selection of the candidates. As this method may only be applied on low-dimensional datasets because of complexity issues, we only report here results that illustrates its efficiency in term of speed convergence for some selected datasets compared to the MMA based active learning algorithm.

B. Results

From tables II, it can be seen that with the same number of queries, our approach can collected the different labels better. And after a small of queries, our approach can collect at least one seed for each cluster. It can be explained by the fact that using the Min-Max approach, the selected points will cover whole the data set and so that it appears in a good position for labeling.

Table I
MAIN CHARACTERISTICS OF THE REAL DATASETS

ID	Data	#Objects	#Attributes	#Clusters
1	Iris	150	4	3
2	Soybean	47	35	4
3	Thyroid	215	5	3
4	Haberman	306	3	2
5	Pima	768	8	2
6	Art1	400	2	4
7	Art2	1000	2	2
8	Art3	900	2	9

Table II
MEANS OF COLLECTED NUMBER LABELS FOR CLUSTERS OF OUR APPROACH AND RANDOM APPROACH FOR SEED K-MEANS, σ_1 AND σ_2 ARE STANDARD DEVIATION.

Iris	#Queries	3	4	5	6
	Proposed[σ_1]	**2.6**[0.50]	**2.97**[0.20]	**3.0**[0.0]	**3.0**[0.0]
	Random[σ_2]	2.14[0.5]	2.37[0.54]	2.51[0.57]	2.72[0.45]
Soy-bean	#Queries	5	6	7	8
	Proposed[σ_1]	**3.75**[0.43]	**3.99**[0.09]	**4.0**[0.0]	**4.0**[0.0]
	Random[σ_2]	3.06[0.62]	3.32[0.59]	3.49[0.553]	3.64[0.52]
Thy-roid	#Queries	16	17	18	19
	Proposed[σ_1]	**2.99**[0.1]	**2.99**[0.1]	**2.99**[0.1]	**3.0**[0.0]
	Random[σ_2]	2.85[0.36]	2.87[0.33]	2.87[0.33]	2.87[0.35]
Har-berman	#Queries	2	3	4	5
	Proposed[σ_1]	**1.71**[0.45]	**1.79**[0.41]	**1.99**[0.10]	**2.0**[0.0]
	Random[σ_2]	1.38[0.48]	1.54[0.50]	1.62[0.49]	1.74[0.44]
Pima	#Queries	4	5	6	7
	Proposed[σ_1]	**1.97**[0.18]	**1.98**[0.14]	**2.0**[0.0]	**2.0**[0.0]
	Random[σ_2]	1.84[0.37]	1.84[0.35]	1.92[0.27]	1.92[0.27]
Art1	#Queries	5	6	7	8
	Proposed[σ_1]	**3.96**[0.19]	**4.0**[0.0]	**4.0**[0.0]	**4.0**[0.0]
	Random[σ_2]	3.02[0.67]	3.34[0.62]	3.46[0.57]	3.76[0.42]
Art2	#Queries	2	3	4	5
	Proposed[σ_1]	**1.96**[0.19]	**1.98**[0.130]	**2.0**[0.0]	**2.0**[0.0]
	Random[σ_2]	1.38[0.48]	1.8[0.4]	1.84[0.36]	1.92[0.27]
Art3	#Queries	27	28	29	30
	Proposed[σ_1]	**8.94**[0.23]	**8.96**[0.19]	**8.90**[0.29]	**9.0**[0.0]
	Random[σ_2]	8.66[0.51]	8.56[0.60]	8.72[0.44]	8.74[0.43]

Finally, we compared our MMA approach to our variant based on LDS on some datasets as reported in the table III. For this experiment, we use the Seed K-Means to evaluate the efficiency of our variant approach. Table III shows the number of iterations to convergence for both approach. As before, the results have been computed over 1000 runs and so table III presents the mean and its associated standard deviation. For all 7 low-dimensional datasets used in this experiment, the *LDS*-variant obtains better results for the evaluation criterion. It is certainly due to the fact that the variant introduce a filtering step that help focusing on the most interesting points to formulate the queries to the user. This experiment shows that the *LDS*-variant can be more efficient than our new MMA algorithm, but is limited to low-dimensional data.

Table III
COMPARE THE SPEED OF CONVERGENCE BETWEEN OUR APPROACH AND ITS VARIANT. ITER.1 AND ITER.2 ARE THE AVERAGE NUMBER OF ITERATION OF OUR APPROACH AND ITS VARIANT.

	Iris	Thyroid	Haberman	Pima	Art1	Art2
Iter.1	7.3[2.5]	12.7[8.1]	12.3[3.8]	17.3[2.6]	8.45[2.9]	6.04[0.7]
Iter.2	**4.2**[1.1]	**8.01**[4.8]	**8.42**[2.2]	**15.1**[2.5]	**7.32**[1.9]	**4.32**[0.3]

V. CONCLUSION

A framework with two new active learning algorithms is proposed for semi-supervised clustering algorithms like Seed Based K-Means and Constraint K-Means. Naturally, our algorithm can be easily adapted to the seed base Fuzzy C-Means algorithm. Our main approach is based on the Min-Max strategy that allows to select a set of points that cover all the dataset and thus allows to formulate user queries where the cluster label information is the most uncertain considering the previous answer of the user. We also propose a variant of our main algorithm, based on Local Density Score to filter the candidates points before applying the Min-Max strategy. This method is more efficient but is limited to low-dimensional dataset because of its time complexity. Experiments conducted on artificial and real datasets show that, using our active seeds selection algorithm, our algorithm can collect the seeds such that, for each data set, each cluster has at least one seed after a very small number of queries, and using the collected seeds, the number of convergence iteration of K-Means clustering will be reduced, which is crucial in many KDD applications.

REFERENCES

[1] S. Basu, A. Banerjee, and R. J. Mooney. *Semi-supervised Clustering by Seeding*. In Proc. 19th ICML, 2002.

[2] P.K. Mallapragada, R. Jin, and A.K. Jain. *Active query selection for semi-supervised clustering*. In Proc. ICPR, 2008.

[3] L. Lelis and J. Sander. *Semi-supervised Density-Based Clustering*. In Proc. 9th IEEE ICDM, 2009.

[4] A.K. Jain. *Data clustering: 50 years beyond K-means*. Pattern Recognition Letters, 2009.

[5] A. M. Bensaid, L.O. Hall, J.C. Bezdek, and L.P. Clarke. *Partially Supervised clustering for image segmentation*. Pattern Recognition, Vol. 29, No.5, 1996.

[6] D.-D. Le, and S. Satoh. *Unsupervised Face Annotation by Mining the Web*. In Proc. 9th IEEE ICDM, 2008

[7] M.A. Hasan, V. Chaoji, S.Salem, M.J. Zaki. *Robust partitional clustering by outlier and density insensitive seeding*. Pattern Recognition Letters 30(11): 994-1002, 2009.

[8] R.A. Jarvis, and E.A. Patrick. *Clustering using a similarity measure based on shared near neighbors*. In IEEE Transactions on Computer, number 11, 1973.

[9] M. Breunig, H.-P. Kriegel, R.T. Ng, and J. Sander. *LOF: Identifying Density-Based Local Outliers*. In Proc. 19th ACM SIGMOD, 2000.

[10] N. Labroche, N. Monmarch, and G. Venturini. *A new clustering algorithm based on the chemical recognition system of ants*. In Proc. ECAI, 2002.

[11] C.L. Blake, and C.J. Merz. *UCI machine learning repository*, 1998.