Middle-East Journal of Scientific Research 25 (1): 196-202, 2017
ISSN 1990-9233

DOI: 10.5829/idosi.mejsr.2017.196.202

# Cluster Ensemble Approach for Semi Supervised Clustering

*[1]M. Pavithra and [2]R.M.S. Parvathi*

[1]Assistant Professor, Department of C.S.E, Jansons Institute of Technology, Coimbatore, India
[2]Dean-PG Studies, Sri Ramakrishna Institute of Technology, Coimbatore, India

**Abstract:** The semi-supervised clustering are important extensions of the standard clustering paradigm. Consensus clustering (also known as aggregation of clustering) can improve clustering robustness, deal with distributed and heterogeneous data sources and make use of multiple clustering criteria. Semi-supervised clustering can integrate various forms of background knowledge into clustering. In this paper, we show how consensus and semi-supervised clustering can be formulated within the framework of nonnegative matrix factorization (NMF). We show that this framework yields NMF-based algorithms that are: (1) extremely simple to implement; (2) provably correct and provably convergent. We conduct a wide range of comparative experiments that demonstrate the effectiveness of this NMF-based approach.Clustering ensemble is one of the most recent advances in unsupervised learning. It aims to combine the clustering results obtained using different algorithms or from different runs of the same clustering algorithm for the same data set, this is accomplished using on a consensus function, the efficiency and accuracy of this method has been proven in many works in literature. The semi-supervised clustering algorithms used to generate the base partitions; the second parameter is used to provide the user feedback on these partitions. The two parameters are engaged in a "relabeling and voting" based consensus function to produce the final clustering.

**Key words:** Clustering · Semi-supervised clustering · Consensus clustering · Cluster ensemble · Consensus function

## INTRODUCTION

Semi-supervised clustering refers to the situation in which constraints are imposed on pairs of datapoints; in particular, there may be "must-linkconstraints" (two data points must be clustered into the same cluster) and "cannot-link constraints" (two data points can not be clustered in to the same cluster) [1]. Semi-supervised clustering is another hot topic in machine learning, in which both labelled and unlabelled data are used for training - typically a small amount of labelled data with a large amount of unlabelled data. Semi-supervised learning falls between supervised learning (with completely labelled training data) and unsupervised learning (without any labelled training data) [2]. It has been proven by many machinelearning researchers that unlabelled data, when used in conjunction with a small amount of labelled data, can produce considerable improvement in learning accuracy. Generating labelled data for a learning problem often requires a human effort (supervision) to manually classify training examples [3]. The cost producing a fully labelled training set in the labelling process may be infeasible in many cases, whereas producing unlabelled data is relatively inexpensive. In such situations, semi-supervised learning can be of great practical value [4].

Based on the above viewpoints, semi-supervised clustering research can be roughly divided into three directions: Based on the constraint mechanism, based on the distance and hybrid. Related research at present basically belong to the three class, which based on the pairwise constraints algorithm include: reference [5] is based on density clustering algorithm, can deal with any shapes of clusters and based on the constraint set to split or merge clusters; reference [6] presented an effective semi-supervised clustering algorithm and introduced fuzzy constraint thought, with minimal supervision information clustering; reference [7] puts forward a kind of distinguishing nonlinear transformation metrics in measurement and based on image retrieval to test, its effect is good.

**Correspondig Author:** M. Pavithra, Assistant Professor, Department of C.S.E.,
Jansons Institute of Technology, Coimbatore, India.

We have several contributions. First, we propose an efficientsemi-supervised clustering algorithm based on clusterim purity measure. Second, we apply our technique to classify evolving data streams [8]. To our knowledge, there are no stream data classification algorithms that apply semisupervised clustering. Third, we provide a solution to the more practical situation of stream classification when labeled data are scarce [9]. We show that our approach can achieve better classification accuracy than other stream classification approaches, utilizing only a fraction (e.g. 5%) of the labeled instances used in those approaches. Finally, we apply our technique to detect botnet traffic and obtain 98% classification accuracy on average. We believe that the proposed method provides a promising, powerful and practical technique to the stream classification problem in general [10].

**Related Work:** Our work is related to both stream classification and semi-supervised clustering techniques. We briefiy discuss both of them. Semi-supervised clustering techniques utilize a small amount of knowledge available in the form of pairwise constraints (must-link, cannot-link), or class labels of the data points. Recent approaches for semi-supervised clustering incorporated pairwise constraints on top of the unsupervised K-means clustering algorithm and formulated a constraint-based K-means clustering problem [11], which was solved with an Expectation-Maximization (EM) framework. Our approach is different from these approaches because rather than using pair-wise constraints, we utilize a cluster-impurity measure based on the limited labelled data contained in each cluster. If pair-wise constraints are used, then the running time per E-M step is quadratic in total number of labelled points, whereastherunning time is linear if impurity measures are used. So, the impurity measures are more realistic in classifying a highspeed stream data [12].

Our approach is also an ensemble approach, but it is different from other ensemble approaches in two aspects. First, previous ensemble-based techniques use the underlying learning algorithm (such as decision tree, Naive Bayes, etc.) as a black-box and concentrate only on building an efficient ensemble. But we concentrate on the learning algorithm itself and try to construct efficient classification models in an evolving scenario [1].

Semi-supervised clustering techniques utilize a small amount of knowledge available in the form of pairwise constraints (must-link, cannot-link), or class labels of the data points. Recent approaches for semi-supervised clustering incorporated pairwise constraints on top of the unsupervised K-means clustering algorithm and formulated a constraint-based K-means clustering problem [2], which was solved with an Expectation-Maximization (EM) framework [3]. Our approach is different from these approaches because rather than using pair-wise constraints, we utilize a cluster-impurity measure based on the limited labeled data contained in each cluster. If pair-wise constraints are used, then the running time per E-M step is quadratic in total number of labeled points, where as the running time is linear if impurity measures are used. So, the impurity measures are more realistic in classifying a highspeed stream data. There have been many works in stream data classification. The reare two main approaches-single model classification and ensemble classification [4].

Single model classification techniques incrementally update their model with new data to cope with the evolution of the stream [5]. These techniques usually require complex operations to modify the internal structure of the model and may perform poorly if there is concept-drift in the stream. To solve these problems, several ensemble techniques for stream data mining have been proposed [6]. These ensemble approaches have the advantage that they can be more efficiently built than updating a single model and they observe higher accuracy than their single model counterpart [7]. Our approach is also an ensemble approach, but it is different from other ensemble approaches in two aspects. First, previous ensemble-based techniques use the underlying learning algorithm (such as decision tree, Naive Bayes, etc.) as a black-box and concentrate only on building an efficient ensemble. But we concentrate on the learning algorithm itself and try to construct efficient classification models in an evolving scenario. In this light, our work is more closely related with the work of Aggarwal *et al*. [8]. Secondly, previous techniques (including [9] require completely labeled training data. But in practice, a very limited amount of labeled data may be available in the stream, leading to poorly trained classification models. Soourapproach is more realistic in a stream environment [10].

**Clustering Ensemble:** The objective of clustering Ensemble is to integrate clustering partitions obtained using various techniques. Clustering ensemble algorithms are usually divided into two stages [1]. At the first stage different partitions of same dataset are produced using independent runs of various clustering algorithms or the same clustering algorithms. Then, in the next stage, a consensus function is used to find a final partition from the partitions generated in the first stage. Figure 1 shows clustering ensembles procedure.

There are various approaches of consensus function: Mixture Models - a probabilistic model of consensus using a finite mixture of multinomial distributions in a space of clustering's, Voting Based Approach - solving explicitly the correspondence problem between the labels of known and derived clusters [2] and determining the final consensus function by assigning objects in clusters by using a simple voting procedure, Information Theory Approach – considering the Mutual Information between the probability distribution of labels of the consensus partition and the labels of the ensemble as the objective function for ensemble, Co-Association Based Approach - considering the number of clusters shared by two objects in all the partition of an ensemble as the measurement of similarity between those two objects,

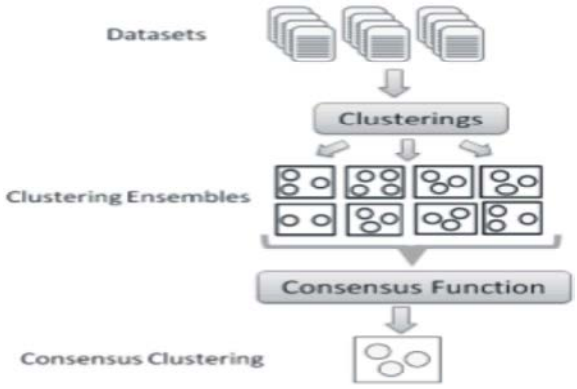


Fig. 1: Overview of clustering ensemble procedure

**Proposed Work**

**Semi-Supervised Clustering Ensemble by Voting (Scev Algorithm):** As mentioned in the introduction the aim of this paper is to combine the benefit of clustering ensemble and semi supervised clustering algorithms to improve clustering accuracy for the results, this is accomplished by engaging supervision in the clustering ensemble procedure in two places: either in the clustering ensemble generation step or in the form of a feedback on the generated partition that can be used in the consensus function stage [3].

The second parameter is used to carry feedback about the first stage (base partitions) to the second stage (consensus function). Feedback can provide indicators about the good clustering that the user desire, this may include the part information that is not addressed by supervision. If the feedback and the supervision are consistent, this gives an indication of good accuracy for the results. If not, the user has to consider one of them more than the other, the amount of trustworthy between the supervision and the feedback is case dependent and we left it to the user to decide the amount of significance that each one of them must contribute in the final results [5].

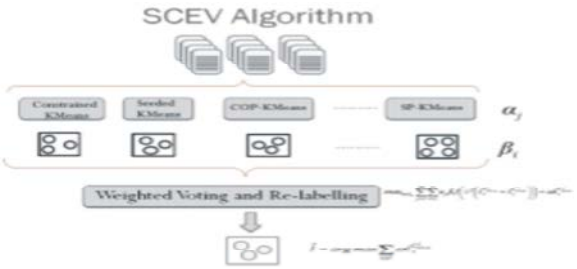


Fig. 2: The places where supervision and feedback can be applied

**Ensemble-Update (ENU):** Every time a new data chunk Dn appears, we train a new model Mn from Dn and update the ensemble by choosing thebest L modelsfromtheexisting L+1models(M∪{Mn}). Algorithm 1 sketches this updating process.

**The Algorithm of Ensemble-Update:**

Input: Xn,Yn: training data points and class labels associated
With some of these points in chunk Dn
Zn: test data points in chunk Dn
K: number of clusters to be created
M: current ensemble of L models {M1... ML}

Output: Updated ensemble M
1: Obtain K clusters {Xn 1... Xn K} using E-M algorithm. Andcompute their summary of statistics {Mn 1... Mn K}
2: if no cluster Mi j ◎ M contains some class c that is seen in thenew model Mn then
3: Refine-Ensemble (M, Mn)
4: end if
5: Test each model Mi j ◎ M and Mn on the labelled data ofXnand obtain its accuracy
6: M ←Best L models in M ∪ {Mn} based on accuracy.
7: Predict the class labels of data points inZn with M.

We will probably have maximum reduction in prediction error for a single model. However, if the same set of micro-clusters are injected in all the models, then the correlation among them may increase, resulting in reduced prediction accuracy of the ensemble [6]. Lemma 1 states that the ensemble error is the lowest when $\rho = 0$.

**Semi Supervised Clustering with DCP:** Unlike for a classification problem, the 'name' of a particular cluster is irrelevant. For example, consider the clustering {{1,4},{2,3}}. Whether we call the set {1, 4} 'cluster 1' or 'cluster 2' is arbitrary, the important information is that 1 and 4 belong to the same cluster whilst 2 and 3 belong to another one.

We therefore can encode the relevant information about the clustering of points in X using a binary indicator matrix C, where for xi,xj ◎ X. Moreover notice that every partition S ◎ S will have a unique such binary matrix representation which we denote CS.

In the semi-supervised setting we assume that some portion of our data set is labelled. Let Z ◎ X be the set of observed points for which we have labels, i.e. we observe the binary indicator matrix ˆ CZ defined on pairs of inputs xi,xj◎ Z. For a given kernel function and hyperparameters θ ◎ Θ our DCP likelihood function becomes

$$p(\mathcal{S} \in \mathbb{S} | \hat{C}_Z, \theta, \tau) \propto \prod_{S \in \mathcal{S}} \det\left(K_S^\theta\right)^{-\tau} \prod_{x,y \in Z} \mathbb{1}\left(C_{\mathcal{S}}(x,y) = \hat{C}_Z(x,y)\right)$$

$$C(x_i, x_j) = \begin{cases} 1 & \text{if } x_i, x_j \text{ in the same cluster,} \\ 0 & \text{otherwise.} \end{cases}$$

**Clustering Laplacian Score (CLS):** In this section we present a brief description of the CLS score [7] upon which we depend in our framework. In fact, CLS utilizes both parts of data, labeled and unlabelled. The labeled part is transformed into pairwise constraints, which can be classified on two subsets: ΩML (a set of Must-Link constraints) and ΩCL (a set of Cannot-Link constraints)

– Must-Link constraint (ML): involving two instances xi and xj, specifies that they have the same label.
– Cannot-Link constraint (CL): involving two instances xi and xj, specifies that they have difierent labels.

Let X be a dataset of n instances characterized by p features. X consists of two subsets: XL for labeled data and XU for unlabelled data. Let r be a feature to evaluate. We define its vector by fr = (fr1...frn). The CLS of r, which should be minimized, is computed by:

$$CLS_r = \frac{\sum_{i,j}(f_{ri} - f_{rj})^2 S_{ij}}{\sum_i \sum_{j|\exists l,(x_i,x_j)\in\Omega_{CL}} (f_{ri} - \alpha_{rj}^i)^2 D_{ii}}$$

**The Algorithm of CLS:**
**Require:** A data set X (n×p), which consists of two subsets: XL (L×p), the subset of labeled training instances and XU (U ×p), the subset of unlabeled training instances; the input space (F = {f1,...,fp}); the constant λ and the neighbourhood degree k.

1: Construct the constraint sets (ΩML and ΩCL) from the labeled part: XL.
2: Calculate the dissimilarity matrix S and the diagonal matrix D.
3: for r = 1 to p do
4: Calculate CLSr according to eq (1).
5: end for

**Experiments:** In this section, we empirically demonstrate that our proposed semi-supervised clustering algorithm is both eficient and efiective.

**Datasets:** Four real-world benchmark datasets with varied sizes are used in our experiments, which are:

- Soybean, There are 19 classes, only the first 15 of which have been used in prior work. The folklore seems to be that the last four classes are unjustified by the data since they have so few examples. There are 35 categorical attributes, some nominal and some ordered. The value "dna" means does not apply. The values for attributes are encoded numerically, with the first value encoded as "0," the second as "1," and so forth. An unknown value is encoded as "?".

- Wine, These data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines.

I think that the initial data set had around 30 variables, but for some reason I only have the 13 dimensional versions. I had a list of what the 30 or so variables were, but a.) I lost it and b.), I would not know which 13 variables are included in the set.

- Musk, This dataset describes a set of 92 molecules of which 47 are judged by human experts to be musks and the remaining 45 molecules are judged to be non-

musks. The goal is to learn to predict whether new molecules will be musks or non-musks. However, the 166 features that describe these molecules depend upon the exact shape, or conformation, of the molecule. Because bonds can rotate, a single molecule can adopt many different shapes. To generate this data set, the low-energy conformations of the molecules were generated and then filtered to remove highly similar conformations. This left 476 conformations. Then, a feature vector was extracted that describes each conformation.

- Zoo, A simple database containing 17 Boolean-valued attributes. The "type" attribute appears to be the class attribute. Here is a breakdown of which animals are in which type: (I find it unusual that there are 2 instances of "frog" and one of "girl"!)

## EXPERIMENTAL RESULTS

### Soybean Dataset Results

| | SOYBEAN Dataset | | | |
|---|---|---|---|---|
| Algorithm | Accuracy | Precision | Recall | F-Measure |
| SCEV | 89.45 | 87.90 | 92.77 | 90.89 |
| ENU | 79.90 | 76.08 | 74.78 | 86.56 |
| DCP | 70.90 | 79.67 | 79.89 | 85.78 |
| CLS | 84.67 | 90.67 | 86.78 | 77.67 |

The above graph shows that performance of Soybean dataset. The Accuracy of SCEV algorithm is 89.45 which is higher when compare to other three (ENU, DCP, CLS) algorithms. The Precision of CLS algorithm is 90.67 which is higher when compare to other three (SCEV, ENU, DCP) algorithms. The Recall of SCEV algorithm is 92.77 which is higher when compare to other three (ENU, DCP, CLS) algorithms. The F-Measure of SCEV algorithm is 90.89 which is higher when compare to other three (ENU, DCP, CLS) algorithms.

### Wine Dataset Results

| | Wine Dataset | | | |
|---|---|---|---|---|
| Algorithm | Accuracy | Precision | Recall | F-Measure |
| SCEV | 78.45 | 85.90 | 94.77 | 88.89 |
| ENU | 74.98 | 86.08 | 94.78 | 60.56 |
| DCP | 70.89 | 90.67 | 91.89 | 85.78 |
| CLS | 80.67 | 96.67 | 70.78 | 88.67 |

The above graph shows that performance of Wine dataset. The Accuracy of CLS algorithm is 80.67 which is higher when compare to other three (ENU, DCP, SCEV) algorithms. The Precision of CLS algorithm is 96.67 which is higher when compare to other three (SCEV, ENU, DCP) algorithms. The Recall of ENU algorithm is 94.78 which is higher when compare to other three (SCEV, DCP, CLS) algorithms. The F-Measure of SCEV algorithm is 88.89 which is higher when compare to other three (ENU, DCP, CLS) algorithms.

### Musk Dataset Results

| | Musk Dataset | | | |
|---|---|---|---|---|
| Algorithm | Accuracy | Precision | Recall | F-Measure |
| SCEV | 79.45 | 88.90 | 84.77 | 88.89 |
| ENU | 74.90 | 90.08 | 90.78 | 70.56 |
| DCP | 80.98 | 76.67 | 72.89 | 85.78 |
| CLS | 88.67 | 70.67 | 77.78 | 90.67 |

The above graph shows that performance of Musk dataset. The Accuracy of CLS algorithm is 88.67 which is higher when compare to other three (ENU, DCP, SCEV) algorithms. The Precision of ENU algorithm is 90.08 which is higher when compare to other three (SCEV, CLS, DCP) algorithms. The Recall of ENU algorithm is 90.78 which is higher when compare to other three (SCEV, DCP, CLS) algorithms. The F-Measure of CLS algorithm is 90.67 which is higher when compare to other three (ENU, DCP, SCEV) algorithms.

**Zoo Dataset Results**

| | Zoo Dataset | | | |
|---|---|---|---|---|
| Algorithm | Accuracy | Precision | Recall | F-Measure |
| SCEV | 93.45 | 88.9 | 87.77 | 78.89 |
| ENU | 80.9 | 90.08 | 93.78 | 70.56 |
| DCP | 84.98 | 93.67 | 79.89 | 85.78 |
| CLS | 88.67 | 84.67 | 71.78 | 89.67 |

The above graph shows that performance of Zoo dataset. The Accuracy of SCEV algorithm is 93.45 which is higher when compare to other three (ENU, DCP, CLS) algorithms. The Precision of DCP algorithm is 93.67 which is higher when compare to other three (SCEV, CLS, ENU) algorithms. The Recall of ENU algorithm is 93.78 which is higher when compare to other three (SCEV, DCP, CLS) algorithms. The F-Measure of CLS algorithm is 89.67 which is higher when compare to other three (ENU, DCP, SCEV) algorithms.

## CONCLUSION

Combining supervision with clustering ensemble is assumed to give higher level of accuracy. The supervision at ensemble generation step can aid and bias different clustering to produce better and high quality base partitions. The consensus function can also take benefit from user feedback about the base partitions to produce higher quality target partition. This approach gives user flexibility of choosing multiple types of supervision and feedback in both steps. This type of weighting scheme can be applied to other consensus functions as well. However, we choose relabeling and voting based approach for its simplicity, high robustness, scalability and ease of implementation. The proposed consensus function is computationally very efficient with complexity linear to the data size; N.

The remarkable property of this model is the fact that it overcomes the typically dificult task of dealing with complex high dimensional data. As long as any sensible positive definite kernel can be computed between pairs of data points, the determinantal clustering process can infer interesting properties about the data. This feature, combined with not having to prespecify the number of clusters makes the DCP a great contender for analysing complex data sets such as biological sequences, images, text and other multimedia.

## REFERNCES


1. Adams, R.P. and Z. Ghahramani, 2009. Archipelago: Nonparametric Bayesian Semi-Supervised Clustering, Proceedings of the 26th International Conference on Machine Learning,
2. Lawrence, N.D. and M.I. Jordan, 2010. Semi-supervised Clustering via Gaussian Processes, Advances in Neural Information Processing Systems, pp: 753-760.
3. X. Zhu, 2011. Semi-Supervised Clustering Literature Survey. Technical Report 1530, Dept. of Computer Sciences, University of Wisconsin, Madison,
4. Bilenko, M., S. Basu and R.J. Mooney, 2014. Integrating constraints and metric learning in semi-supervised clustering, In ICML,
5. Fern, X.Z. and C.E. Brodley, 2012. Solving cluster ensemble problems by bipartite graph partitioning, In ICML, 2012.
6. Strehland, A. and J. Ghosh, 2009. Clusterensembles –aknowledge reuse framework for combining multiple partitions, JMLR, 3: 583-617.
7. Topchy, A., A. Jain and W. Punch, 2010. A mixture model for clustering ensembles, In Proc. SIAM Data Mining, pp: 379-390.
8. Topchy, A., A.K. Jain and W. Punch, 2011. Clustering ensembles: Models of consensus and weak partitions, IEEE Transaction on Pattern Analysis and Machine Intelligence, 27(12): 1866-1881.

9. Ghaemi, R., M.N. Sulaiman, H. Ibrahim and N. Mustapha, 2009. A Survey: Clustering Ensembles Techniques, Proceedings of World Academy of Science, Engineering AND Technology, 38: 2070-3740.
10. Azimi, J., M. Abdoos and M. Analoui, 2010. A new efficient approach in clustering ensembles, IDEAL LNCS, 4881: 395-405.
11. Basu, S., M. Bilenko and R.J. Mooney, 2014. A probabilistic framework for semi-supervised clustering, In Proc. KDD.
12. Frigui, H. and O. Nasraoui, 2004. Semi supervised Clustering of prototypes and attribute weights, Pattern Recognition, 37(3): 567-581.