ISSN 2320 – 2602
Volume 4 No.4, April 2015
International Journal of Advances in Computer Science and Technology
Available Online at http://www.warse.org/ijacst/static/pdf/file/ijacst04442015.pdf

An efficient method for active semi-supervised density based clustering

Viet-Vu Vu
Electronics Faculty, Thai Nguyen University of Technology, Thai Nguyen city, Viet Nam, vuvietvu@tnut.edu.vn

ABSTRACT

Semi-supervised clustering algorithms relies on side information, either labeled data (seeds) or pairwise constraints (must-link or cannot link) between data objects, to improve the quality of clustering. This paper proposes to extend an existing seed-based clustering algorithm with an active learning mechanism to collect pairwise constraints. My new semi-supervised algorithm can deal with both seeds and constraints. Experiment results on real data sets show the efficient of my algorithm when compared to the initial seed-based clustering algorithm.

Key words: semi-supervised clustering, active learning, seed, constraint.

1. 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 unsupervised algorithms [2].

The works on semi-supervised clustering can be divided into two main families depending on the type of side information provided to the algorithm. On the one hand, seed based clustering [3, 4, 6, 12] relies on a small set of labeled data, while on the other hand, constraint based clustering relies on a small set of pairwise constraints (must-link - ML or cannot link – CL) between data objects [2].

Each of these methods has advantages and drawbacks: seeds are useful for initialization of clusters but can be more difficult to set, while constraints are more adapted to delimit the frontier between clusters but needs clusters to already exist to be efficient. In both case, the difficulty of the semi-supervised methods, as in supervised learning, is to initiate the algorithms with labeled data or pairwise constraints that are likely to be beneficial for the clustering algorithm. This problem has been tackled in [5, 8, 9, 10] where the authors propose an active learning algorithm to: (1) select the best constraints/seed based on a nearest-neighbors density criterion and, (2) propagates the constraints selected by the expert to infer new constraints automatically and thus minimizing the number of expert solicitations.

Figure 1 and figure 2 illustrate different types of prior knowledge that can be included in the process of classifying data: dots correspond to points without any labels; points with labels are denoted by circles, asterisks and crosses. In figure 2 (left), the must-link and cannot-link constraints are denoted by
solid and dashed lines [1].

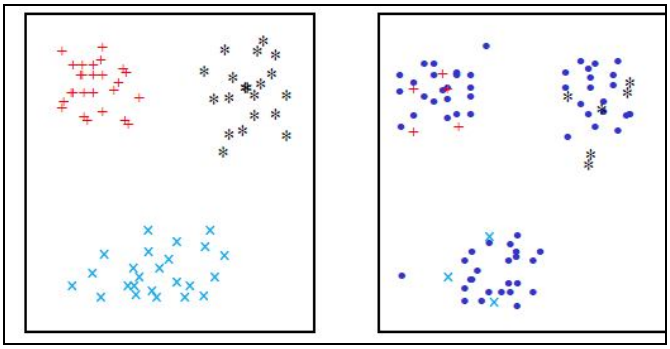


Figure 1: Spectrum of supervised (left) and partially labeled (right) Learning

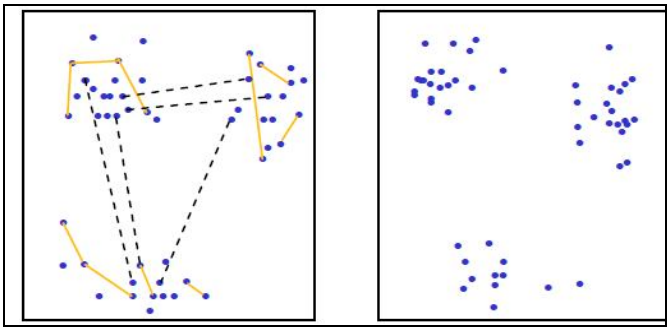


Figure 2: Spectrum of constrained (left) and unsupervised (right) learning

In this paper, I extend the Seed based DBSCAN algorithm (SSDBSCAN) [4] and propose the ActSSDBSCAN algorithm that integrates an active learning strategy to collect ML and CL constraints. Thus, the proposed algorithm is probably, to the best of my knowledge, the first method that includes at the same time seeds and constraints. Preliminary experiments conducted on some real datasets show that, using my new active algorithm, the performance of SSDBSCAN can be improved after only few expert solicitations.

This paper is organized as follows: Section 2 presents the main principles of the seed-based DBSCAN on which relies my new Active SSDBSCAN algorithm described in Section 3.

Then, section 4 presents the experimental protocol and the preliminary results. Finally, Section 5 concludes and devises some perspectives of this research.

2. SEED-BASED CLUSTERING ALGORITHMS

This section introduces some existed words with sees-based clustering like Seed-based DBSCAN and Seed-based K-means.

2.1. Seed-based DBSCAN

The seed-based DBSCAN extends the original DBSCAN algorithm [14] with a small set of labeled data to enable the discovery of clusters with distinct densities.

Indeed, following [15], in the algorithm DBSCAN, the notion of density is formalized according to two parameters: *MinPts* specifies a minimum number of objects, and ε is the radius of a hypers-phere in the space of the objects. However, as these parameters are set once for all clusters, DBSCAN can only detect clusters with the same density.

The objective of SSDBSCAN is to overcome this limit by using seeds to compute an adapted radius ε for each cluster. Thus, SSDBSCAN has only one parameter *MinPts*, the ε parameter being deduced from the set of provided seeds. Another difference is that, contrary to DBSCAN, in SSDBSCAN, the clustering is seen more like a graph partitioning problem.

To this aim, the data set is represented as a weighted undirected graph where each vertex corresponds to an unique data objects and each edge between objects p and q has a weight determined by the *rDist()* measure described hereafter.

The *rDist(*p,q*)* measure indicates the smallest radius value for which *p* and *q* are core points and directly density connected with respect to *MinPts*. Thus, *rDist()* can be formalized as follows:

$\forall p, q \in X$, $rDist(p,q) = max(cDist(p),cDist(q), d(p,q))$ (1)

where *D* denotes the data set, *d()* is the metric used in the clustering and $\forall$ o ∈ D, *cDist(o)* is the minimal radius such that *o* is a core-point and has *MinPts* nearest-neighbors.

Then, given a set of seeds D_L, the SSDBSCAN algorithm proceeds as follows. Using the previous distance *rDist()*, it is possible to construct a density-based cluster *C* that contains the first seed point p, by first adding p to C and then iteratively adding the next closest point in term of *rDist()* distance to C. The process continues until there is a point q that has a different label from p. At that time, the algorithm backtracks to the point o with the largest *rDist()* before adding q. The current expansion stops and includes all points up to but excluding o, having a cluster C containing p. Conceptually, this is the same as constructing a minimum spanning tree (MST) for a complete graph where the set of vertices equals *D* and the edge weights are given by *rDist()*.

2.2. Seed-based K-Means

The seed based K-Means algorithm has been proposed by Basu et al. [4]. 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 cannot be removed from the cluster they have been affected to by the user. The seed based K-Means is presented in the algorithms \ref{alg1}.

Algorith 1: Seed-based KMeans

Input: Set of data points $X = \{x_1,x_2,\ldots,x_n\}$; $x_i \in R^d$, number of clusters K, $\{S_1, S_2,\ldots,S_K\}$ of initial seeds

Output: Disjoint K partitioning $\{X_1, X_2,\ldots,X_K\}$ of X such that K-Means objective function is optimized.

Method:

Step 1. $g_h = \sum_{x \in Sh} x / |S_h|$, for h = 1,…,K; t = 0

Step 2: Repeat until *convergence*

- Assign_cluster: Assign each data point x to the cluster h*, for $h^* = \text{argmin}||x-g_h^{(t)}||^2$
- Estimate_mean: $g_h^{(t+1)} = \sum_{x \in Sh} x/|X_h^{(t+1)}|$
- t = t+1

3. ACTIVE LEARNING SEED-BASED DBSCAN

In the context of density based clustering, distant points are not necessarily in different clusters and the choice of the largest edge in the set of all density-connection paths to decide of the separation between two clusters *may not be the best solution*.

My proposal is to use the expert knowledge to define the appropriate separation distance between clusters. To this aim, my algorithm integrates an active learning mechanism to gather constraints and can be summarized as follows:

Algorithm 2: ActSSDBSCAN

Input: Set of data points $X = \{x_1,x_2,\ldots,x_n\}$; $x_i \in R^d$, $\{S_1, S_2,\ldots,S_K\}$ is the set of seeds

Output: Disjoint K partitioning $\{X_1, X_2,\ldots,X_K\}$

Repeat

Step 1: Build cluster as in SSDBSCAN**,** if the *stop condition* is true, go to step 3

Step 2: For each sorted edge, ask the expert if the relation between the vertices is a must-link (*ML*) or cannot-link (*CL*) constraint;

Step 3: While the expert answer is *ML* go to step 2;

Step 4: If the expert answer is *CL* then choose the edge *rDist* value as a separation distance and obtain a cluster.

Until *the set of seeds is empty*

4. EXPERIMENT RESULTS

I use 5 real datasets from the Machine Learning Repository [15] named: Protein, Iris, Glass, Thyroid, and LetterIJL to evaluate my algorithm. The detail of datasets is shown in Table 1.

Table 1. Data set for testing

ID	Name	N	M	k
1	Protein	115	20	6
2	Iris	150	4	3
3	Glass	214	9	6
4	Thyroid	215	5	3
5	LetterIJL	227	16	3

I use the Rand Index (RI) measure [13], as it is widely used in evaluation of clustering results.

The RI measure computes the agreement between the theoretical partition of each dataset and the output partition of evaluated algorithms.

This measure is based on n(n-1)/2 pairwise comparisons between the n points of a data set X. For each pair of point x_i and x_j in X, a partition assigns them either to the same cluster or to different clusters.

Let us consider two partitions P_1 and P_2, and let a be the number of decisions where the point x_i is in the same cluster as x_j in P_1 and P_2. Let b be the number of decisions where the two points are placed in different clusters in both partitions. A total agreement can then be calculated as shown in equation (1).

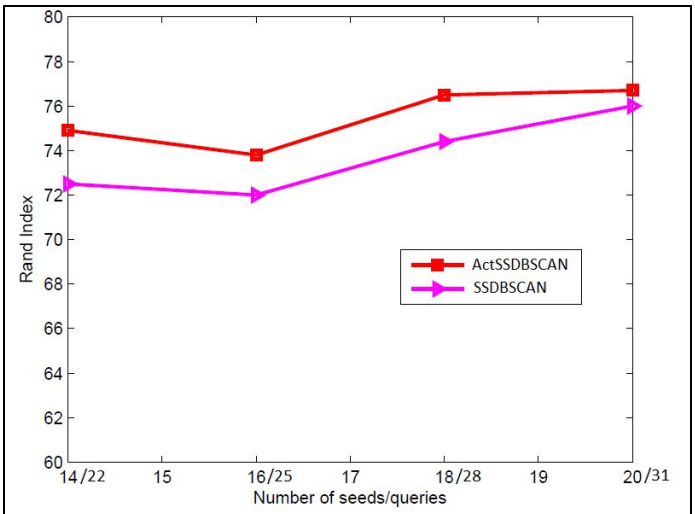
$$RI(P_1, P_2) = \frac{2(a+b)}{n(n-1)} \qquad (1)$$

RI takes values between 0 and 1; RI = 1 when the result is the same as the ground-truth. The larger the RI, the better the result.

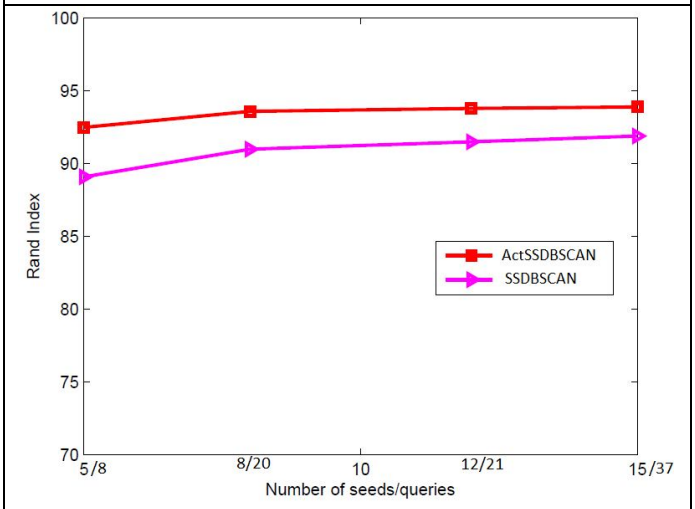
It can be seen from figure 3 that ActSSDBSCAN outperforms SSDBSCAN for each of the benchmark data sets. This experiments show the benefit of using both seeds and constraints to build the clusters and validate my hypothesis that the longest edge may not be the best criterion in the case of density based clustering algorithms.

5. CONCLUSION

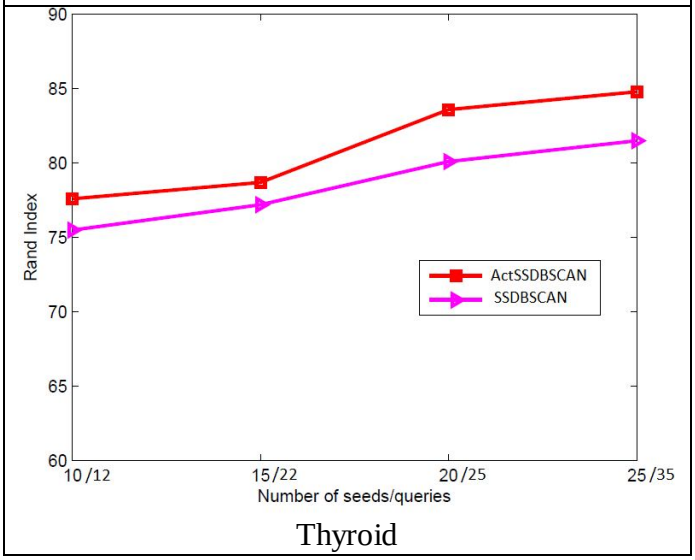
This paper presents a new active learning density based clustering algorithm named ActSSDBSCAN. To the best of my knowledge, this is the first semi-supervised algorithm to use both seeds and constraints as side information. Preliminary results on real data sets show the benefit of my approach when compared to SSDBSCAN. Future research should help minimizing expert solicitations during the active learning step.



Protein



Iris



Thyroid

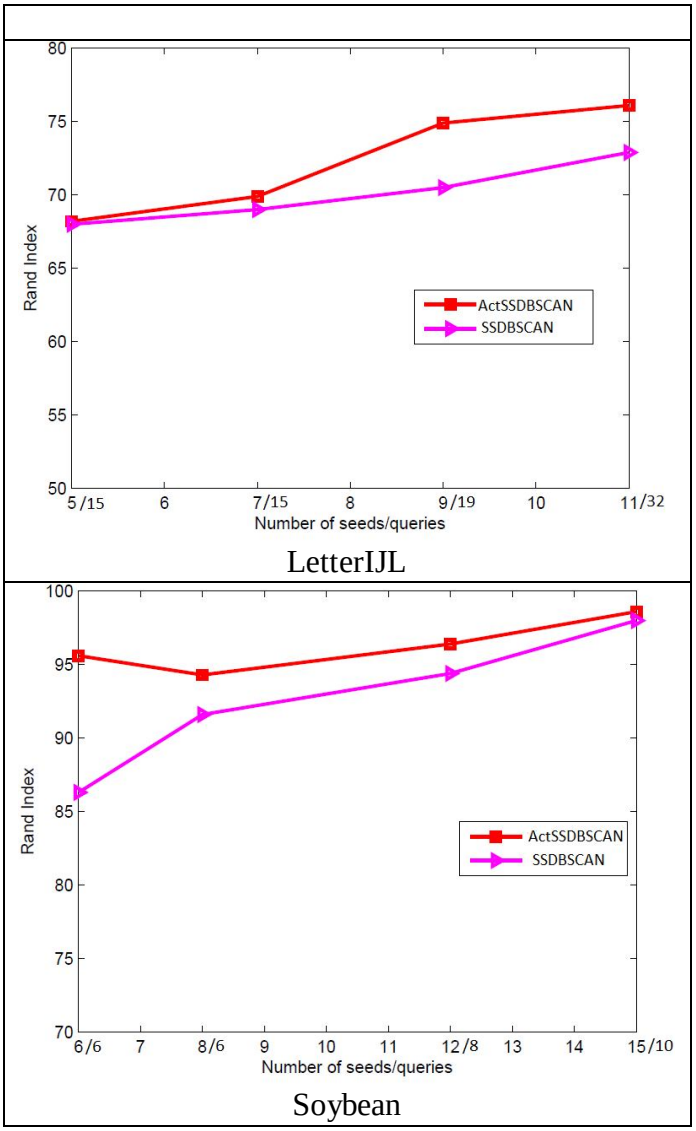


Figure 3: Experiment results

REFERENCES

1. T. Lange, M.H. Law, A.K. Jain and J.B. Buhmann. **Learning with constrained and unlabeled data**, in proc. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pp. 730-735, 2005.
2. S. Basu, I. Davidson, and K. L. Wagstaff, **Constrained Clustering: Advances in Algorithms, Theory, and Applications,** Chapman and Hall/CRC Data Mining and Knowledge Discovery Series, 1st ed., 2008.
3. Levi Lelis, Jörg Sander. **Semi-supervised Density-Based Clustering**. In Proc. IEEE International Conference on Data mining, 2009, pp. 842-847
4. Sugato Basu, Arindam Banerjee, Raymond J. Mooney: **Semi-supervised Clustering by Seeding**. In Proc. ICML 2002: 27-34
5. Viet-Vu Vu, Nicolas Labroche, and Bernadette Bouchon-Meunier, **Improving Constrained Clustering with Active Query Selection**, Pattern Recognition 45(4): 1749-1758 (2012), ISSN: 0031-3203.
6. Carlos Ruiz, Myra Spiliopoulou, Ernestina Menasalvas Ruiz: **Density-based semi-supervised clustering**. Data Min. Knowl. Discov. 21(3): 345-370 (2010)
7. Anil K. Jain.: **Data clustering: 50 years beyond K-means**. Pattern Recognition Letters (PRL) 31(8):651-666 (2010).
8. Viet-Vu Vu, Nicolas Labroche, and Bernadette Bouchon-Meunier. **Active Learning for Semi-Supervised K-Means Clustering**. In Proceedings of the 22nd IEEE International Conference on Tools with Artificial Intelligence (ICTAI-2010), Arras, France, October, 2010.
9. Viet-Vu Vu, Nicolas Labroche, and Bernadette Bouchon-Meunier. **Boosting Clustering by Active Constraint Selection**. In Proceedings of the 19th European Conference on Artificial Intelligence (ECAI-2010), Lisbon, Portugal, August, 2010.
10. Viet-Vu Vu, Nicolas Labroche, and Bernadette Bouchon-Meunier. **An Efficient Active Constraint Selection Algorithm for Clustering**. In Proceedings of the 20th IEEE International Conference on Pattern Recognition (ICPR-2010), Istanbul, Turkey, August, 2010.
11. Kiri Wagstaff, Claire Cardie, Seth Rogers, Stefan Schrödl: **Constrained K-means Clustering with Background Knowledge**. In Proc. ICML 2001: 577-584
12. Amine Bensaid, Lawrence O. Hall, James C. Bezdek, Laurence P. Clarke. **Partially supervised clustering for image segmentation**. Pattern Recognition 29(5): 859-871 (1996)
13. http://archive.ics.uci.edu/ml/
14. Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu. **A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise**. In Proc. KDD 1996: 226-231.
15. Christian Böhm and Claudia Plant: **HISSCLU: a hierarchical density-based method for semi-supervised clustering**. In Proc. EDBT, 2008: 440-451.