Adaptive Cluster Ensemble Selection

Javad Azimi, Xiaoli Fern
Department of Electrical Engineering and Computer Science
Oregon State University
{Azimi, xfern}@eecs.oregonstate.edu

Abstract

Cluster ensembles generate a large number of different clustering solutions and combine them into a more robust and accurate consensus clustering. On forming the ensembles, the literature has suggested that higher diversity among ensemble members produces higher performance gain. In contrast, some studies also indicated that medium diversity leads to the best performing ensembles. Such contradicting observations suggest that different data, with varying characteristics, may require different treatments. We empirically investigate this issue by examining the behavior of cluster ensembles on benchmark data sets. This leads to a novel framework that selects ensemble members for each data set based on its own characteristics. Our framework first generates a diverse set of solutions and combines them into a consensus partition P*. Based on the diversity between the ensemble members and P*, a subset of ensemble members is selected and combined to obtain the final output. We evaluate the proposed method on benchmark data sets and the results show that the proposed method can significantly improve the clustering performance, often by a substantial margin. In some cases, we were able to produce final solutions that significantly outperform even the best ensemble members.

1 Introduction

A fundamental challenge in clustering is that different clustering results can be obtained using different clustering algorithms and it is difficult to choose an appropriate algorithm given a data set. Cluster ensembles address this issue by generating a large set of clustering results and then combining them using a consensus function to create a final clustering that is considered to encompass all of the information contained in the ensemble. Existing research on cluster ensembles has suggested that the diversity among ensemble members is a key ingredient for the success of cluster ensembles [Fern and Brodley, 2003], noting that higher diversity among ensemble members tends to produce higher performance gain. In contrast, some studies have also indicated that a medium level of diversity is preferable and leads to the best performing ensembles [Hadjitodorov *et al.*, 2006]. Such seemingly contradicting observations can be explained by the fact that each data set has its own characteristics and may require a distinct treatment. A few recent studies have investigated the question of how to design or select a good cluster ensemble using diversity-related heuristics [Hadjitodorov *et al.*, 2006; Fern and Lin, 2008]. While it has been shown that cluster ensemble performance can be improved by the proposed heuristics, they are designed to be universally applicable for all data sets. This is problematic as different data sets pose different challenges, and it is likely that such differences require different strategies for selection. This motivates our work reported in this paper. In particular, based on our investigation on cluster ensembles' behavior using a set of four *training data sets*, we propose to form an ensemble based on the characteristics of the given data set so that the resulting ensemble is best suited for that particular data set.

In particular, we first generate an ensemble Π, which contains a diverse set of solutions, and then aggregate Π into a single partition P* using a consensus function. Different from traditional methods, we do not output P* as the final solution. Instead, we use P* to gain understanding of the ensemble Π. Specifically, we measure the difference between the ensemble members and the consensus partition P* to categorize the given data set into a *stable* or *non-stable* category. Our experiments on the four training data sets indicated clear differences between these two categories, which necessitates a different treatment for each category. Accordingly, our method selects a special range of ensemble members based on the categorization to form the final ensemble and produce the consensus clustering. We empirically validate our method using six *testing* data sets. The results demonstrate that by adaptively selecting the ensemble members, our method significantly improves the cluster ensemble performance. We further compare to a state-of-the-art ensemble selection method and our approach achieved highly competitive results, and demonstrated significant benefit for data sets in the *non-stable* category.

2 Background and Related Works

Below we review the basic steps in clustering ensembles and some recent developments on cluster ensemble design.

2.1 Ensemble Generation

It is commonly accepted that for cluster ensembles to work well the member partitions need to be different from one another. Many different strategies have been used to generate the initial partitions for a cluster ensemble. Examples include: (1) using different clustering algorithms to produce the initial partitions [e.g., Strehl and Ghosh, 2003]; (2) changing initialization or other parameters of a clustering algorithm [e.g., Fern and Brodley, 2004]; (3) Using different features via feature extraction for clustering [e.g., Fern and Brodley, 2003]; and (4) partitioning different subsets of the original data [e.g., Strehl and Ghosh, 2003].

2.2 Consensus Function

Once a set of initial partitions are generated, a consensus function is used to combine them and produce a final partition. This has been a highly active research area and numerous consensus functions have been developed. We group them into the following categories: (1) Graph based methods, [Strehl and Ghosh, 2003, Fern and Brodley, 2004]; (2) relabeling based approaches [Dudoit and Fridlyand, 2003]; (3) Feature-based approaches [Topchy *et al.*, 2003]; and 4) Co-association based methods [Fred and Jain, 2000].

Note that here we do not focus on ensemble generation or consensus functions. Instead, we assume that we are given an existing ensemble (and a consensus function), and investigate how to select a subset from the given ensemble to improve the final clustering performance.

2.3 Diversity and Ensemble Selection

Existing research revealed that the diversity among the ensemble members is a vital ingredient for achieving improved clustering performance [Fern and Brodley, 2003]. In this section we will first review how diversity is defined and then discuss some recent developments on using diversity to design cluster ensembles.

Diversity Measures. Existing literatures have devised a number of different ways to measure the diversity of ensemble members [Hadjitodorov *et al.*, 2006]. Most of them are based on label matching between two partitions. In essence, we deem two partitions to be diverse if the labels of one partition do not match well with the labels of the other. Two measures commonly used in the literature are the *Adjusted Rand Index (ARI)* [Hubert and Arabie, 1985] and the *Normalized Mutual Information (NMI)* [Strehl and Ghosh, 2003]. Note that both measures can be used in our framework. We experimented with both measures in our investigation, and they produced comparable results. In this paper, we present results obtained using NMI as the diversity measure.

Ensemble Selection. After generating the initial partitions, most of the previous methods used all generated partitions for final clustering. This may not be the best because some ensemble members are less accurate than others and some may have detrimental effects on the final performance. Recently a few studies sought to use the concept of diversity to improve the design of cluster ensemble by selecting an ensemble from multiple ensembles [Hadjitodorov *et al.*, 2006], by selecting only a subset of partitions from a large library of clustering solutions [Fern and Lin 2008], or by assigning varying weights to different partitions [Li and Din, 2008].

Hadjitodorov *et al.* [2006] generate a large number of cluster ensembles as candidate ensembles for selection, and they rank all ensembles based on their diversity. They propose to choose ensembles with median diversity based on empirical evidence suggesting that such ensembles are often more accurate than others for data sets that were tested in their experiments.

Note that the above method is not directly comparable to our method because it requires generating a large number of candidate ensembles. In contrast, we assume that we are given an existing ensemble and try to select a subset from it, which is defined as the *cluster ensemble selection* problem by Fern and Lin [2008]. In their paper, Fern and Lin investigated a variety of heuristics for selecting subsets that consider both the diversity and quality of the ensemble members, among which the *Cluster and Select* method was empirically demonstrated to achieve the most robust performance. This method first clusters all ensemble members and then selects one solution from each cluster to form the final ensemble. In our experiments we will compare with this method and refer to it as CAS_FL.

Note that the above reviewed methods are fundamentally different from ours because they aim to design selection heuristics without considering the characteristics of the data sets and ensembles. In contrast, our goal is to select adaptively based on the behavior of the data set and ensemble itself.

3 Adaptive Ensemble Selection

In this section, we will first describe our initial investigation on four *training* data sets that informed our design choices.

3.1 Ensemble System Setup

Below we describe the ensemble system setup we used in our investigation. This includes how we generate the ensemble members, and the consensus function used to combine the partitions. Note that our proposed system is not limited to these choices; other methods can be used as well.

Ensemble Generation. Given a data set, we generate a cluster ensemble of size 200 using two different algorithms to explore the structure of the data. The first is K-means, which has been widely used in cluster ensemble research as a basis algorithm for generating initial partitions of the data due to its simplicity and its unstable nature when different initializations are used.

In addition to K-means, we also introduce a new clustering algorithm, named Maximal Similar Features (MSF), for producing the ensemble members. This algorithm is chosen because one of our companion investigations (unpublished) has shown that MSF works well together with K-means for generating diverse cluster ensembles. In particular, when these two algorithms are used together, the resulting ensembles tend to outperform those generated by K-means or MSF alone. Below we describe the MSF algorithm.

MSF works in an iterative fashion that is highly similar to K-means. In particular, it begins with an initial random assignment of data points into k clusters, where k is a pre-specified parameter. After the initial assignment, the algorithm iteratively goes through the re-estimation step (i.e., re-estimate the cluster centers) and the re-assignment step (i.e., re-assign data points to their most appropriate clusters).

In MSF, the center re-estimation step is exactly the same as K-means, which simply computes the mean of all data points in the same cluster. The critical difference comes from the re-assignment step. Recall that in K-means, to reassign a data point to a cluster, we compute its Euclidean distances to all cluster centers and assign it to the closest cluster. In contrast, MSF considers each feature dimension one by one, and for each feature it assigns a data point to its closest center. Note that different features may vote for the data point to be assigned to different clusters and MSF assigns it to the cluster that has the most votes, or in other words, has the Maximal Similar Features.

Consensus Function. To combine the initial partitions, we choose a popular co-association matrix based method that applies standard hierarchical agglomerative clustering with average linkage (HAC-AL) [Fisher and Buhmann, 2003; Fern and Brodley, 2003] as the consensus function.

While one might suspect that the choice of consensus function will play an important role in the performances that we achieve, our initial investigation using an alternative consensus function introduced by Topchy *et al.* [2003] suggested that our results were robust to the choice of the consensus function.

3.2 Ensemble Performance versus Diversity

We apply the above described cluster ensemble system to four benchmark data sets from the UCI repository: Iris, Soybean, Thyroid and Wine [Blake and Merz].

For each data set, we generate an ensemble of size 200 $\Pi=\{P_1, P_2, \ldots, P_{200}\}$, using K-means and MSF. For each of the 200 partitions, *K*, the number clusters, is set to be a random number drawn between 2 and 2**C*, where C is the total number of known classes in the data. We then apply HAC-AL to the co-association matrix to produce a consensus partition P* of the data, where *K*, the number of clusters, is *C*.

In attempt to understand the behavior of the cluster ensembles, we examined the diversity between the ensemble members and the consensus partition P^*. In particular, we compute the NMI values between P_i and P^*, for i=1, …, 200. Inspecting these NMI values, we found that the four data sets demonstrate drastically different behavior that can be roughly grouped into two categories. The first category contained the Iris and Soybean data sets, for which majority of the ensemble members were quite similar to P^* (NMI values >0.5). In contrast, the other two data sets showed an opposite trend. We will refer to the first category as the *stable* category to reflect the belief that the structure of the data set is relatively stable such that most of the ensemble members are similar to one another. The second category is referred to as *non-stable.* In this case, the final consensus partition, which can be viewed as obtained by averaging the ensemble members, is dissimilar to the members. This fact suggests that the ensemble contains a set of highly different clustering solutions. In this case, we can argue that the clustering structure of the data is unstable. The distinction between the two categories can be easily seen from Table 1, which shows the average NMI values for the four data sets computed as described above. In column 3, we show the number of ensemble members that are similar to P^* (with NMI > 0.5).

Table 1. The diversity of ensemble members with regards to P* and the data set categorization

Name	Average NMI	# of ensemble with NMI >0.5	Class
Iris	0.693	197	S
Soybean	0.676	179	S
Wine	0.471	85	NS
Thyroid	0.437	61	NS

See Figure 1 for a more complete view of the distribution of the NMI values for the four data sets. In particular, for each data set it shows a histogram for the NMI values. The x-axis shows the NMI values and the y-axis shows the number of ensemble members at that particular NMI value.

This suggests that we can classify an ensemble into one of the two categories, *Stable(S)* or *Non-stable (NS)*, based on the diversity (as measured by NMI) between ensemble members and the final consensus partition. In particular, we classify an ensemble as *stable* if its average NMI values between the ensemble members and P* is greater than τ=0.5. Alternatively, one can also classify an ensemble as *stable* if more than 50% of its ensemble members have NMI (with P*) values larger than τ=0.5.

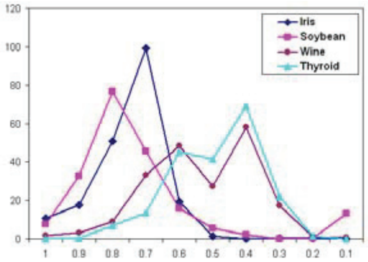


Figure 1. The distribution of ensemble members diversity with regards to P*.

Note that in our experiments, the categorization of a data set is highly stable from run to run and also appears to be not sensitive to the exact choice of τ as long as it is within a reasonable margin (e.g., [0.48-0.52]). Further, we expect this margin to increase as we increase the ensemble size.

We conjectured that the *stable* category will require a different treatment from the *non-stable* category in ensemble selection design. To verify this conjecture, we devised four simple subsets of the ensemble members, according to their NMI values with P*. In particular, given a cluster ensemble Π, and its consensus partition P*, we first sort all ensemble members according to their NMI with P* in decreasing order. We then define four subsets of interest as 1) all ensemble members (**F**ull); 2) the first half of the ensemble members (**L**ow diversity to P*); 3) the second half of the ensemble members (**H**igh diversity from P*); 4) the medium half of the ensemble members (**M**).

In Table 2, we see that our conjecture was confirmed for these data sets. In particular, we see that for the *stable* data sets, the first two options (**F** and **L**) work the best, whereas for the *non-stable* data sets, the third option (**H**), which contains ensemble members that are highly different from P*, works the best.

Table 2. The performance of 4 different subsets.

Name	1st (F)	2nd (L)	3rd (H)	4th (M)	Category
Iris	**0.744**	**0.744**	0.640	0.725	S
Soybean	**1**	**1**	0.557	0.709	S
Thyroid	0.257	0.223	**0.656**	0.325	NS
Wine	0.474	0.376	**0.680**	0.494	NS

Here we offer some possible explanations for the observed behavior. For the *stable* data sets, we suspect that the ensemble members generally reveal similar structures, and the differences mainly come from the slight variance introduced by the clustering procedure. In this case, using **F** is expected to be the best option because variance reduction can be maximized. On the other hand, by selecting **H** for the non-stable data sets, we essentially select high diversity solutions. Conceptually, if we map all clustering solutions in the ensemble into points in some high dimensional space, P* can be viewed as their centroid. By selecting **H** for the non-stable data sets, we choose the outmost quartile of points (solutions), i.e., these solutions that are most diverse from one another. Our results suggested that high diversity is desirable for the *non-stable* data sets. This is consistent with previous literature where high diversity was shown to be beneficial [Fern and Brodley, 2003]. One possible explanation is that in such cases the differences among ensemble members may be originated from different biases of the clustering procedure. To achieve the most bias correction, we need to include a set of most diverse solution by selecting subset **H**. An alternative explanation is that because most ensemble members are dissimilar to P*, it can be argued that the P* is not an appropriate result and selecting the most dissimilar ensemble members to P* (**H**) may lead to better results. We can see some supporting evidence for this claim in our experimental results, especially in Figure 3 of Section 4.4.

3.3 Proposed Framework

Given a data set, the proposed framework works as follows.

- Generate an ensemble Π of different partitions.
- Obtain consensus partition P* by applying a consensus function.
- Compute NMI between ensemble members and P* and rank the ensemble members based on the NMI values in decreasing order.
- If the average NMI values > 0.5, classify the ensemble as *stable* and output P*.
- Otherwise, classify the ensemble as *non-stable* and select subset **H** (the most dissimilar subset from the P*) and apply a consensus function to this subset, and output the consensus partition.

4 Experimental Results

Below we first describe the data sets used in the experiments and the basic experiment set up.

4.1 Experimental Setup

Our method was designed based on empirical evidence on four data sets. We consider these data sets as our training sets. To test the general applicability of our method, we need to use a new collection of data sets for testing. Toward this goal, we perform experiments on six new data sets, including the Vehicle, Heart, Pima, Segmentation, and Glass data sets from UCI machine learning repository and a real world data set O8X from image processing [Gose *et al.,* 1996].

As described in Section 3.1, we generate our cluster ensembles with 100 independent k-means runs and 100 independent MSF runs, each with a randomly chosen clustering number *K,* forming ensembles of size 200. The consensus function that we use is HAC-AL. Note that our initial experiments on different consensus functions suggested that our method is robust to the choice of consensus functions.

The reported results are the NMI values of the final consensus partitions with the known class labels. Note that the class labels are only used for evaluation purpose and not used in any part of the clustering procedure. Each value we report here is averaged across 100 independent runs.

4.2 Data Set Categorization

Recall that the first step of our framework is to generate an initial cluster ensemble and classify it into one of the categories based on the ensemble characteristics. In this section, we will present the categorization of each data set.

With the initial cluster ensemble and its resulting consensus partition P*, we compute the NMI value between each ensemble member and the consensus partition P*. The results are summarized in Table 3. In particular, the first column lists the name of each data set, and the second column provides the average NMI between ensemble members and P*. The third column demonstrates the number of ensemble members which have an NMI more than 0.5. The last column shows the categories to which the data set is assigned based on the NMI values.

Table 3.Categorization of the data sets

Name	Mean NMI	#members NMI >0.5	Class
Segmentation	0.602	169	S
Glass	0.589	131	S
Vehicle	0.670	199	S
Heart	0.241	11	NS
Pima	0.299	26	NS
O8X	0.488	91	NS

It can be seen that the Glass, Vehicle and Segmentation data sets are classified as *stable* data sets because their average NMI values are greater than 0.5. In contrast, the O8X, Heart and Pima data sets are classified as *non-stable* data sets. Note that if we use the alternative criterion of having more than half of ensemble members with an NMI more than 0.5, we obtain exactly the same results.

4.3 Selecting Subset

Once we classify a data set, we then move on to the ensemble selection stage and apply the strategy that is most appropriate for its category. For *stable* data sets, we keep the full ensemble and directly output the consensus partition P*. For *non-stable* data sets, we choose the **H** subset in the ensemble, i.e., the set that is most diverse from P*.

To test the effectiveness of this strategy, we evaluate all four subsets as presented in Section 3.2 and show the results in Table 4. The numbers shown here are the NMI values between the final partition and the ground truth, i.e., the class labels. In particular, the 2nd column provides the full ensemble results. The 3rd column records the performance of subset **L**, containing ensemble members that are similar to P*. The 4th column shows the clustering ensemble result of subset **H**, consisting of the members that are dissimilar to P*. The 5th column shows the results of subset **M**, containing the medium diversity members. For comparison purpose, we also show the performance of the best ensemble member in column six. Finally, the last column shows the categorization for each data set for reference.

The best performance for each data set is highlighted using bold face (the differences are statistically significant using paired t-test, $p<0.05$). The selected subset by our method for each data set is marked out with a '*' character. Note that the top four data sets (Iris, Soybean, Thyroid and Wine) are the *training* data sets used to develop our method and the rest are the *testing* data sets for validation of our method.

The first thing to note is that no single subset consistently performs the best for all six testing data sets. This confirms our belief that selecting a particular subset is not the best solution for all data sets.

Our proposed framework allows for flexible selection based on the characteristics of the given data set and ensemble. We can see that we were able to select the best performing subset for most of the cases. What is particularly interesting is that by selecting the ensemble members most different from P* for the *non-stable* data sets, we were able to achieve significant performance improvement in comparison to using the full ensemble (see O8X, Heart and Pima).

Table 4. The clustering ensemble results of 4 different subsets of ensemble members and the best ensemble member result.

Name	1st (F)	2nd (L)	3rd(H)	4th(M)	Best P	Data set Class
Iris	**0.744***	0.744	0.640	0.725	0.768	S
Soybean	**1***	1	0.557	0.709	0.978	S
Thyroid	0.257	0.223	**0.656***	0.325	0.471	NS
Wine	0.474	0.376	**0.680***	0.494	0.584	NS
O8X	0.491	0.444	**0.655***	0.582	0.637	NS
Glass	0.269*	**0.272**	0.263	0.269	0.397	S
Vehicle	**0.146***	0.141	0.119	0.136	0.227	S
Heart	0.095	0.079	**0.340***	0.104	0.169	NS
Pima	0.071	0.071	**0.127***	0.060	0.076	NS
Seg.	0.406*	0.379	0.390	**0.438**	0.577	S

The performance of our method is more striking when compared to the best performance among all ensemble members. Take the Heart data set for example; its ensemble members are highly inaccurate, suggesting a strong bias of the clustering procedure for this data set. We categorize Heart as *non-stable* and select subset **H**. This produced a final result substantially more accurate than even the best ensemble member. To our best knowledge, such significant improvement is rarely seen in the cluster ensemble literature, which typically compares the final ensemble performance with the average performance of all ensemble members.

Table 5. Comparing the proposed method with CAS_FL

Name	*Proposed method*	CAS_FL
Iris(S)	0.74	0.613
Soybean(S)	**1**	0.866
Thyroid(NS)	**0.656**	0.652
Wine(NS)	**0.680**	0.612
O8X(NS)	0.655	0.637
Glass(S)	0.269	**0.301**
Vehicle(S)	**0.146**	0.122
Heart(NS)	**0.340**	0.207
Pima(NS)	**0.127**	0.092
Seg.(S)	0.406	**0.550**

We further compared the proposed method with the state-of-the-art ensemble selection method, namely the CAS_FL method by Fern and Lin [2008]. The NMI values of the final partitions produced by both methods are presented in Table 5. From the table it can be seen that, our method is highly competitive compared to CAS_FL. In particular, it consistently outperformed CAS_FL on all ***non-stable*** data sets. For stable data sets, we notice that CAS_FL sometimes performed better, namely for the *Glass* and the *Segmentation* data sets. Note that among all *stable* data sets, these two data sets are the most unstable ones. This suggests that two categories may not be enough to characterize the differences among all data sets, and we may need to use a different selection strategy for data sets like *Glass* and *Segmentation.*

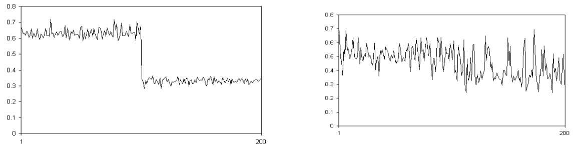
4.4 Discussion

In this section we seek possible explanations for the superior performance of our proposed method.

One interesting question is that is our selection method choosing one clustering algorithm over another for the *non-stable* data sets? We looked into this question by examining the selected ensemble members to see if they are generated by the same algorithm. The answer is: no, it depends. In particular, please see Figure 2 for two example *non-stable* data sets: wine and thyroid. The *x*-axis shows the indexes of the clustering solutions. We place all of the K-means clustering solutions together at position 1-100, whereas the MSF solutions are placed at position 101-200. The *y*-axis shows the NMI values of the solutions in relation to P*. For the Wine data set, because it was classified as a *non-stable* data set, our system selects subset **H**. From the figure we can see that the MSF solutions had lower NMI values, thus were selected over K-means. However, for the Thyroid data set, it was not a clear cut selection, suggesting that the proposed approach is more complex than selecting one method over another.

Note that we have also tested our method on ensembles generated using only the k-means algorithm and the proposed selection strategy still works well in comparison to other ensemble selection methods. However, using both algorithms generated more diverse ensemble members and achieved better final results than using K-means alone.

Figure 3 shows another set of results that may shed some lights on our performance improvement. The x-axis shows the ensemble member indexes and the y-axis shows the NMI values between the ensemble members and the ***real class labels*** (instead of P*). The ensemble members are ranked in decreasing order according to their NMI values with P*. This means that, the leftmost ensemble member is most similar to P*, and the right most ensemble member is most different from P* based on its NMI value with regards to P*.

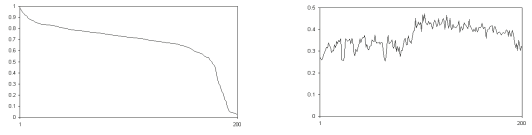


Wine Thyroid

Figure 2. The accuracy of k-means and MSF ensemble members with regards to real label values.

Figure 3 shows two representative data sets, one for each category. It can be seen that, for the *stable* category (Soybean), we observe a negative slope. This suggests that, for *stable* data sets, the NMI value between an ensemble member and P* is positively correlated with the NMI value between the ensemble member and the real label. Higher NMI values with P* implies higher NMI values with the real class label. This corroborates with our theory that for *stable* data sets the clustering procedure has limited or no bias and ensembles mainly work by reducing the variance. In such cases, it is not surprising that **F** (the full ensemble) performs the best because it achieves the maximum variance reduction.

In contrast, we observe an opposite trend for the *non-stable* data set, which showed negative correlation between the set of NMI values. By selecting subset **H**, our method was actually selecting the more accurate clustering solutions to form the ensemble, which may be the reason for the observed performance improvement for *non-stable* data sets. The strong contrast between the *stable* and *non-stable* data sets observed here confirms our fundamental hypothesis -- that is, different data sets require different treatment in ensemble design.



Soybean Thyroid

Figure 3. The NMI between ensemble members and the real label.

5 Conclusion

It is our belief that a truly intelligent clustering system should adapt its behavior based on the data set characteristics. To our best knowledge, there has not been any serious attempt at such a system. In this paper, we introduced an adaptive cluster ensemble selection framework as an initial step toward this direction. The framework starts by generating a diverse set of solutions and then combines them into a consensus partition P*. We introduce a simple heuristic based on the diversity between the ensemble members and the consensus partition P* to classify the given data set into the *stable* or *non-stable* category. Based on the categorization of the data set, we then select a special range of ensemble members to form the final ensemble and produce the final clustering. As a result, for different data sets, the selection strategy is different based on the feedback we obtain from the data in the original cluster ensemble. Experimental results demonstrate that by adaptively selecting the ensemble members, the proposed method can significantly improve the cluster ensemble performance, sometimes by a substantial margin (more than 200% for the Heart data set). In some cases, we were able to produce final solutions that significantly outperform even the best ensemble members.

6. References

[Blake and Merz] C. Blake and C. Merz. The UCI Machine Learning repository. www.ics.uci.edu/~mlearn/MLRepository.html.

[Dudoit and Fridlyand, 2003] S. Dudoit and J. Fridlyand. Bagging to improve the accuracy of a clustering procedure. *Journal of Bioinformatics 19*:1090- 1099, 2003.

[Fern and Brodley, 2003] X. Fern and C. Brodley. Random projection for high dimensional data clustering: a cluster ensemble approach. In *Proceedings of ICML* 2003, pages 186-193.

[Fern and Brodley, 2004] X. Fern and C. Brodley. Solving cluster ensemble problems by bipartite graph partitioning. In *Proceedings of ICML*, 2004.

[Fern and Lin, 2008] X. Fern and Wei Lin. Cluster Ensemble Selection. *Statistical Analysis and Data Mining*, 1(3):128-141, 2008.

[Fisher and Buhmann, 2003] B. Fischer and J.M. Buhmann. Bagging for path-based clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(11):1411- 1415, 2003.

[Fred and Jain, 2000] A. L. N. Fred and A. K. Jain. Data Clustering Using Evidence Accumulation. *In Proceedings of ICPR*, 2000, pages 276 – 280.

[Gose et al., 1996] E. Gose, R. Johnsbaugh and S. Jost. *Pattern Recognition and Image Analysis.* Prentice Hall, 1996.

[Hadjitodorov et al., 2006] S. Hadjitodorov, L. I. Kuncheva, and L. P. Todorova. Moderate Diversity for Better Cluster Ensembles. *Information Fusion Journal,* 7(3):264-275, 2006.

[Hubert and Arabie, 1985] L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, 1985.

[Strehl and Ghosh, 2003] A. Strehl and J. Ghosh. Cluster ensembles a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning*, 3: 583-617, 2003.

[Topchy et al., 2003] A. Topchy, A. K. Jain, and W. Punch. Combining Multiple Weak Clusterings. In Proceedings of *ICDM 2003*, pages 331-338.

[Topchy et al., 2004] A. Topchy, A. K. Jain and W. Punch. A mixture model for clustering ensembles. *In Proceedings of SDM* 2004, pages 379–390.

[Li and Ding, 2008] Tao Li, Chris Ding. Weighted Consensus Clustering. *In Proceedings of SDM* 2008, pages:798-809.