Distributed Clustering using Semi-supervised Fusion and Feature Reduction Preprocessing

Huaying Li and Aleksandar Jeremic*
Department of Electrical and Computer Engineering, McMaster University, Hamilton, Ontario, Canada

Keywords: Clustering, Information Fusion, Cluster Ensemble, Semi-supervised Learning.

Abstract: In the recent years there has been tremendous development of data acquisition system resulting in a whole new set of so called big data problems. In addition to other techniques data analysis of these data sets involves significant amount of clustering and/or classification. Due to a heterogeneous nature of the data sets the performance of these algorithms can vary significantly in different applications. In our previous work we proposed semi-supervised information fusion system and demonstrated its performance in various applications. In this paper we proposed to improve the performance of the proposed system by applying data preprocessing algorithms using feature reduction as well as various base clustering techniques. We demonstrate the applicability of the proposed techniques using real data sets.

1 INTRODUCTION

Thee major goal of data clustering is to find the hidden structure of a given data set by dividing data points into distinct clusters based on certain criteria. Data points in the same cluster are expected to be similar to each other than to a data point from another cluster. Although many clustering algorithms exist in the literature, in practice no single algorithm can correctly identify the underlying structure of all data sets (Jain and Dubes, 1988), (Xu and Wunsch, 2008). Furthermore, it is usually difficult to select a suitable clustering algorithm for a given data set when the prior information about cluster shape and size is not available. Therefore, in many applications one option to improve the clustering results is to generate multiple base clusterings and combine them into a consensus clustering (Strehl and Ghosh, 2003),(Vega-Pons and Ruiz-Shulcloper, 2011). This is often referred to as clustering ensemble. Many existing clustering ensemble methods consist of two major steps: generation and fusion of multiple base clusterings. Nowadays, there is a growing interest in utilizing additional supervision information in the unsupervised learning process (such as clustering) to improve the performance. This is often referred to as semi-supervised clustering (Chapelle et al., 2006).

Motivated by the success of both approaches, researchers become interested in combining the benefits of both techniques to further improve clustering results. The supervision information of semi-supervised learning can be provided and utilized in either step of clustering ensemble methods. In (Iqbal et al., 2012), the supervision information is utilized in the base clustering generation step, i.e., applying semi-supervised clustering algorithms to generate the set of base clusterings and fuse the cluster labels without supervision. In this paper, we propose to utilize the supervision information in the fusion step, i.e., applying unsupervised clustering algorithms to generated the set of base clusterings and fuse the cluster labels with supervision. The remainder of this paper is organized as follows. In Section 2, we propose the modified semi-supervised clustering ensemble algorithm using data preprocessing based on variable base clustering generation and normalization. In Section 3, we demonstrate the performance of our proposed algorithms and the effect of normalization in clustering ensemble. In the last section, we give the summary of current research work and also list some future research direction we will continue to work on.

2 SEMI-SUPERVISED CLUSTERING ENSEMBLE

Clustering ensemble methods usually consists of two major steps: the generation and fusion of base clusterings, as shown in Fig. 1. In this section, we propose four different ways to generate a set of base clusterings and two different ways to combine the set into a consensus clustering.

*This work was supported by Natural Sciences and Engineering Research Council of Canada.

Li, H. and Jeremic, A.
Distributed Clustering using Semi-supervised Fusion and Feature Reduction Preprocessing.
DOI: 10.5220/0006658002330239
In *Proceedings of the 11th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2018) - Volume 4: BIOSIGNALS*, pages 233-239
ISBN: 978-989-758-279-0

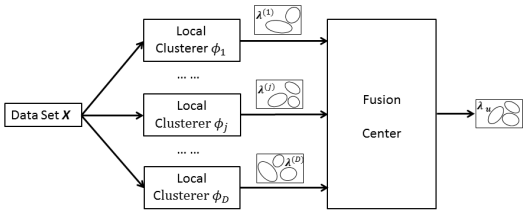

Figure 1: Distributed clustering system.

2.1 Base Clustering Generation

In this paper, the term clusterer represent the processing unit that produces cluster labels for the given data input. The set of D local clusterers (Fig. 1) is viewed as a black box, which takes data set $\mathbf{X}$ as the input and produces a set of base clusterings as the output. We name it as the base clustering generator Φ (BCG). The internal structure of base clustering generator is shown in Fig. 2.

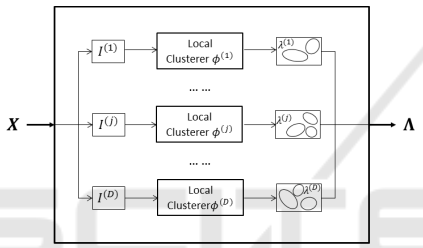

Figure 2: Base clustering generator with D local clusterers.

As a preliminary approach we apply K-means algorithm in each local clusterer. In practice, different clustering algorithms can be implemented in the local clusterers to generate base clusterings. In order to describe the setting of base clustering generator, we first define some necessary parameters as follow:

- $\phi^{(j)}$: j-th local clusterer
- D: total number of local clusterers in Φ
- $I^{(j)}$: input of local clusterer $\phi^{(j)}$
- $\lambda^{(j)}$: output of local clusterer $\phi^{(j)}$
- $K^{(j)}$: number of clusters in $\lambda^{(j)}$

One possible way to design the base clustering generator is to build D identical local clusterers and apply the same clustering algorithm with different initializations in each local clusterer. We set $D = 21$ and denote this base clustering generator as Φ_1. The set of base clusterings generated by Φ_1 is named as "BASE1". The parameter settings of Φ_1 is listed in Table 1. In this design, the clustering processes are distributed over different local clusterers. The advantage is that each local clusterer has the access to the entire data matrix and generates base clusterings based on all the information. In the literature, many clustering ensemble methods are evaluated by generating base clusterings in this way (Strehl and Ghosh, 2003),(Fred and Jain, 2005),(Visalakshi and Thangavel, 2009).

Table 1: Base clustering generators: $\mathbf{X}$ represents the input data matrix, F represents the number of features (columns) of $\mathbf{X}$, and $\mathbf{x}^{(j)}$ represents the j-th feature (column) of $\mathbf{X}$.

Base Clustering Generator	Set Name	No. of Local Clusterers (D)	Local Clusterer $\phi^{(j)}$ Input ($I^{(j)}$)	Local Clusterer $\phi^{(j)}$ No. of Clusters ($K^{(j)}$)
Φ_1	BASE1	21	$\mathbf{X}$	$K^{(j)} = K_0$
Φ_2	BASE2	F	$\mathbf{x}^{(j)}$	$K^{(j)} = K_0$
Φ_3	BASE3	21	$\mathbf{X}$	$K^{(j)} \in [K_0, 40]$
Φ_4	BASE4	F	$\mathbf{x}^{(j)}$	$K^{(j)} \in [K_0, 40]$

Another way to design the base clustering generator is to apply clustering algorithm to only one of the data features in each local clusterer. For a data set containing F features, there are $D = F$ local clusterers in the generator. We denote this base clustering generator as Φ_2 and the set of base clusterings generated by this generator as "BASE2". The parameter settings of Φ_2 is listed in Table 1. In this design, data features are distributed over different local clusterers. Each local clusterer only has the access to one of the features and partitions data points from a specific aspect of the data. It is suitable for data sets whose features are measured in diverse scales. It is also suitable for data sets whose features are heterogeneous or categorical when the dissimilarity measure based on all features does not have a real meaning. Furthermore, the aforementioned approach may be the only choice when the features or attributes of the data set are not shareable between organizations due to privacy, ownership or other reasons.

Note that recently proposed MCLA algorithm (Strehl and Ghosh, 2003) is also based on clustering clusters. Similar clusters (from different clusterings) are grouped together to form a meta-cluster, which is finally collapsed into a consensus cluster. Intuitively, it is easier to identify similar clusters with less number of data points. Therefore, we modify base clustering generator Φ_1 and Φ_2 by setting $K^{(j)}$ to relatively larger integers. Due to the fact that optimal value of $K^{(j)}$ is data dependent and to avoid the selection of a suitable value for $K^{(j)}$, we propose to randomly select an integer value for $K^{(j)}$ of each local clusterer. The parameter settings of the modified base clustering generators Φ_3 and Φ_4 are also listed in Table 1. The sets of base clusterings generated by the modified generators are denoted as "BASE3" and "BASE4" respectively.

Suppose the input data set $\mathbf{X}$ is the combination of a training set $\mathbf{X}_r$ and a testing set $\mathbf{X}_u$. The training set $\mathbf{X}_r$ contains data points $\{\mathbf{x}_1, \ldots, \mathbf{x}_{N_r}\}$, for which labels are provided in a label vector λ_r. The testing data set

$\mathbf{X}_u$ contains data points $\{\mathbf{x}_{N_r+1},\ldots,\mathbf{x}_N\}$, the labels of which are unknown. The consensus cluster label vector (output of SEA) of testing set $\mathbf{X}_u$ is denoted by λ_u. The size of training set $\mathbf{X}_r$ is measured by the number of data points in the training set and denoted by N_r, i.e., $|\mathbf{X}_r| = N_r$. Similarly, the size of testing set $\mathbf{X}_u$ is measured by the number of data points in the testing set and denoted by N_u, i.e., $|\mathbf{X}_u| = N_u$. According to the training and testing sets, the label matrix $\mathcal{F}$ can be partitioned into two block matrices $\mathcal{F}_r$ and $\mathcal{F}_u$, each of which contains all the labels corresponding to the data points in the training set $\mathbf{X}_r$ and testing set $\mathbf{X}_u$ respectively. Suppose training data points belong to K_0 classes and all training points from the k-th class form one cluster, denoted by C_r^k ($k = 1,\ldots,K_0$). Therefore, the training set $\mathbf{X_r}$ consists of a set of K_0 clusters $\{C_r^1,\ldots,C_r^k,\ldots,C_r^{K_0}\}$. If the size of cluster C_r^k is denoted by N_r^k, the total number of training points equals to the sum of N_r^k, i.e., $N_r^k = \sum_{k=1}^{K_0} N_r^k$. We rearrange label matrix $\mathcal{F}_r$ to form K_0 block matrices: $[\mathcal{F}_r^1,\ldots,\mathcal{F}_r^k,\ldots,\mathcal{F}_r^{K_0}]$. Each block matrix $\mathcal{F}_r^k$ contains the base cluster labels of data points in the k-th training cluster C_r^k.

The SHSEA is defined as follows: (1) for a particular data point count the number of agreements between its label and the labels of training points in each training cluster, according to an individual base clustering (2) calculate the association vector between this data point and the corresponding base clustering, (3) compute the average association vector by averaging the association vectors between this data point and all base clusterings and (4) repeat for all data points and derive the soft consensus clustering for the testing set. Since the overall consensus cluster labels are derived from the fuzzy(soft) label matrix, we name this approach as the soft-to-hard semi-supervised clustering ensemble algorithm (SHSEA).

According to the j-th clustering $\lambda^{(j)}$, we compute the association vector $\mathbf{a}_i^{(j)}$ for the i-th unlabelled data point $\mathbf{x}_i$, where $i = 1,\ldots,N_u$ and $j = 1,\ldots,D$. Since there are K_0 training clusters, the association vector $\mathbf{a}_i^{(j)}$ has K_0 entries. Each entry describes the association between data point $\mathbf{x}_i$ and the corresponding training cluster. The k-th entry of the association vector $\mathbf{a}_i^{(j)}$ is measured as the occurrence of cluster label of data point $\mathbf{x}_i$ among the labels of reference data points in the k-th training cluster (according to base clustering $\lambda^{(j)}$), i.e.,

$$\mathbf{a}_i^{(j)}(k) = \frac{\text{occurrence of } \mathcal{F}_u(i,j) \text{ in } \mathcal{F}_r^k(:,j)}{N_r^k}, \qquad (1)$$

where $\mathcal{F}_u(i,j)$ represents the cluster label of data point $\mathbf{x}_i$ and $\mathcal{F}_r^k(:,j)$ represents the labels of reference points in the k-th training category generated according to base clustering $\lambda^{(j)}$. In order to fuse the set of base clusterings, the weighted average association vector $\mathbf{a}_i$ of data point $\mathbf{x}_i$ is computed by averaging D association vectors $\mathbf{a}_i^{(j)}$, i.e.,

$$\mathbf{a}_i = \sum_{j=1}^{D} \omega_j \mathbf{a}_i^{(j)}, \qquad (2)$$

where ω_j is the corresponding weight of the j-th local clusterer. When local clusterers are equally important, $\omega_j = 1/D$. Each entry of $\mathbf{a}_i$ describes the overall association between data point $\mathbf{x}_i$ and the corresponding training cluster. As a consequence, the summation of all the entries of $\mathbf{a}_i$ could be used to describe the association between data point $\mathbf{x}_i$ and all the training clusters quantitatively. We define it as the association level of data point $\mathbf{x}_i$ to all training clusters and denote it as γ_i, i.e.,

$$\gamma_i = \sum_{k=1}^{K_0} \mathbf{a}_i(k). \qquad (3)$$

By computing association level for all data observations, the association level vector γ_u for the testing set $\mathbf{X}_u$ is made up by stacking association level γ_i for all $i = 1,\ldots,N_u$, i.e., $\gamma_u = [\gamma_1,\gamma_2,\ldots,\gamma_{N_u}]^T$. We have two options to present the overall consensus clustering for testing set $\mathbf{X}_u$. One option is to produce a soft consensus label matrix Λ_u. The i-th row of Λ_u is computed by normalizing the average association vector $\mathbf{a}_i$, i.e.,

$$\Lambda_u(i,:) = \mathbf{a}_i^T/\gamma_i. \qquad (4)$$

The other option is to produce a hard consensus label vector λ_u. The consensus cluster label assigned to each data point is its most associated category labels in the corresponding average association vector. Since the overall hard cluster labels are assigned according to the soft label matrix, we name this algorithm as the soft-to-hard semi-supervised clustering ensemble algorithm (SHSEA). The normalized soft consensus label matrix (Λ_u) can be used as the output of the algorithm.

Following the naming convention, the other semi-supervised ensemble method is called hard-to-hard semi-supervised clustering algorithm (HHSEA), since the overall cluster labels are assigned based on hard label matrix. The HHSEA is defined as follows: (1) for a particular data point count the number of agreements between its label and the labels of training points in each training cluster, according to an individual base clustering, (2) calculate the association vector between this data point and the corresponding base clustering, (3) assign this data point to its most associated cluster label (4) repeat for all data points

and all base clusterings to relabel the labels in matrix $\mathcal{F}_u$ and (5) apply majority voting to derive hard consensus clustering. The details of both SHSEA and HHSEA are given in our previous work (Li and Jeremić, 2017).

3 NUMERICAL EXAMPLES

In this section, we evaluate the performance of the proposed distributed clustering system using the breast cancer data cells. This data is used to study human breast cancer cells undergoing treatment of different drugs. The cancer cells are plated into clear-bottom well plates and 10 types of treatments are taken placed to the cells. Images of the untreated and treated cells are captured using the high content imaging system and processed by the CAFE (Classification and Feature Extraction of micro-graphs of cells) software to extract useful information. In total 705 attributes/features per cell are recorded for further analysis (Razeghi Jahromi, 2014).

Since the ground truth of class assignments for each data set are available, we use micro-precision (Modha and Spangler, 2003) as our metric to measure the accuracy of clustering result with respect to the expected (true) labelling. Recall that data set $\mathbf{X}$ contains N data points that belong to K_0 classes and N_k represents the number of data points in the k-th cluster that are correctly assigned to the corresponding class. Corresponding class here represents the true class that has the largest overlap with the k-th cluster. The micro-precision (MP) is calculated by $\text{MP} = \sum_{k=1}^{K_0} N_k/N$. The data set that are used in this paper are listed Table 2, including the number of data points, features and classes. The available data points are divided in testing and training sets (data points with know reference labels).

Table 2: Data Information I: the number of data points, features and classes.

Data Sets	Data Points	Features	Classes
DataSet1	300	705	2
DataSet2	300	705	2
DataSet3	300	705	2
DataSet4	450	705	3

3.1 Data Pre-processing

Data pre-processing is a necessary step to improve the results of cluster analysis (Liu and Motoda, 1998), (Pyle, 1999). In practice, many data sets to be clustered contain features that are measured in different units and scales. Features measured in relatively large scales may play a dominant role in the similarity measure and influence the accuracy of the clustering results. As a consequence, normalizing the features is an important pre-processing procedure, especially when the similarity measure is based on Euclidean distances (de Souto et al., 2008). Min-max normalization is a linear transformation of features into a specified range, which equalize the magnitude of the features and prevents over weighting features measured in relatively large scale over features measured in relatively small scale. Suppose $\mathbf{x}^{(f)}$ represents the f-th feature of data set $\mathbf{X}$. Let $\mathbf{x}_{max}^{(f)}$ and $\mathbf{x}_{min}^{(f)}$ represent the maximum and minimum value of the f-th feature respectively. Min-max normalization maps the f-th feature into range [0, 1] by

$$\mathbf{x}_{Norm}^{(f)} = \frac{\mathbf{x}^{(f)} - \mathbf{x}_{max}^{(f)}}{\mathbf{x}_{max}^{(f)} - \mathbf{x}_{min}^{(f)}}. \quad (5)$$

In this paper, we demonstrate the effect of normalization in clustering ensemble methods by comparing the clustering results using original data sets (without any pre-processing) and normalized data sets.

3.2 Original Data Sets

To study the effect of base clusterings on clustering ensemble problem, we generate four different sets of base clusterings (BASE1 to BASE4) for each data set. Note that base clustering generator $\mathcal{F}_1$ is designed based on the common way used in the literature to generate base clusterings (Strehl and Ghosh, 2003),(Wang et al., 2011),(Dudoit and Fridlyand, 2003),(Fred and Jain, 2005). To evaluate different clustering ensemble methods, we apply the unsupervised HGPA, CSPA, MCLA (Strehl and Ghosh, 2003) and BCE (Wang et al., 2011) in the fusion center and compare the performance to the proposed SHSEA and HHSEA. Recall that the ratio of number of reference data points (N_r) to number of testing data points (N_u) is denoted by P. We set $P = 25\%$ in the experiments and repeat each experiment 100 times to calculate the average micro-precision.

The micro-precision of K-means clustering algorithm using all original features is listed in Table 3. The maximum and minimum micro-precision of K-means using features individually are also listed in Table 3. Among all 11 data sets maximum MP of K-means using single feature is higher than MP of K-means using all features. Recall that BASE1 set of base clusterings is generated by repetitively applying K-means to all features together, while BASE2 is generated by applying K-means to each feature individually. Therefore, we expect the micro-precision of ensemble methods using BASE2 to be higher than that

Table 3: Micro-precision of K-means using all features and single feature of original data.

Data Sets	Kmeans		
	All Features	Single Feature	
		Max	Min
DWALabSet1	0.5033	0.7917	0.5000
DWALabSet2	0.5033	0.7233	0.5000
DWALabSet3	0.5367	0.7933	0.5000
DWALabSet4	0.3400	0.5642	0.3333

of BASE1 since BASE2 contains a certain number of "better" base clusterings. In addition, the performance of SHSEA using BASE2 is expected to be better than HHSEA, since base clusterings with higher MP are given larger weights in the consensus fusion step. Furthermore, recall that BASE3 (BASE4) is generated in the same way as BASE1 (BASE2) respectively expect that $K^{(j)}$ (number of clusters in each local clusterers) are set to be greater than K_0 (expected number of clusters). Therefore, we expect the performance of SHSEA and HHSEA using BASE3 (BASE4) to be better than BASE1 (BASE2), since the proposed semi-supervised methods are expected to perform better when data points are divided into smaller groups.

The micro-precision of our proposed system (four unsupervised and two semi-supervised ensemble methods) using four sets of base clusterings (BASE1 to BASE4) is illustrated by sub-figure (a) of Fig. 3 to Fig. 6. The performance of SHSEA and HHSEA is represented by series SH(P25) and HH(P25) respectively and $P25$ means the ratio of reference and testing points is $P = 25\%$. Among four groups of clustering results, the bar corresponding to the highest average MP of the unsupervised ensemble methods and the bars corresponding to the highest MP of SHSEA and HHSEA are labelled in each chart. It is clear that the performance of the proposed semi-supervised methods conforms with our expectations.

Compared to the micro-precision of K-means algorithm (Table 3), the clustering results has been improved by both operational modes of the proposed system. The performance of the semi-supervised mode is better than the unsupervised mode (except "DataSet1"). The winning set of base clusterings is either BASE2 or BASE4. In all the example the best performance is achieved by utilizing SHSEA.

To study the effect of quantity of reference points on semi-supervised clustering ensemble methods, we repeat the experiments in semi-supervised mode by selecting different numbers of reference points, i.e., by varying the value of P in $N_r = P \cdot N_u$. Compared to the performance of K-means (Table 3), micro-precision of SHSEA or HHSEA increases dramatically when P is relatively small. It becomes steady and sometimes starts to decrease as P increases. Therefore, for the purpose of improving the performance of semi-supervised ensemble algorithms may not be beneficial to label more data points. It is due to the facts that more reference points do not guarantee the improvement and obtaining additional labels is time-consuming and expensive.

Recall that the number of clusters in the j-th base clustering $K^{(j)}$ is randomly generated in the base clustering generator Φ_3 and Φ_4. To study the effect of randomized $K^{(j)}$ on the clustering ensemble methods, we repeat the experiments by setting the number of clusters in each base clustering the same and varying the value of $K^{(j)}$. Among these data sets, the highest MP occurs at different $K^{(j)}$. The performance of the proposed system using randomized $K^{(j)}$ is either the best of all tested values of $K^{(j)}$ or it is very closed to best. Due to the fact that we lack the knowledge on how to select the optimal $K^{(j)}$, we use randomized $K^{(j)}$ in the following experiments to avoid the selection of $K^{(j)}$ for each data set.

3.3 Normalized Data Sets

The micro-precision of K-means using all normalized features and normalized features individually is shown in Table 4. The performance of K-means using all features has been improved significantly by normalization except the first three data sets, as compared to Table 3. As discussed earlier the performance of distance-based clustering algorithms may be affected when data sets to be clustered contains features measured in diverse scales. By investigating features of each data set, we noticed that the data sets contain features measured in quite different ranges. Moreover, the performance of K-means using normalized features individually is similar to the performance of K-means using original features individually. This result is expected since similarity measure for single feature is based on 1-dimensional distance calculation and it is invariant to the feature scales.

Table 4: Micro-precision of K-means using all features and single feature or normalized data.

Data Sets (Normalized)	Kmeans		
	All Features	Single Feature	
		Max	Min
DWALabSet1	0.6628	0.7920	0.5000
DWALabSet2	0.5609	0.7233	0.5000
DWALabSet3	0.6120	0.7933	0.5000
DWALabSet4	0.5058	0.5644	0.3333

To study the effect of normalization on clustering ensemble methods, we repeat the experiments previously described in Section 3.2 using normalized data sets. The micro-precision of the proposed system is illustrated by sub-figures (b) of Fig. 3 to Fig. 6.

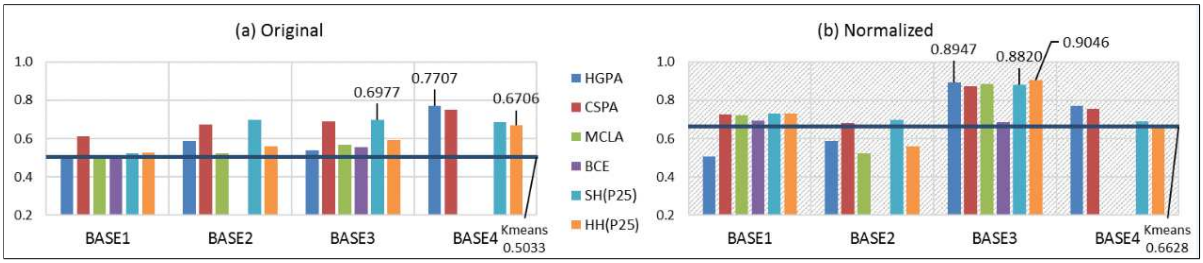

Figure 3: Data Set: DWALabSet1.

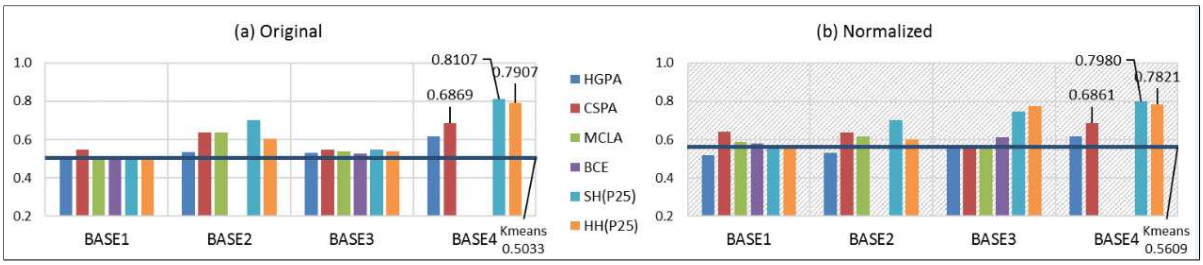

Figure 4: Data Set: DWALabSet2.

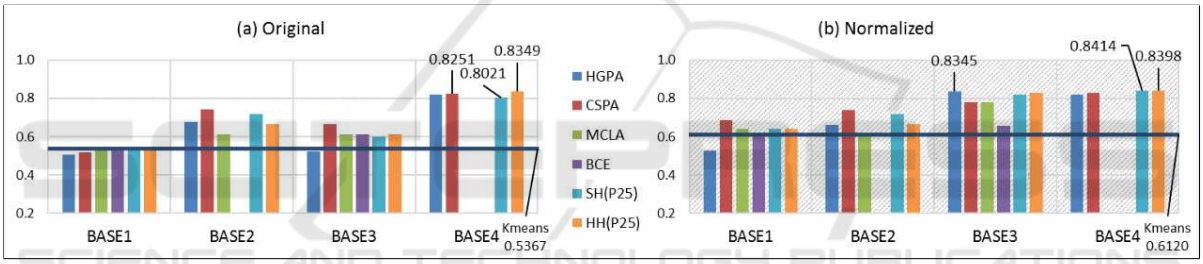

Figure 5: Data Set: DWALabSet3.

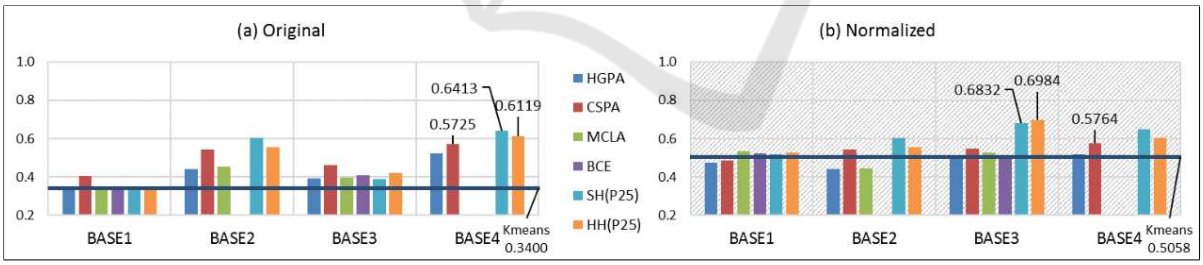

Figure 6: Data Set: DWALabSet4.

Note that the system performance using BASE1 and BASE3 has been improved by normalization, while the system performance using BASE2 and BASE4 stays close to the system performance using the corresponding sets of base clusterings obtained by clustering original data sets. It is also expected since normalization does not affect the performance of K-means using single feature. Overall it can be observed that the performance of SHSEA is very close to the performance of HHSEA using normalized data.

4 CONCLUSIONS

In this paper we have proposed semi-supervised clustering ensemble algorithms based on utilizing labelled training data to improve the clustering results. We designed four different ways to generate base clusterings and two ways to fuse them in the fusion center with supervision. We provided numerical examples to demonstrate the effect of base clusterings on the clustering ensemble methods and the performance

of semi-supervised clustering algorithms. We also demonstrated the effect of normalization in the clustering ensemble. In the future, we will focus on utilizing the supervision information in both steps of the clustering ensemble methods.

REFERENCES

Chapelle, O., Schölkopf, B., Zien, A., et al. (2006). Semi-supervised learning.

de Souto, M. C., de Araujo, D. S., Costa, I. G., Soares, R. G., Ludermir, T. B., and Schliep, A. (2008). Comparative study on normalization procedures for cluster analysis of gene expression datasets. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 2792–2798. IEEE.

Dudoit, S. and Fridlyand, J. (2003). Bagging to improve the accuracy of a clustering procedure. *Bioinformatics*, 19(9):1090–1099.

Fred, A. L. and Jain, A. K. (2005). Combining multiple clusterings using evidence accumulation. *IEEE transactions on pattern analysis and machine intelligence*, 27(6):835–850.

Iqbal, A. M., Moh'd, A., and Khan, Z. (2012). Semi-supervised clustering ensemble by voting. *arXiv preprint arXiv:1208.4138*.

Jain, A. K. and Dubes, R. C. (1988). *Algorithms for clustering data*. Prentice-Hall, Inc.

Li, H. and Jeremić, A. (2017). Semi-supervised distributed clustering for bioinformatics - comparison study. In *BIOSIGNALS 2017*, pages 649–652.

Liu, H. and Motoda, H. (1998). *Feature extraction, construction and selection: A data mining perspective*. Springer Science & Business Media.

Modha, D. S. and Spangler, W. S. (2003). Feature weighting in k-means clustering. *Machine learning*, 52(3):217–237.

Pyle, D. (1999). *Data preparation for data mining*, volume 1. Morgan Kaufmann.

Razeghi Jahromi, M. (2014). *FRECHET MEANS OF RIEMANNIAN DISTANCES: EVALUATIONS AND APPLICATIONS*. PhD thesis.

Strehl, A. and Ghosh, J. (2003). Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *The Journal of Machine Learning Research*, 3:583–617.

Vega-Pons, S. and Ruiz-Shulcloper, J. (2011). A survey of clustering ensemble algorithms. *International Journal of Pattern Recognition and Artificial Intelligence*, 25(03):337–372.

Visalakshi, N. K. and Thangavel, K. (2009). Impact of normalization in distributed k-means clustering. *International Journal of Soft Computing*, 4(4):168–172.

Wang, H., Shan, H., and Banerjee, A. (2011). Bayesian cluster ensembles. *Statistical Analysis and Data Mining*, 4(1):54–70.

Xu, R. and Wunsch, D. (2008). *Clustering*, volume 10. John Wiley & Sons.