Cluster Ensemble Selection

Xiaoli Z. Fern and Wei Lin*

Abstract

This paper studies the ensemble selection problem for unsupervised learning. Given a large library of different clustering solutions, our goal is to select a subset of solutions to form a smaller yet better performing cluster ensemble than using all available solutions. We design our ensemble selection methods based on quality and diversity, the two factors that have been shown to influence cluster ensemble performance. Our investigation revealed that using quality or diversity alone may not consistently achieve improved performance. Based on our observations, we designed three different selection approaches that jointly consider these two factors. We empirically evaluated their performance in comparison with both full ensembles and a random selection strategy. Our results indicate that by explicitly considering both quality and diversity in ensemble selection, we can achieve statistically significant performance improvement over full ensembles.

1 Introduction

Clustering for unsupervised data exploration and analysis has been investigated for decades in the statistics, data mining and machine learning communities. The goal of clustering is to group similar objects together based on some notion of similarity. Over the years, many clustering algorithms have been developed, each utilizing different distance/similarity measures and/or objective functions. Applying different methods, or the same methods with different parameter choices to the same data, we can obtain varying clustering results. A fundamental question is: given so many possible options, how should we choose among them? One possible answer to this question is that we do not need to choose at all; because we can leverage these different options by applying all of them and then combining their clustering results. This is the basic philosophy behind cluster ensembles [20], which have gained increasing popularity in the clustering community [5, 6, 10, 11, 13, 21, 22, 23] in recent years.

A cluster ensemble framework typically produces a large set of clustering results and then combines them using a consensus function to create a final clustering that is considered to encompass all of the information contained in the ensemble. In practice, a cluster ensemble can be obtained in many different ways. Multiple clustering algorithms, different representations of the data, and different parameter choices can all be used to produce a diverse set of clustering solutions. It is common to produce hundreds or even more clustering solutions to form a single cluster ensemble. Traditionally, all of the available clustering solutions are combined together to produce the final consensus clustering. However, is it always the best to include all available solutions in the ensemble? Given a large library of clustering solutions, can we select the clustering solutions carefully so that we can actually do better than using the whole library? This is the question we investigate in this paper and we refer to it as the *cluster ensemble selection* problem following the practice of supervised ensemble learning [3].

Given a large library of clustering solutions, the goal of cluster ensemble selection is to choose a subset from the library to form a smaller cluster ensemble that performs as well as or better than using all available clustering solutions. Toward this goal, we investigate two properties that have been identified by existing research [5, 13, 11] as important factors for cluster ensembles to perform well: the quality and the diversity of the clustering solutions in the ensemble. We first consider ensemble selection based on quality and diversity respectively. The results indicate that: 1) it is often possible to select a smaller ensemble and achieve better performance than using the full ensemble; 2) while it is possible to do so, using quality or diversity alone can not reliably achieve this goal.

Based on these results, we propose three ensemble selection approaches that jointly consider quality and diversity in selection. The first method proposes a joint objective function that combines both factors. The second method organizes different solutions into groups such that similar solutions are grouped together and then selects one quality solution from each group. The last method creates a scatter plot of points, where each point corresponds to a pair of clustering solutions represented by their average quality and diversity, and then uses the convex hull of all points to select solutions.

*School of Electrical Engineering and Computer Science
Oregon State University, Corvallis, OR, USA 97331
{xfern, linwe}@eecs.oregonstate.edu

We empirically compare our methods with the full ensemble. Our evaluation results suggest that by explicitly considering quality and diversity together, our methods were able to achieve statistically significant performance improvements over the full ensembles. We further evaluated a random selection strategy, which failed to achieve statistically significant improvements. This confirms that the performance improvements we see is not due to chance. Empirical sensitivity analysis verifies the robustness of the proposed methods with respect to the choice of libraries and the outlier (degenerate) solutions in the library.

The remainder of the paper is organized as follows. In Section 2, we will review the related literature. Section 3 presents the basic selection strategies based on quality and diversity alone and their performance is evaluated in Section 4. Section 5 and 6 present the improved selection strategies and their empirical evaluations in comparison with the full ensemble and a random strategy. In Section 7, we conduct sensitivity analysis experiments. Finally, we summarize our contributions and conclude the paper in Section 8.

2 Related Work

The basic idea of combining different clustering solutions to obtain improved clustering has been explored under different names such as consensus classification/clustering [17, 16] and evidence accumulation [7]. The framework of cluster ensembles was recently formalized by Strehl and Ghosh [20]. Many different approaches for generating cluster ensembles have been proposed in the literature [7, 20, 21, 5, 16]. Representative examples include using different subsamples of the original data, using different subsets of the original features, using different random parameters such as the number of clusters and random initializations to the clustering algorithm, and using different clustering methods. To the best of our knowledge, however, all of these prior approaches utilize all of the generated ensemble members when combining them into a final consensus clustering. The only exception is the work by Hadjitodorov et al [11], where multiple cluster ensembles were generated and the ensemble with the median diversity was used to produce the final clustering. In contrast, our work seeks to select a small subset from a large given library to form the ensemble.

In supervised ensemble learning, it has been shown that by carefully selecting a subset of a large number of classifiers, one can achieve performance similar or even better than using all available classifiers [15, 3, 2]. For supervised ensemble learning, there are two main families of selection methods: one is based on the quality and diversity of the ensemble members, and the other is guided by cross-validated external objective functions (such as the prediction accuracy and the area under ROC curves). In unsupervised learning, cross-validation based methods are difficult to apply because we do not have any external objective function to optimize. Therefore, in this paper we focus on selection methods that are based on quality and diversity measures of the ensemble members.

3 Selection Based on Quality and Diversity

In supervised learning, quality and diversity are well defined concepts, where quality measures the accuracy of the ensemble members and diversity measures the difference in the predictions made by the ensemble members. For unsupervised learning, however, these concepts are not so clearly defined. In this section, we first explain how we measure the quality and diversity of clustering solutions. We then describe a simple selection strategy for each of the two measures.

3.1 Definitions: Quality and Diversity

Quality. For unsupervised clustering tasks, we do not have any external objective function such as accuracy to measure the quality of the clustering solutions. In clustering literature, it is common to use predefined class labels as a surrogate for the true underlying structure and then measure the quality of a clustering solution based on how well it recovers the class labels. This, however, cannot be used in our ensemble selection because supervised information such as class labels can not be included in the clustering process. Here we propose to use an internal quality measure based on an objective function introduced by Strehl and Ghosh for designing consensus functions [20]. In particular, given an ensemble E of r clustering solutions denoted by $E = \{C_1, C_2, \cdots, C_r\}$, Strehl and Ghosh sought to find a consensus clustering that maximizes the following criterion:

$$SNMI(C, E) = \sum_{i=1}^{r} NMI(C, C_i) \tag{3.1}$$

where $NMI(C, C_i)$ is the normalized mutual information between clustering C and C_i. If two clusterings define completely independent partitions, their expected NMI value is 0. In contrast, if two clustering defines the same partition of the data, the NMI value is maximized to be one. Here we refer to this objective function as the sum of $NMI(SNMI)$. Intuitively, a clustering C maximizing $SNMI$ maximizes the information it shares with all the clusterings in the ensemble, thus can be considered to best capture the general trend contained in the ensemble.

In our case, given a large library of clustering solutions $L = \{C_1, C_2, \cdots, C_r\}$ to select from, we use $SNMI(C_i, L)$ to measure the quality of each clustering solution C_i. Intuitively, this measures how well a particular clustering agrees with the general trend contained in L.

Diversity. There have been a number of different diversity measures proposed for cluster ensembles. Here we use the measure introduced by Fern and Brodley [5], which is based on pair-wise normalized mutual information among clustering solutions. In particular, we measure the pair-wise similarity of two clusterings as $NMI(C_i, C_j)$ and compute the sum of all pairwise similarities $\sum_{i \neq j, C_i, C_j \in E} NMI(C_i, C_j)$ within the ensemble as a measure of the ensemble diversity. The lower the value, the higher is the diversity.

We chose the above diversity measure because it has been shown to impact the cluster ensemble performance. Note that the selection methods we develop in this paper do not limit themselves to any particular diversity measure. Part of our future work is to experiment with other diversity measures proposed in the literature.

3.2 Simple selection strategies

Quality. As the first step of our investigation, we use the above defined quality measure to guide our selection and include only these solutions that are of high quality into the ensemble. In particular, given a large library of clustering solutions L, this strategy simply ranks all clustering solutions in L based on their qualities as measured by $SNMI(C, L)$ defined above and selects the top K solutions to include in the ensemble, where K is the desired ensemble size. Below we will refer to this strategy as *Quality.* Note that if a clustering has high SNMI value, conceptually it suggests that this solution has high consistency with the general trends shown by the overall library. Clustering solutions with low SNMI values, on the other hand, can be considered as "outliers" of the library and may be detrimental to be included in the ensemble. Generally, we expect the ensembles selected by "Quality" to have high redundancy in the chosen solutions.

Diversity In contrast, we also look at the selection strategy that seeks to maximize the ensemble diversity. This can be viewed as *a heaviest K-vertex subgraph* problem. In particular, the clustering solutions in the library are represented as vertices in a completely connected graph, and their pairwise diversity values (1-NMI) are assigned as the weights of the edges connecting the vertices. Selecting an ensemble of size K with maximum diversity can be achieved by finding a K-vertex subgraph whose edge weights are maximized, i.e., the heaviest K-vertex subgraph. However, this problem is known to be NP-hard [12]. Here we use a simple greedy strategy described as follows. We begin with an ensemble E containing the single solution of highest quality (as measured by SNMI).[1] It then incrementally selects one solution at a time from the library to add to E such that the resulting ensemble has the highest diversity, that is, the lowest value of $\sum_{i \neq j, C_i, C_j \in E} NMI(C_i, C_j)$. This process repeats until we reach the desired ensemble size K. Below we will refer to this strategy as *Diversity.*

In the literature, various heuristics have been suggested for generating diverse clustering solutions for cluster ensembles and it is commonly believed that diversifying the cluster ensemble has beneficial effect because mistakes made by different ensemble members may cancel each other out. The *Diversity* strategy described here follows this philosophy and explicitly searches for highly diverse subset from the library to form ensembles. Note that a potential problem with this method is that it may result in the inclusion of some low quality solutions into the ensemble.[2]

4 Preliminary Results

In this section, we examine the performance of the ensembles produced by the above described selection methods and compare them with the performance of the full ensembles. First we describe the data sets and the basic settings of our experiments that we use in the evaluation, including the library generation procedure, the choice of consensus functions, and the evaluation criterion.

4.1 Data sets and experimental setting

Data sets. Our experiments use both benchmark and real-world data sets. See Table 1 for the basic information about these data sets. Among them, CHART, SEGMENTATION, WINE and ISOLET6 (This is a 6-class subset of the original ISOLET data set, which contains 26 classes) are benchmark data sets from the UCI machine learning data repository [1]. We further included two real-world data sets in our evaluation. They are a content based image retrieval (CBIR) data set [4] and a EOS remote sensing data set which has been used

[1] Note that alternatively we can initialize the greedy search procedure with two solutions whose NMI value is the smallest among all pairs. This, however, does not produce qualitatively different results.

[2] Conceptually, a set of completely random clustering solutions will have the maximum diversity. However, they will not form good cluster ensembles. This is why we start out our greedy procedure by including the solution with the best quality measure.

for land cover type predictions [9]. Although these data sets are not very large, they do present significant challenges to standard clustering algorithms due to factors such as high dimensionality. The performance of standard algorithms like K-means (with or without ensemble) on these data sets leave ample room for improvement. That is why these data sets were chosen for the experiments.

Table 1: Basic information of the data sets

	#inst.	#features	#classes
CBIR	1545	183	8
CHART	600	60	6
EOS	2398	20	8
ISOLET6	1440	617	6
SEGMENTATION	2310	18	7
WINE	178	13	3

It should be noted that all six data sets are labeled and contain supervised class information. The class labels, however, were only used in evaluating the final clustering solutions and not used in any way during clustering or ensemble selection.

Generating the library. To build our clustering library, we used the K-means algorithm [14] as our base learner. K-means is chosen because it is one of the most widely used clustering algorithms and has been used in many previous cluster ensemble studies. In order to include a broad range of clustering solutions in our library, we used three different settings to generate clustering solutions.

K-means is an iterative algorithm that starts with an initial assignments of data points into random clusters and then refine the clusters to improve a squared error criterion. Different initial assignments will lead to different local optimal solutions. Our first setting uses this property and apply K-means with different random initializations to obtain different clustering solutions. In this setting, K-means has access to all of the features and the variations among clustering runs only come from different initializations. Therefore, the clustering solutions obtained in this setting are expected to be of relatively good quality but low in diversity.

In the second setting, different clustering solutions are obtained by applying K-means to different random feature subsets. Note that for each run, we select d features, where d is a number drawn randomly between 2 and half of the dimension of the original data.

Finally, we use different random linear projections of the features to create different clustering solutions. Similar to the second setting, we set d, i.e., the number of linear projections we produce, by randomly drawing a number between 2 and half of the original dimension.

Following the common practice for cluster ensembles, we further employ some heuristics to diversify the clustering solutions in the library. In particular, in all three settings, for each clustering run we set k, the number of clusters for that run, by randomly drawing a number between 2 and $2 \times c$, where c is the number of classes in the data[3]. Each of the above three settings is used to generate 200 clustering solutions, resulting a collection of 600 models, which we then use as the library to select from. For each data set, we repeat this process ten times to generate ten libraries and all reported results are averaged across these ten runs.

It should be noted that we did not focus on generating optimal libraries — our choices, including the base clustering algorithm and the diversifying heuristics, are not necessarily optimized but do provide us with a set of representative libraries. Later we will present some further experiments to investigate different libraries.

Consensus function. Once a cluster ensemble is formed via selection, we need a consensus function to combine the selected solutions to produce a final consensus clustering. Many consensus functions have been proposed in the literature. We experimented with a number of popular approaches including the CSPA approach [20], the HBGF method [6], and the hierarchical agglomerative approach based on co-association matrix [8]. Different consensus functions obtained qualitatively similar results in terms of how different methods relate to each other. Therefore we will focus on the CSPA method and present only results obtained using CSPA as the consensus function. Below we briefly describe the CSPA method.

CSPA stands for Cluster-based Similarity Partitioning Algorithm. As suggested by its name, CSPA builds a similarity matrix based on the clustering solutions in the ensemble, which measures for each pair of data points the frequency of them being clustered together in the ensemble. This sometimes is also referred to as the co-association matrix. A graph partitioning algorithm is then applied to the similarity matrix to obtain a final clustering solution. Here we apply spectral graph partitioning [18] to produce a final partition of the data points into c clusters, where c is the number of known classes in the data. For more details of the CSPA method and spectral clustering, please refer to [20] and [18].

Evaluation criterion. To evaluate the final performance of the selected ensembles, we use the known

[3] When this information is not available, a good rule of thumb is to set the upper bound to be $\sqrt{n}$ [7].

class labels as a surrogate for the true underlying structure of the data and measure the normalized mutual information (NMI) [20] between the final consensus cluster labels and the class labels. Note that if the two labels are independent from each other, the expected NMI value is zero. The best NMI value is 1, which is attained when the class and cluster labels define exactly the same partition of the data. In general, the higher the NMI value, the better is the quality.

4.2 Results

We apply the *Quality* and *Diversity* selection strategies to form ensembles of size ten, twenty, and so on, up to 200. Once an ensemble is selected, the CSPA method is applied to obtain a consensus clustering solution, whose NMI value is then computed using the class label information. In Figure 1, we plot the NMI values of both selection methods as a function of the ensemble size. Also plotted is the full ensemble performance, obtained by applying CSPA to the full library. Note that each point in the graph is obtained by averaging the results of ten independent runs (libraries).

We first note from Figure 1 that for all data sets, it is possible to improve the performance over the full ensemble by selecting a smaller subset of solutions. In some cases, significant improvements can be obtained as demonstrated by the WINE data.

We notice that when quality is used to guide the selection, the resulting ensembles achieve competitive performance early on when the ensemble size is small. As we increase the ensemble size, the performance either level off quickly (see CHART, EOS, ISOLET6) or become unstable and/or worse (see CBIR, SEGMENTATION and WINE). This suggests that selecting only solutions that have good quality can be beneficial when the ensemble size is small. As we increase the ensemble size, because the selected good solutions may be highly similar to one another, it becomes unlikely to see performance improvement.

In contrast, we see a rather different trend for the *Diversity* strategy. Notably, with all but the WINE data set, we see relatively steady performance gain as we include more and more diverse solutions into the ensemble. However, the rate of improvement can be too slow sometimes for this strategy to outperform the full ensemble with a small subset of solutions. For example, for the CBIR, ISOLET6 and SEGMENTATION data sets, we see the diversity method failed to create small ensembles that outperform the full ensemble even when the ensemble size is increased to 200.

The contrasting behavior of these two methods suggest that in order to reliably select a good subset of solutions, quality and diversity should be considered jointly. In next section, we develop three different selection strategies to achieve this goal.

5 Joint Consideration of Quality and Diversity

Intuitively, an ensemble should work the best when its clustering solutions are of good quality and at the same time differ from one another significantly. The trade off between quality and diversity is the key design choice that we need to make for effective ensemble selection. In this study, we investigate a number of different ways to address this trade off. Below we describe the three methods that we develop to jointly consider quality and diversity for ensemble selection.

Joint criterion The trade-off between quality and diversity can be viewed in a multi-objective optimization framework [19], which seeks to optimize two or more conflicting objective functions. An intuitive and popular approach for solving multi-objective problems is to use a single aggregated objective function (AOF), which we adopt in our study. In particular, to build an ensemble of size K, we select K clustering solutions from the library that optimize the following AOF:

$$\alpha \sum_{i=1,\cdots,K} SNMI(C_i, L) + (1-\alpha) \sum_{i\neq j} (1 - NMI(C_i, C_j)) \tag{5.2}$$

where the first component summarizes the quality of the selected clustering solutions and the second component measures their pair-wise diversity. The parameter α controls how much emphasis we put on each objective. Note that optimizing the above AOF is also an NP-hard problem. This can be easily shown by noticing that the diversity maximization problem (the heaviest k-vertex subgraph problem) is a special case of this problem.

To perform selection using this AOF criterion, we use a greedy procedure similar to what was used in *Diversity* as described in Section 3.2. In particular, we start with the ensemble containing the single highest-quality solution and incrementally add one solution at a time to the ensemble to maximize the proposed objective function. For the remainder of this paper, we will refer to this method as *Joint Criterion* (JC). In our experiments, we set α to 0.5 because there is no clear reason to favor either one without knowing the specifics of the data. Later we will examine different choices for α to investigate its sensitivity.

Cluster and select. In our second method, we consider each clustering solution in the library as an entity and examine how they relate to each other. Despite the fact that we used numerous diversifying heuristics in generating our library, it is still quite likely to have clustering solutions that are highly similar to one another. If two clustering solutions C_1 and C_2 are similar and C_1 has been included in the ensemble, it is intuitive to not include C_2 to avoid redundancy even though C_2 might have good quality as well. However, the *Joint Criterion* method does not necessarily achieve this. Consider the situation where the existing ensemble contains a large number of clustering solutions that are highly different from C_2, being similar to C_1 will not prevent C_2 from being selected.

One way to address the above issue is to explicitly remove possible redundancies by grouping the clustering solutions into similar groups and selecting only one clustering solution from each group. Specifically, to form a cluster ensemble of size K, the library of solutions will be partitioned into K groups. Each group contains a set of solutions that are considered to be similar to one another. We then simply select one solution from each group to form the ensemble. To take quality (as measured by SNMI) into consideration, we select the solution with the highest quality from each group. Note that when K is set to be the size of the library, this method degrades to using full library. When K is set to 1, it is equivalent to choosing the solution with the highest quality.

There are many possible ways to partition the clustering solutions. Here we apply spectral clustering [18] to the pair-wise NMI matrix, which in essence can be considered as a similarity matrix describing the relationship among clustering solutions. We refer to this method as *Cluster and Select* (CAS).

Convex Hull The last method was inspired by the Kappa-Error Convex-hull pruning method of Margineantu and Dietterich for pruning classifiers generated by AdaBoost [15]. This method works as follows. First, we produce a quality-diversity diagram for the given library. The quality-diversity diagram is a scatter-plot where each point corresponds to a pair of clustering solutions in the library. Given a library of size n, we will produce a scatter plot of $n \times (n-1)/2$ points. Consider a point corresponding to solution pair C_i and C_j, its x coordinate is simply the value of $NMI(C_i, C_j)$ (e.g., $1-$diversity). Its y coordinate is the average of C_i's and C_j's SNMI values (e.g., average quality). This diagram visually depicts the diversity and quality level of a given library and it has been successfully used in previous literature for analyzing the impact of diversity and quality on the final cluster ensemble performance [5]. We leverage the information contained in this diagram and create a succinct summary of the entire diagram using the convex hull of all the points in the diagram. These points will include both the solutions with the highest quality and the most diverse pair of solutions. We form an ensemble by including all the clustering solutions that appeared in a solution-pair corresponding to a point on the convex hull. Note that some clustering solutions may appear multiple times on the convex hull. In such cases, we only select them once in the ensemble to avoid redundancy. Different from the previous two methods, the ensemble size here is automatically determined and can not be adjusted freely. Below we will refer to it as the *Convex Hull* (CH) method.

6 Experimental Evaluation

In this section, we evaluate the proposed ensemble selection methods by comparing their performance with the full ensembles. We use the same data sets and the same basic settings for experiments as described in Section 4.1. For each data set, we generate ten libraries; each library contains six hundred clustering solutions. To evaluate an ensemble selection method or base line method, we apply it to each of the ten libraries. Each resulting ensemble is then combined using the CSPA consensus function to produce the final consensus clustering. The consensus clustering is then evaluated against known class labels using the NMI measure. The reported final results are obtained by averaging across ten independent runs. In Figure 2 we plot the performance of our methods as a function of the ensemble size. Note that for the *Convex Hull* method and the full ensemble, the ensemble sizes are fixed, therefore the performance are shown as flat lines.

To provide more information about each method and the variance of their performance, Tables 2 - 7 report the NMI values of each method for ensemble sizes 30, 60, 90, 120 and 150 together with the NMI values of the full ensembles. In addition, we also report the performance of a random selection strategy, which forms ensembles by selecting randomly from the library [4]. As described earlier, each number reported here is the average of ten runs. For the proposed selection methods and the random method, we compare each of their results with the full ensemble and highlight those results that are better than full ensemble at a statistically significant level ($p < 0.05$, paired t-test) in bold face. Below we discuss the performance of each individual method based on the results shown in Figure 2 and Tables 2- 7.

6.1 Joint Criterion

Comparing with full ensembles, the *Joint Criterion* method achieved comparable or improved performance in most of the data sets. In particular, it achieved statistically significant improvement for the CHART,

[4] The random selection method was not shown in the figures because its performance tend to bounce around and make it difficult to read the figures.

Table 2: Results for CBIR

size	30	60	90	120	150	ConvexHull	Full
Joint	0.310	0.319	0.303	0.297	0.294		
Criterion	(0.017)	(0.017)	(0.027)	(0.024)	(0.021)		
Cluster and	**0.325**	0.308	0.310	**0.323**	0.311	**0.341**	0.308
Select	(0.023)	(0.029)	(0.030)	(0.026)	(0.029)	(.007)	(.026)
Random	0.294	0.304	0.301	0.306	0.299		
	(0.027)	(0.031)	(0.030)	(0.029)	(0.030)		

Table 3: Results for CHART

size	30	60	90	120	150	ConvexHull	Full
Joint	0.737	0.739	**0.777**	**0.778**	**0.779**		
Criterion	(0.033)	(0.033)	(0.006)	(0.008)	(0.003)		
Cluster and	0.747	0.742	0.738	0.742	0.750	0.734	0.735
Select	(0.039)	(0.039)	(0.041)	(0.037)	(0.035)	(.028)	(.036)
Random	0.731	0.731	0.730	0.743	0.742		
	(0.039)	(0.038)	(0.039)	(0.038)	(0.040)		

Table 4: Results for EOS

size	30	60	90	120	150	ConvexHull	Full
Joint	0.295	**0.304**	**0.306**	**0.300**	**0.300**		
Criterion	(0.014)	(0.017)	(0.019)	(0.015)	(0.006)		
Cluster and	0.290	**0.294**	**0.297**	**0.295**	**0.296**	0.277	0.287
Select	(0.012)	(0.006)	(0.005)	(0.003)	(0.004)	(.009)	(.003)
Random	0.284	0.290	0.289	0.289	0.287		
	(0.009)	(0.005)	(0.007)	(0.004)	(0.003)		

Table 5: Results for ISOLET6

size	30	60	90	120	150	ConvexHull	Full
Joint	0.819	0.811	0.813	0.816	0.816		
Criterion	(0.029)	(0.005)	(0.003)	(0.003)	(0.003)		
Cluster and	**0.850**	0.849	**0.850**	**0.851**	**0.851**	0.806	0.838
Select	(0.001)	(0.002)	(0.002)	(0.002)	(0.002)	(.048)	(.016)
Random	0.797	0.797	0.822	0.822	0.832		
	(0.052)	(0.052)	(0.041)	(0.041)	(0.033)		

Table 6: Results for SEGMENTATION

size	30	60	90	120	150	ConvexHull	Full
Joint	0.576	0.583	0.582	0.578	0.577		
Criterion	(0.049)	(0.026)	(0.008)	(0.013)	(0.009)		
Cluster and	0.597	**0.603**	0.596	0.597	0.595	0.584	0.576
Select	(0.015)	(0.012)	(0.027)	(0.013)	(0.018)	(.040)	(.030)
Random	0.567	0.566	0.561	0.563	0.574		
	(0.035)	(0.035)	(0.045)	(0.027)	(0.027)		

Table 7: Results for WINE

size	30	60	90	120	150	ConvexHull	Full
Joint	**0.458**	**0.458**	0.458	0.457	**0.459**		
Criterion	(0.005)	(0.004)	(0.007)	(0.006)	(0.006)		
Cluster and	**0.620**	**0.701**	**0.730**	**0.696**	**0.627**	0.429	0.447
Select	(0.124)	(0.079)	(0.119)	(0.079)	(0.082)	(.021)	(.013)
Random	0.432	0.429	0.438	0.433	0.429		
	(0.019)	(0.013)	(0.013)	(0.015)	(0.015)		

EOS and WINE data sets. Note that the performance trend of this method is very similar to that of selecting using diversity alone for most of the data sets, especially for large ensemble sizes, with the only exception of the WINE data set. (See Figure 1) This similarity indicates that our joint objective function places a rather heavy weight on diversity, especially for large ensemble sizes. The influence of quality is more prominent with small ensemble sizes, producing a better and more stable performance for small sizes than using diversity alone.

We also observe that the performance of this method typically levels off before the ensemble size reaches one hundred (most times much earlier), suggesting that this method is more appropriate for selecting small ensembles.

Sensitivity analysis of α. Note that so far we have set $\alpha = 0.5$ in our experiments. Here we would like to examine how sensitive this method is to the choice of α. We experimented with a variety of α values including $0.1, 0.2, ..., 0.9$ and compare their results with *Quality* ($\alpha = 1$) and *Diversity* ($\alpha = 0$). As one might expect, we observed that smaller values of α result in performance that are similar to *Diversity*, whereas larger α values led to performance similar to *Quality*. We further observe that setting α to particularly high or low values can be beneficial for some cases, but in general the performance is more robust when α is around 0.5. Here we will focus on the results of a small set of α values in the middle range $\alpha = 0.4, 0.5, 0.6$ and show in Figure 3 their performance together with *Diversity*($\alpha = 0$) and *Quality* ($\alpha = 1$).[5] From this figure we can see that the *Joint Criterion* method is relatively stable with respect to the α values tested, especially for large ensemble sizes.

6.2 Cluster and Select

Examining the results, we observe that *Cluster and Select* was able to achieve statistically significant improvements over full ensembles for five out of six data sets and it never degraded the performance. Particularly striking is the Wine data set, where we see drastic performance improvements across a wide range of ensemble sizes.

Interestingly, for the Wine data set, we observe that increasing the ensemble size first improves the performance, then started to hurt the performance once it went beyond 100. This is possibly because the library does not contain many distinct groups in the clustering solutions and forcing such grouping may have caused the performance to degrade. This suggests that a possible way to improve this method is to automatically decide how many groups to partition the solutions into based on the evidence from the data using techniques such as the EigenGap [18].

6.3 Convex Hull

The method that we adopted from the supervised learning community did not live up to its expectation. It failed to achieve significant improvement for all but the CBIR data set. Further it incurred a significant loss for the ISOLET6 data set. We conjecture that this is because the convex hull of the scatter plot often contains points that are on the extreme end and may actually be outliers. These points may well correspond to clustering solutions that are of both low quality and diversity, resulting in suboptimal ensembles. This suggests that an alternative approach to use the quality-diversity diagram is to explicitly search for those points that are located in the high-diversity and high-quality quadrant of the diagram and avoid outlier points.

6.4 Random strategy

The Random selection strategy is included in this evaluation to ensure that the performance improvement we observe with our proposed methods can not be achieved by chance. Our results confirm this because the random selection method did not significantly improve over full ensemble in any of the data sets across different ensemble sizes. It is interesting to note that for the CHART and EOS data sets, the random selection method performed respectably well for ensemble size as small as 30. This suggests that there exists large amount of redundancy in the libraries. Due to such redundancy, we can expect strategies favoring diversity to work well for these data sets. This is consistent with our experimental results, where the *Joint Criterion* method achieved the best performance for these two data sets.

6.5 Comparison across methods

Comparing the three proposed methods and the base line systems, we see that both *Joint Criterion* and *Cluster and Select* achieved promising performance toward our goal, that is to select smaller and better performing ensembles. The random selection method provided a good reference point to confirm that the performance improvements are not created by chance. In particular, the *Cluster and Select* method achieved the best overall performance and statistically significantly improved over full ensembles for all but one data set. Further examination of this method reveals that this method attains a good compromise between the *Quality* method and the *Diversity* method. We believe this is because this strategy explicitly seeks to remove redundant solutions and retain quality solutions at the same time.

[5] Note that these figures were generated using the extended libraries as described in Section 7.

7 Sensitivity Analysis

In our previous section, we have identified *Cluster and Select* as the best performing method. In this section, we will focus on this method and conduct a set of additional experiments to investigate the sensitivity of this method to the choice of library by varying the library and considering outlier (degenerate) solutions in the library.

7.1 Extending the libraries

To gain insights about how the size and design of the library impact the proposed selection strategies, we extended our library by adding 400 solutions that are generated using the following two different settings.

First, we adopt a recently proposed method by Caruana et al. [2] for generating diverse clustering solutions using K-means with random feature weighing. In particular, we assign each feature a random weight, which is used to scale the feature values. For example, if a feature is assigned a weigh of 10, for each data point it value for this feature is multiplied by 10. In effect, features with larger weights are considered more important in computing the distance function for K-means. The Zipf power law distribution is used to generate the random weights. Given an integer number Z, the Zipf distribution generates random integer numbers from 1 to Z such that the probability of the number i is proportional to $\frac{1}{i^\alpha}$, where α is the shape parameter. When alpha is zero, this is simply a uniform distribution from 1 to Z. As α increases, the probability of generating large numbers decreases. We use this setting to create 200 clustering solutions, each with a random α value choosen from the set $\{0.00, 0.25, 0.50, 0.75, 1.00, 1.25, 1.50\}$, and a random k value chosen from 2 to $2 \times c$, where c is the number of classes.

Finally, we also consider an additional clustering algorithm as our base learner, the spectral clustering algorithm [18],which takes a graph partitioning perspective for data clustering and has been shown to be highly effective in many applications where K-means fails. Spectral clustering requires as input a similarity matrix S describing the pairwise relationship among all data points. We use a Gaussian Kernel to compute S such that $S(i,j) = \exp(\frac{-||X_i - X_j||^2}{\sigma^2})$, where σ is the kernel width parameter. Different σ values result in different similarity matrices, hence different clustering results. For a thorough exploration, we consider a set of different kernel widths computed as $\sigma = \frac{\max ||X_i - X_j||}{2^{\frac{\beta}{8}}}$, for $\beta = 0, \ldots, 64$. This setting is again used to create 200 solutions, each with a random k value chosen between 2 and $2 \times c$, and a random kernel width chosen by selecting a random β value between 0 and 64.

Adding this additional 400 solutions to our original library, we obtain libraries of 1000 solutions. Note that the new libraries generated as such not only contain a larger number of solutions (for testing the impact of library size) but also contain a set of qualitatively different solutions due to the inclusion of spectral clustering (for testing the impact of library quality).

7.2 Considering outlier solutions

Interestingly for three of our data sets (namely CBIR, CHART, and ISOLET6) some of the clustering results we obtained using the above described procedure are degenerate solutions where the vast majority of the data points are assigned to one cluster. This presents a perfect opportunity to test whether our methods are sensitive to the presence of such outlier solutions in the library. To achieve this goal, for each of these three data sets, we created two set of different experiments, one with the full library (with degenerate outliers included) and another set with the outlier-free libraries that are created by the following procedure.

In particular, we remove 10% of the solutions whose cluster distributions have the lowest normalized entropy (small entropy indicates uneven distribution among clusters). More specifically, we consider each clustering solution as a multi-nomial random variable with possible values ranging from 1 to k, where k is the number of clusters. We compute its empirical entropy and normalize it to the zero-one range by the maximum entropy achievable $log_2(k)$. For each library (for CBIR, CHART, and ISOLET6), we rank all of 1000 clustering solutions based on this normalized entropy in increasing order and remove the first 100 solutions. Note that the libraries of the other three data sets were not processed because there were no degenerate solutions.

7.3 Results

In Figure 4, we show the performance of the *Cluster and Select* (CAS) method under different settings for each data set, including the original small library (CAS-S), the extended outlier-free library (CAS-L), and the extended library with outliers included (CAS-LO) (for the CBIR, CHART and ISOLET6 data sets). Also shown are the full ensemble performance for both the small libraries (Full-S) and the extended outlier free libraries (Full-L).[6]

We first noticed that the extended library led to improved full ensemble performance for three data sets (CBIR, CHART, and ISOLET6), and no difference or slightly reduced performance for the other data sets.

[6] We omitted Full-LO, i.e., full ensemble with large library and outliers because it is highly similar to Full-L.

For all data sets, however, the relationship between the CAS and Full remained the same, i.e., CAS can often improve the performance and never causes detrimental effect except for very small ensemble sizes. This suggests that CAS is relatively robust with regard to the library size and composition. On the other hand, we observe that applying CAS to the extended library sometimes resulted in worse performance in comparison to the smaller library, even though we were selecting from a larger pool of solutions. The most striking example is the WINE data set. However, this is not that surprising considering the difference between Full-L and Full-S. In particular, the inclusion of the additional 400 solutions was not beneficial, suggesting that these solutions are not useful toward improving the ensemble performance. By adding these 400 solutions, it actually becomes more difficult to select the good subset of solutions since the selection pool is now larger and more "diluted".

What is also interesting is that we observed almost no negative effect on CAS when the degenerate outlier solutions are included in the library, except for when the ensemble size is very small. This indicates that the proposed method is highly robust to the (small amount of) degenerate solutions in the library.

8 Conclusions

In this paper, we make the following contributions.

First, we defined the cluster ensemble selection problem. Given a large library of clustering solutions, the goal is to select a subset of solutions to form a small ensemble that achieves better performance than using all available solutions. While the ensemble selection problem has been studied in the supervised setting, our work is the first investigation in the unsupervised domain.

Second, we proposed and examined three different selection strategies that jointly consider the quality and diversity of the clustering solutions. Among them, we identified the *Cluster and Select* method as the best performing method and consider it highly promising toward our goal. In particular, in our experiments it achieved statistically significant improvements over full ensembles for five out of six data sets. Our experimental evaluation of the random selection strategy further confirmed that such performance improvements can not be obtained by chance. We further examined the sensitivity of this method to the choice of different libraries and to outlier solutions in the library. The results suggested that it is robust with respect to both factors.

In this study, we chose to use SNMI to measure the quality of a clustering solution and use NMI to measure pair-wise diversity. It should be noted that the methods we developed are not restricted to these particular choices. They can be easily replaced by other quality and diversity measures, which will be part of our future work. Another future direction is to revise the *Cluster and Select* method to automatically determine the number of groups into which we should partition the library, which will enable us to choose the most appropriate ensemble size.

References

[1] C. L. Blake and C. J. Merz. UCI repository of machine learning databases, 1998.

[2] R. Caruana, A. Munson, and A. Niculescu-Mizil. Getting the most out of ensemble selection. In *Proceedings of the Sixth international Conference on Data Mining*, 2006.

[3] R. Caruana, A. Niculescu-Mizil, G. Crew, and A. Ksikes. Ensemble selection from libraries of models. In *Proceedings of the twenty first International Conference on Machine Learning*, 2004.

[4] J. Dy, C. E. Brodley, A. Kak, C. Shyu, and L. S. Broderick. The customized queries approach to CBIR using EM. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 400–406, 1999.

[5] X. Z. Fern and C. E. Brodley. Random projection for high dimensional data clustering: A cluster ensemble approach. In *Proceedings of the Twentieth International Conference on Machine Learning*, pages 186–193, 2003.

[6] X. Z. Fern and C. E. Brodley. Solving cluster ensemble problems by bipartite graph partitioning. In *Proceedings of the Twenty First International Conference on Machine Learning*, pages 281–288, 2004.

[7] A. L. N. Fred and A. K. Jain. Data clustering using evidence accumulation. In *Proceedings of International Conference on Pattern Recognition*, 2002.

[8] Ana L. N. Fred and Anil K. Jain. Combining multiple clusterings using evidence accumulation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(6):835–850, 2005.

[9] M. Friedl, D. McIver, J. Hodges, X. Zhang, D. Muchoney, A. Strahler, C. Woodcock, S. Gopal, A. Schneider, A. Cooper, A. Baccini, F. Gao, and C. Schaaf. Global land cover mapping from MODIS: algorithms and early results. *Remote Sensing of Environment*, 83:287–302, 2002.

[10] D. Greene, A. Tsymbal, N. Bolshakova, and P. Cunningham. Ensemble clustering in medical diagnostics. In *Proc. 17th IEEE Symp. on Computer-Based Medical Systems*, 2004.

[11] S. T. Hadjitodorov, L. I. Kuncheva, and L. P. Todorova. Moderate diversity for better cluster ensembles. *Information Fusion*, 7(3):264–275, 2006.

[12] G. Kortsarz and D. Peleg. On choosing a dense subgraph. In *In Proceedings of the 3th Annual IEEE Symposium on Foundations of Computer Science*, pages 692–701, 1993.

[13] L. Kuncheva and S.T. Hadjitodorov. Using diversity in cluster ensembles. In *Proceedings of IEEE Int. Conf. on Systems, Man and Cybernetics*, 2004.

[14] J. MacQueen. Some methods for classifications and analysis of multivariate observations. In *Proc. Symp. Mathematical Statistics and Probability, 5th*, 1967.

[15] D. Margineantu and T. Dietterich. Pruning adaptive boosting. In *Proc. 14th International Conference on Machine Learning*, pages 211–218, 1997.

[16] S. Monti, P. Tamayo, J. Mesirov, and T. Golub. Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data. *Machine Learning*, 52:91–118, 2003.

[17] D. A. Neumann and V. T. Norton. Clustering and isolation in the consensus problem for partitions. *Journal of Classification*, 3:281–298, 1986.

[18] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, pages 849–856, 2002.

[19] Y. Sawaragi, H. Nakayama, and T. Tanino. Theory of multiobjective optimization. *Mathematics in Science and Engineering*, 176, 1985.

[20] A. Strehl and J. Ghosh. Cluster ensembles - a knowledge reuse framework for combining multiple partitions. *Machine Learning Research*, 3:583–417, 2002.

[21] A. Topchy, A. K. Jain, and W. Punch. Combining multiple weak clusterings. In *Proceedings IEEE International Conference on Data Mining*, pages 331–338, 2003.

[22] A. Topchy, A. K. Jain, and W. Punch. A mixture model for clustering ensembles. In *Proceedings of SIAM Conference on Data Mining*, pages 379–390, 2004.

[23] A. Topchy, A.K. Jain, and W. Punch. Clustering ensembles: Models of consensus and weak partition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1866–1881, 2005.

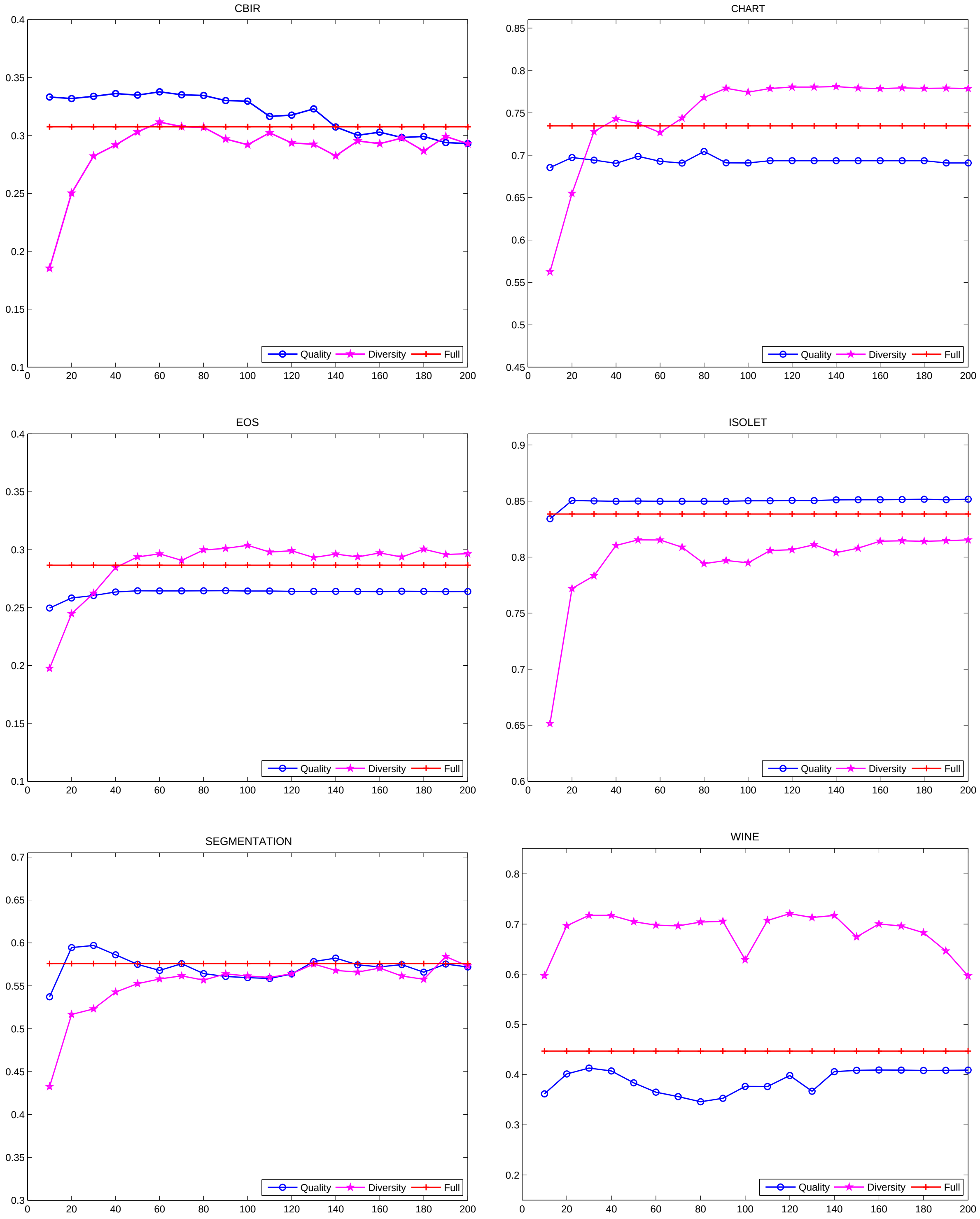

Figure 1: Comparing the *Quality* and *Diversity* selection methods with the full ensembles

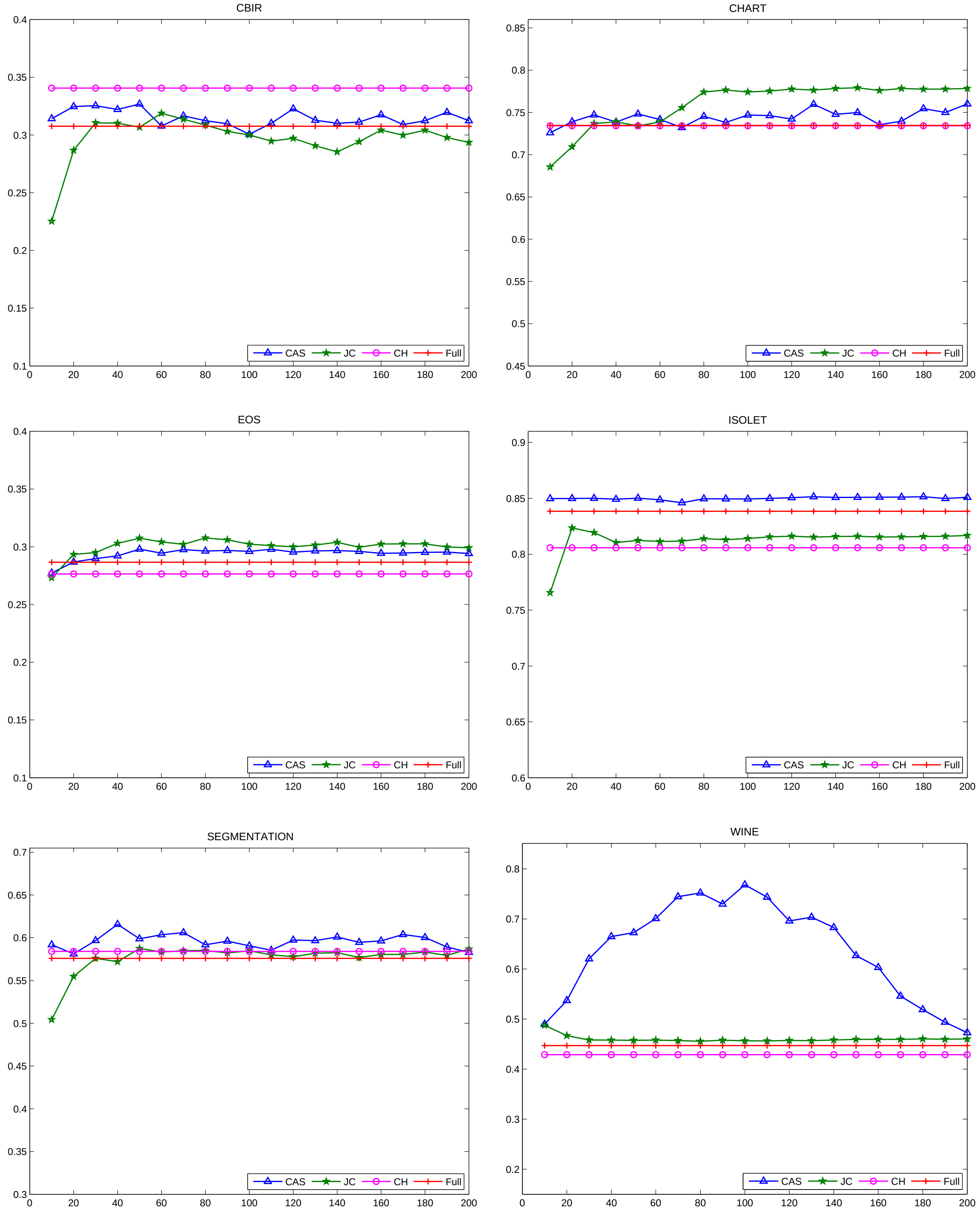

Figure 2: Performance comparison of the "Joint Criterion" (JC), "Cluster and Select" (CAS), "Convex Hull" (CH) methods and Full ensembles

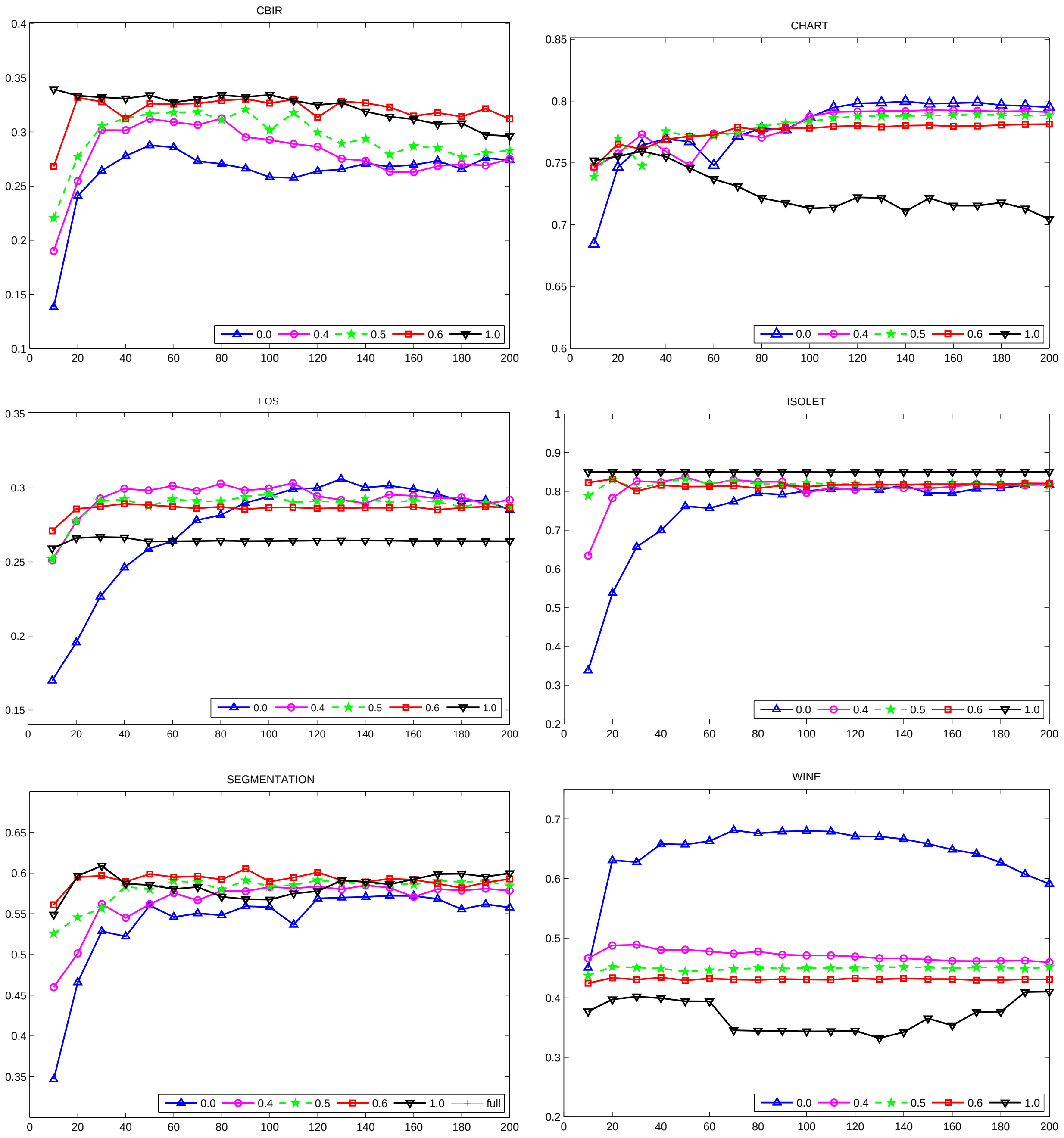

Figure 3: Sensitivity analysis of the "Joint Criterion" (JC) method by varying the values for α.

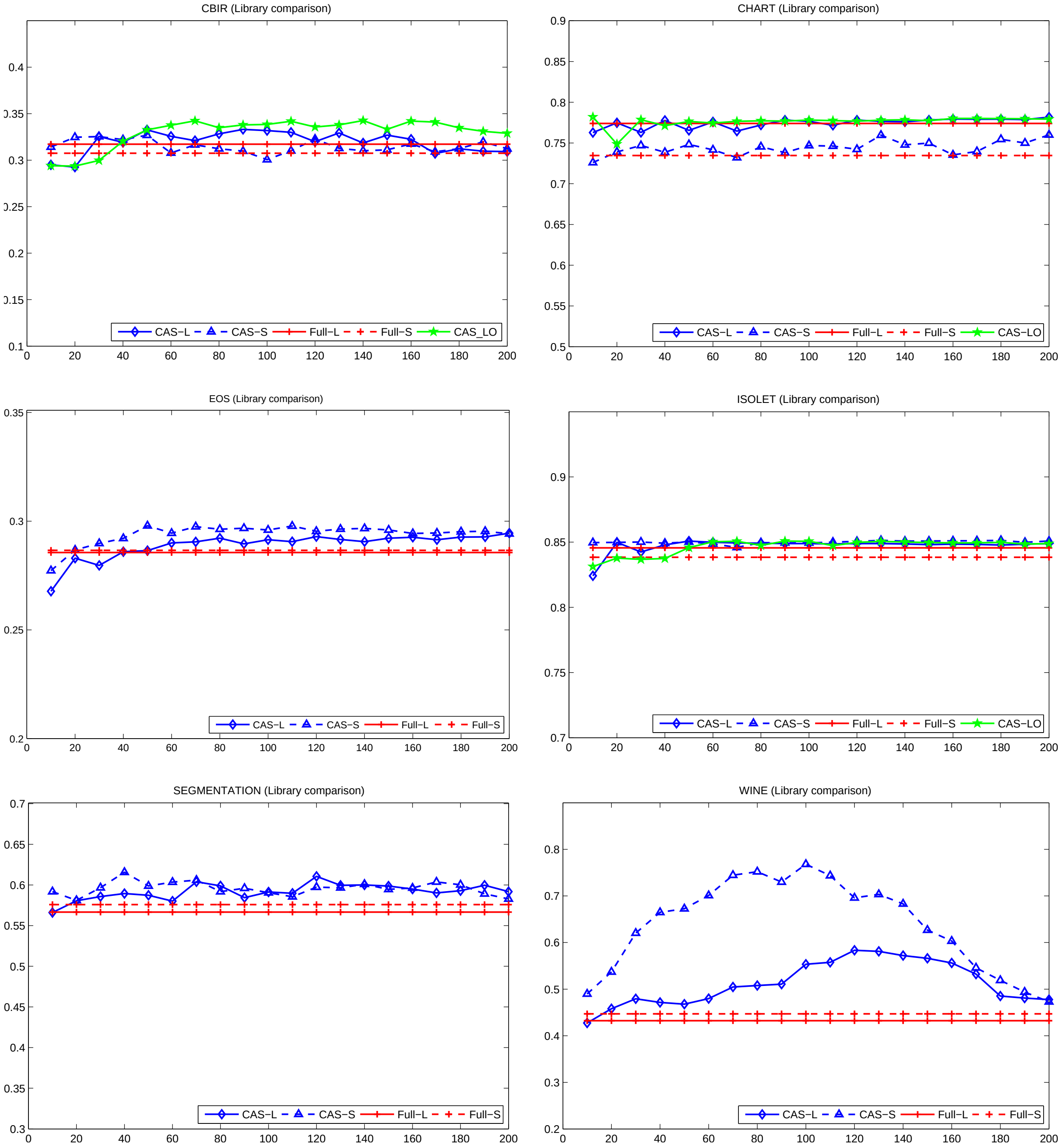

Figure 4: Sensitivity analysis of the “Cluster and Select” (CAS) method considering different libraries