Journal of Intelligent & Fuzzy Systems 19 (2008) 321–334
IOS Press

Improving supervised learning performance by using fuzzy clustering method to select training data

Donghai Guan, Weiwei Yuan, Young-Koo Lee*, Andrey Gavrilov and Sungyoung Lee
Department of Computer Engineering, Kyung Hee University, Korea

Abstract. The crucial issue in many classification applications is how to achieve the best possible classifier with a limited number of labeled data for training. Training data selection is one method which addresses this issue by selecting the most informative data for training. In this work, we propose three data selection mechanisms based on fuzzy clustering method: center-based selection, border-based selection and hybrid selection. Center-based selection selects the samples with high degree of membership in each cluster as training data. Border-based selection selects the samples around the border between clusters. Hybrid selection is the combination of center-based selection and border-based selection. Compared with existing work, our methods do not require much computational effort. Moreover, they are independent with respect to the supervised learning algorithms and initial labeled data. We use fuzzy c-means to implement our data selection mechanisms. The effects of them are empirically studied on a set of UCI data sets. Experimental results indicate that, compared with random selection, hybrid selection can effectively enhance the learning performance in all the data sets, center-based selection shows better performance in certain data sets, border-based selection does not show significant improvement.

Keywords: Classification, data selection, fuzzy clustering, center-based selection, border-based selection, hybrid selection

1. Introduction

Supervised learning is one primary sub-field of classical machine learning. In supervised learning, we are provided with a collection of labeled (preclassified) patterns. And the problem is to label a newly encountered, yet unlabeled, pattern. Typically, the given labeled (training) patterns are used to learn the descriptions of classes which in turn are used to label a new pattern [1].

Usually supervised learning works well only when we have enough training samples. Unfortunately, in many real-world applications, the number of labeled data available for training purpose is limited. This is because labeled data are often difficult, expensive, or time consuming to obtain as they require the efforts of experienced human annotators [2]. For instance, if one is building a speech recognizer, it is easy enough to get raw speech samples, but labeling even one of these samples is a tedious process in which a human must examine the speech signal and carefully segment it into phonemes. Another example is Web page classification in which unlabeled samples are readily available, but labeled ones are fairly expensive to obtain. In these applications, the crucial issue is how to achieve the best possible classifier with a small number of labeled data.

An important topic addressing above issue is selecting the valuable data to label, considering that labeling data is a costly job. This topic is known as active learning [3,4]. In active learning, the learning process iteratively queries unlabeled samples to select the most informative samples to annotate and update its learned models. Therefore, the unnecessary and redundant annotation is avoided.

This paper proposes three new active learning methods based on fuzzy clustering method. Our methods first partition the given unlabeled samples into clusters

*Corresponding author. Tel.: +82 31 201 3732; E-mail: yklee@khu.ac.kr.

and then select the most representative ones from each cluster to label. Our proposed data selection methods are center-based selection (CS), border-based selection (BS) and hybrid selection (HS). In CS, the data with high degree of membership in each cluster are selected. Center-based selection is named because the selected data samples are usually close to the cluster centers. BS selects training samples around the borders between clusters and HS is a hybrid selection method combining CS and BS.

The heuristic of CS, BS and HS is similar with some existing works. In [3], the authors choose the valuable training samples which are closest to the current classification boundary. The intuitive ideas of closest-to-boundary criterion and our BS are similar. The difference is that BS tries to find those samples using clustering information instead of classification. Some other existing works [13,30] put more emphasis on the representative samples, which are basic idea of our CS method.

Compared with existing methods, the proposed fuzzy cluster-based methods usually require much less computational effort. In addition, they are independent with the supervised learning algorithms and the initial labeled data for training purpose. In particular, they can work even in the case when there is no labeled data available.

This paper studies empirically the effects of our three data selection methods for supervised learning. All the data selection methods are implemented by using the fuzzy c-means algorithm. Eleven UCI data sets were used in the empirical study. We regard the performance of random selection (RS) as the baseline and compare it with that of CS, BS and HS. Experimental results clearly indicate that HS outperforms RS in all the selected datasets. While, CS shows better performance as compared to RS in certain datasets. On the other hand, the BS strategy fails to show any significant improvement over the RS technique.

The rest of this paper is organized as follows. In Section 2, related work is presented. Section 3 presents our proposed three data selection mechanisms (center-based selection, border-based selection and hybrid selection) in details. Section 4 reports on the empirical study and discusses some observations. Section 5 discloses conclusions and future work.

2. Related work

In many classification applications, we cannot get enough labeled data for training. And the crucial issue is how to achieve the best possible classifier using the limited number of training data.

Semi-supervised learning [5] is a method aiming to address above issue. In addition to labeled samples, unlabeled ones are exploited in semi-supervised learning to improve learning performance. Many existing semi-supervised learning algorithms use a generative model for the classifier and employ Expectation-Maximization (EM) to model the label estimation or parameter estimation process [6]. For example, mixtures of Gaussians [7], mixture of experts [8], and naive Bayes [9] have been used as the generative model respectively, while EM is used to combine labeled and unlabeled data for classification.

In addition to semi-supervised learning, another important method to address above issue is selecting the valuable data to label, which is known as active learning [3,4]. In this paper we focus our attention on active learning.

For a data set $D = \{x_1, x_2, \ldots, x_n\} \subset R^d$, let D_l represent the labeled set in which every sample is given a label and $D_u = D - D_l$. Most active learning systems comprise two parts: a learning engine and a selection engine. The learning engine uses a supervised learning algorithm to train a classifier on D_l at every iteration. The selection engine then selects a sample from D_u and requests a human expert to label the sample before passing it to the learning engine. The goal of active learning is to achieve the best possible classifier within a reasonable number of calls for labeling by human help.

Existing work on active learning can be characterized by the learning algorithms used by learning engine, which include multilayer perceptrons [10], combination of naive Bayes and logistic regression [3], Support Vector Machine (SVM) [11–13] and so on.

The central part in active learning is data selection strategy since learning algorithm is just a tool to implement active learning process. Most existing work has concentrated on two strategies: certainty-based and committee-based selection. In the certainty-based strategy, an initial system is trained using D_l [14–17]. Then the system labels the samples in D_u and determines the certainties of its predictions of them. The sample with the lowest certainty is then selected and presented to the experts for annotation. In the committee-based methods, a distinct set of classifiers is created using D_l [18–21]. The sample in D_u, whose label differs most when presented to different classifiers are presented to the experts for annotation. In both paradigms, a new system is trained using the new set of

annotated examples, and this process is repeated until it reaches the predefined rounds or some stopping criteria are satisfied.

There are several drawbacks in certainty-based and committee-based selection. First of all, data selection is based on the classification result obtained by using classifier (or classifiers) to classify the unlabeled samples. Therefore, the classifier/classifiers should make sense; otherwise data selection process is disturbed. Considering the first iteration of data selection, classifier is trained on D_l. To get qualified classifier/classifiers, the number of samples in D_l cannot be zero or very small value. However, in real-world applications, it is quite possible that the number of initial labeled data is small or even zero. In this case, most existing data selection methods, including certainty-based and committee-based methods, can not work. Secondly, most data selection methods require much computational effort. The reason is that many iterations are needed and each iteration only selects one single sample. Note that one iteration requires one training-classification (for certainty-based selection) or multiple training-classification processes (for committee-based selection).

To solve above problems, we propose new data selection methods based on fuzzy clustering. Our method first partitions the given unlabeled samples into clusters and then selects the most representative ones from each cluster to label. Actually, using clustering in data selection is not new and several work has been done [22–24]. However, they use clustering only at the initial/preprocessing stage. Then supervised learning methods, Learning Vector Quantization [22], k-nearest neighbor [23] and regularized logistic regression [24] are needed for data selection. Different with them, we aim to give an empirically study on whether clustering solely (without supervised learning) could be used for data selection through appropriate data selection methods. It should be noted that because there is no any supervision used by our methods, our methods might give bad results to those data sets of which the underlying structure cannot be found by fuzzy clustering. In this case, traditional active learning can be used.

Since our methods are based on fuzzy clustering instead of supervised methods, there is no any requirement on the size of initial labeled samples. In particular, the best case to use our methods is when there are no (or very small number) pre-labeled samples available so that traditional supervise-based selection methods cannot be used. Moreover, our methods reduce the computational complexity by selecting a batch of unlabeled samples instead of one sample. Note that for existing works [25–30], even a batch of samples are selected at each iteration, still more computational effort is needed compared with ours. The reason is that the existing work are based on classifiers which requires training process, while our method is based on clustering which does not require training process.

In fact, batch samples selection is not only useful on saving computational effort, but also more convenient for the human experts. Human experts always tend to give the labels more precisely for batch samples than single sample, because they can compare different examples and refine the assigned labels.

3. Our proposed data selection mechanisms

3.1. A brief introduction to fuzzy c-means

Fuzzy c-means clustering (FCM) [31] is a popular data clustering algorithm which combines K-means clustering with fuzzy logic. As with fuzzy sets [32], using FCM, each data point can be a member of more than one cluster with different degrees of membership function between 0 and 1. FCM is an objective function based clustering method. Here objective function measures the overall dissimilarity within clusters. By minimizing the objective function we can obtain the optimal partition. Let $X = \{x_1, x_2, \ldots, x_n\}$ denote the measured data set. The FCM objective function J is defined as:

$$J = \sum_{i=1}^{n} \sum_{j=1}^{c} (u_{ij})^m \left\| x_i - v_j \right\|^2 \tag{1}$$

Clustering of FCM is carried out through an iterative minimization of J according to the following steps:

S1: Choose fuzzy factor (m), number of clusters (c) and $c$initial cluster centers v_j.

REPEAT

S2: At iteration t, compute u_{ij} with v_j by

$$u_{ij} = \left(\sum_{k=1}^{c} \frac{\left\| x_i - v_j \right\|^{2/(m-1)}}{\left\| x_i - v_k \right\|^{2/(m-1)}} \right)^{-1} \tag{2}$$

S3: Update v_j with u_{ij}, by

$$v_j = \frac{\sum_{i=1}^{n} u_{ij}^m \cdot x_i}{\sum_{i=1}^{n} u_{ij}^m} \tag{3}$$

UNTIL ($\|V_t - V_{t-1}\| \leq \varepsilon$, V_t and V_{t-1} denote the vector of clusters centers at iteration t and $t-1$ respectively, ε is convergence criterion with $0 < \varepsilon < 1$)

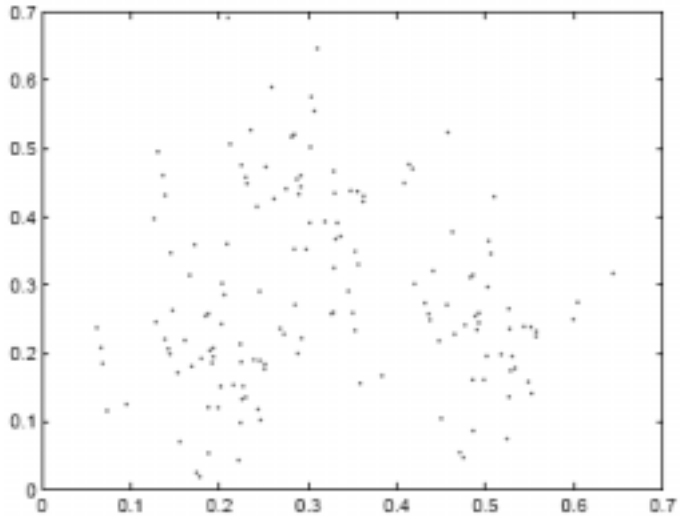


Fig. 1. Artificial data set.

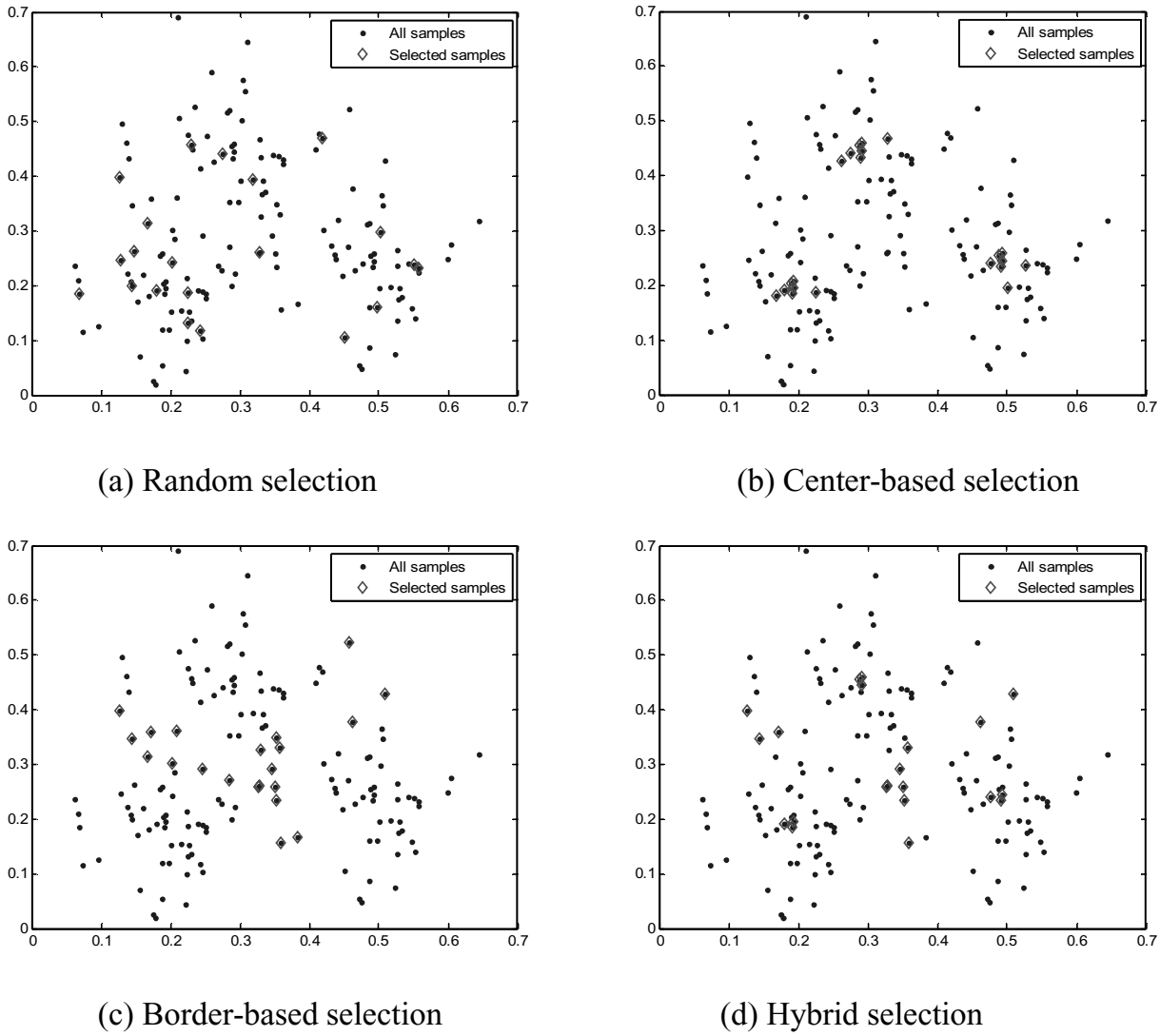
Here u_{ij} is the degree of membership of x_i in cluster j and m is the fuzzy factor that determines the degree of fuzziness ($m > 1$). As m approaches one, fuzziness degrades and the FCM algorithm approaches to the standard K-means algorithm. $V = \{v_1, v_2, \ldots, v_c\}$ is the vector of cluster centers. $\|x_i - v_j\|^2$ is any norm expressing the similarity between the measured data x_i and the center v_j.

3.2. Our training data selection mechanisms

Fuzzy c-means computes the cluster centers and generates the class membership matrix U. We proposed three data selection mechanisms in this paper, and all of them are based on U. To visually see the difference between them, an artificial data set is used. As shown in Fig. 1, this data set includes 150 2-D samples and 3 classes. 21 samples will be selected as training data.

(1) **Center-based selection**: This selection strategy assumes that the samples with high degree of membership in each cluster are more valuable and representative for learning. We extract these samples through analyzing membership matrix $U_{n\times m}$. Here n is the number of samples partitioned and m is the number of clusters. u_{ij} is the element at the i^{th} row and j^{th} column in U. In each cluster $j(j = 1 : m)$, if $i^* = \arg\max_{i=1:n} u_{ij}$, then sample x_i is regarded as the most representative sample in this cluster and selected firstly. The next selected sample is $x_{i^{**}}$ with $i^{**} = \arg\max_{i=1:n, i\neq i^*} u_{ij}$. In turn, other samples in cluster j will be selected using this way, until the number of data equals to k_j (the number of training data allocated to cluster j). Usually in active learning, we are given the total number of training data $K(K = \sum_{j=1}^{m} k_j)$ instead of k_j, so how to determine k_j with the knowledge of K is an issue in center-based selection. To avoid imbalance problem in learning, a simple and effective way we adopted is to select the same or similar number of samples from each cluster $k_j \cong \frac{K}{m}$. This method is sufficient for our purpose since it provides for a basic level of exploring the effect of samples obtained from center-based selection. It could be improved if needed since the detailed information of each cluster, such as sample distribution and size, is not considered. Based on CS, if we select 21 training samples from above artificial data set (7 samples each cluster), the result of selection is shown in Fig. 2(b).

(2) **Border-based selection**: This selection strategy assumes that the samples located at the borders between clusters are more representative. Here we say a sample is located at the border between clusters when its two high degrees of membership are very similar. For example, a data set comprises three clusters. For a sample of it, when its degrees of membership for each cluster is [0.5, 0.49, 0.01], its two high membership de-



(a) Random selection (b) Center-based selection

(c) Border-based selection (d) Hybrid selection

Fig. 2. Training data selected by different mechanisms.

grees (0.5 and 0.49) are very similar. In this case, we say that this sample is located at the border between cluster 1 and 2. Membership matrix $U_{n\times m}$ is also used in this part. Here n is the number of samples partitioned and m is the number of clusters. For each sample $x_i (i = 1 : n)$, if $j^* = \arg\max_{j=1:m} u_{ij}$ and $j^{**} = \arg\max_{j=1:m, j\neq j^*} u_{ij}$, then T_i ($T_i = u_{ij^*} - u_{ij^{**}}$) is calculated. Finally sample x_{i^*} with $i^* = \arg\min_{i=1:n} T_i$ is regarded as the most representative sample. In turn $x_{i^{**}}$ with $i^{**} = \arg\min_{i=1:n, i\neq i^*} T_i$ is the next valuable data to be selected. Other samples will be selected using this way until the number of selected reaches the limitation. The training data selected using the border-based selection are shown in Fig. 2(c).

(3) **Hybrid selection**: This strategy is a hybrid selection method combining above two methods. It assumes that both the samples from CS and BS are representative. Combining them might provide better result than either alone. For a data pool D, let K denote the number of data to be selected as training data. A simple combination scheme in this work is to select about $K/2$ samples from center-based selection and border-based selection respectively. Of course, it is not required to exactly follow this combination scheme in the real applications. For ex-

ample, if it is obvious that samples got from center-based selection are redundant, then more samples could be extracted from border-based selection. For above artificial data set, if 9 samples are selected by CS (3 samples each cluster) and 12 samples are selected by BS. Then the 21 training samples determined by HS are shown in Fig. 2(d).

4. Empirical study

4.1. Configuration

In this work, training samples are selected by analyzing membership matrix computed by fuzzy c-means. Fuzzy c-means is configured as follows: Fuzzy factor (m, in Section 3.1) is set to 2. Convergence criterion (ε, in Section 3.1) is set to 0.00001. Maximum iteration is set to 100. Euclidean distance is used as the similarity measure. This is the default configuration of the fuzzy c-means tool we used [33]. For the number of clusters for each data set, we set it equal to the number of classes since the number of classes is available in many applications.

To test the effect of training data selection mechanisms, a classifier is needed. In this empirical study, Multilayer Perceptron neural network with back propagation (BP) training algorithm is used. In all the experiments, the network with one hidden layer is adopted. TANSIG, LOGSIG activation functions are used in the hidden layer and output layer respectively. Let n_1, n_2, n_3 denote the number of input nodes, hidden nodes and output nodes respectively. In our experiments, n_1 is the number of attributes in each sample, in hidden layer $n_2 = 2 \times n_1 + 1$, n_3 is the number of classes. Consider the example of the iris data set. It contains four attributes and classifies them into three classes. In this case, a 4-9-3 network is used. Each network is trained to 100 epochs. Note that, since the relative instead of absolute performance of the proposed methods are concerned, the architecture and training process of the neural networks have not been finely tuned.

Eleven data sets from the UCI Machine Learning Repository [34] are used in the empirical study, where missing values on continuous attributes are set to the average value while those on binary or nominal attributes are set to the majority value. Information on these data sets is tabulated in Table 1.

In the experiment, we set the number of training data to some small values, because data selection mechanisms aim to improve learning performance in the case when training data is insufficient. To objectively compare the performance of our proposed three mechanisms on each data set, experiments are conducted with different numbers of training data. Consider the example of the iris dataset. As shown in Table 2, experiment on it consists of five parts with training number is set to 3, 6, 9, 15, 21 respectively. Let X denote the number of training data. In each part (For instance, when $X = 21$), Iris (denoted by D) is randomly partitioned into two sets: $D_{\text{test}}(|D_{\text{test}}| = 75)$ and $D_{\text{non-test}}(D_{\text{non-test}} = D \backslash D_{\text{test}})$.$D_{\text{test}}$ represents test set. Then 21 samples will be selected from $D_{\text{non-test}}$ by using random selection, center-based selection, border-based selection and hybrid selection methods and they are represented by D_{rs}, D_{cs}, D_{bs} and D_{hs} respectively. Finally D_{rs}, D_{cs}, D_{bs} and D_{hs} will be used as training data for Multilayer Perceptron. We conducted 100 trials on each part and average the result. In each trail, the partition of D_{test} and $D_{\text{non-test}}$ is different.

As shown in Table 2, for each data set (except glass and echocardiogram), the maximal X is usually not greater than the number of test data. In this study, the selection of X might be different for different data sets. For example, we set the values 3, 6, 9, 15 and 21 for the X in iris and 4, 8, 12, 16 and 20 for the X in soybean. The reason for this setting is to simplify the experiment. One simple configuration for CS is to select the same number of training data from each class, so we select evenly divisible numbers of training data respective of the numbers of classes in the data (recall iris has three classes and soybean has four classes). One measure to evaluate the performance of our proposed methods on each data set is the average classification accuracy of all parts. For example, experiment on iris comprises five parts. Then the performance evaluation on iris is based on the average classification accuracy of these five parts. For data selection methods, classification accuracy is important, but it cannot show the robustness of the methods. For example, as shown in Table 3, the performance of two methods (CS and BS) are compared on one data set when training number X is set to $N1$, $N2$, $N3$ and $N4$ respectively.

In Table 3, the average classification accuracy of BS is better than CS. However, we cannot say that BS is really good, because it is not robust. Concretely, CS is better than BS in three cases ($N1$, $N2$ and $N3$). BS is better than CS only in one case ($N4$). So in this work, we use robustness to evaluate their performance.

"Robustness" here is used to evaluate whether the proposed methods can give a consistent improve-

Table 1
UCI data sets used in the empirical study

Data set	Size	Attribute	Class	Class distribution
iris	150	4C	3	50/50/50
soybean	47	35C	4	10/10/10/17
breast-w	698	9C	2	457/241
wine	178	13C	3	59/71/48
glass	214	9C	6	9/13/17/29/70/76
echocardiogram	131	1B 6C	2	88/43
heart1	303	13C	2	164/139
heart2	294	13C	2	188/106
horse	368	4B 5N 6C	2	232/136
german	1000	24C	2	700/300
wdbc	569	30C	2	357/212

B: Binary, N: Nominal, C: Continuous.

Table 2
Size of training data and test data for each data set

Data set	Size	Test data num.	Training data num.
iris	150	75	3,6,9,15,21
soybean	47	25	4,8,12,16,20
breast-w	698	100	6,10,14,20
wine	178	50	6,9,12,15,18,21,30
glass	214	50	24,30,48,60,80
echocardiogram	131	30	10,20,30,40,50,60
heart1	303	50	10,20,30,40,50
heart2	294	50	10,20,30,40,50
horse	368	50	10,20,30,40,50
german	1000	100	10,20,30,40,50
wdbc	569	100	10,20,30,40,50

Table 3
Performance evaluation measure: classification accuracy

Training Data Num.	Classification Accuracy	
	C-S	B-S
N1	0.5	0.4
N2	0.5	0.4
N3	0.5	0.4
N4	0.6	1
Ave.	0.525	0.55

Table 4
Performance evaluation measure: robustness

Training data num.	Classification accuracy comparison
	A1 (C-S) A2 (B-S)
N1	A1$\succ$A2
N2	A1$\succ$A2
N3	A1$\succ$A2
N4	A2$\succ$A1
Total order	A1 (2) $>$ A2 (-2)

ment under different experiment environment (different number of training data). For each number of training data, the value of "Robustness" for a specified method is the difference value between the times that this method outperforms others and the times that others methods outperform this specified method. The value of "Robustness" under different numbers of training data will be aggregated to be the final "Robustness" value of this method.

As shown in Table 4, for each data set, to make a clearer view of the relative performance between each mechanism, a partial order "$\succ$" is defined on the set of all comparing algorithms for different training data size, where A1$\succ$A2 means that the classification accuracy of method A1 is better than that of method A2 on the specific training data number. Note that the partial order "$\succ$" only measures the relative performance between two method A1 and A2 on one specific number (or size) of training data. However, it is quite possible that A1 performs better than A2 in terms of some numbers but worse than A2 in terms of other ones. In this case, it is hard to judge which method is superior. Therefore, in order to give an overall performance assessment of a method, a score is assigned to it which takes account of its relative performance with other methods on all the numbers of training data. Concretely, for each number of training data, for each possible pair of method A1 and A2, if A1$\succ$A2 holds, then A1 is rewarded by a positive score $+1$ and A2 is penalized by a negative score -1. Based on the accumulated score of each method on all evaluation numbers, a total order "$>$" is defined on the set of all comparing methods as

Table 5
Performance comparison of RS, CS, BS and HS on four datasets: iris, soybean, breast-w and wine

(a) Accuracy comparison on iris

	Dataset: Iris			
	Classification Accuracy			
	RS	CS	BS	HS
T = 3	0.571 ± 0.153	$\mathbf{0.744 \pm 0.090}$	0.441 ± 0.174	$\mathbf{0.744 \pm 0.090}$
T = 6	0.719 ± 0.147	0.761 ± 0.076	0.498 ± 0.175	$\mathbf{0.787 \pm 0.113}$
T = 9	0.819 ± 0.101	0.796 ± 0.074	0.523 ± 0.124	$\mathbf{0.846 \pm 0.089}$
T = 15	0.879 ± 0.077	0.826 ± 0.067	0.565 ± 0.099	$\mathbf{0.883 \pm 0.057}$
T = 21	0.893 ± 0.075	0.836 ± 0.062	0.597 ± 0.057	$\mathbf{0.907 \pm 0.051}$
Average	0.776	0.793	0.525	**0.833**

(b) Robustness comparison on iris

	Dataset: Iris			
	Robustness Comparison			
	RS (A1)	CS (A2)	BS (A3)	HS (A4)
T = 3	A2$\succ$A1, A2$\succ$A3, A1$\succ$A3, A4$\succ$A1, A4$\succ$A3			
T = 6	A4$\succ$A1, A4$\succ$A2, A4$\succ$A3, A2$\succ$A1, A2$\succ$A3, A1$\succ$A3			
T = 9	A4$\succ$A1, A4$\succ$A2, A4$\succ$A3, A1$\succ$A2, A1$\succ$A3, A2$\succ$A3			
T = 15	A4$\succ$A1, A4$\succ$A2, A4$\succ$A3, A1$\succ$A2, A1$\succ$A3, A2$\succ$A3			
T = 21	A4$\succ$A1, A4$\succ$A2, A4$\succ$A3, A1$\succ$A2, A1$\succ$A3, A2$\succ$A3			
Total Order	**A4** (**14**) $>$ **A1** (**1**) $>$ **A2** (**0**) $>$**A3** ($-$**15**)			

(c) Accuracy comparison on soybean

	Dataset: Soybean			
	Classification Accuracy			
	RS	CS	BS	HS
T = 4	0.614 ± 0.139	$\mathbf{0.799 \pm 0.131}$	0.418 ± 0.109	$\mathbf{0.799 \pm 0.131}$
T = 8	0.826 ± 0.136	0.872 ± 0.110	0.517 ± 0.107	$\mathbf{0.905 \pm 0.093}$
T = 12	0.912 ± 0.097	$\mathbf{0.938 \pm 0.064}$	0.593 ± 0.154	0.937 ± 0.057
T = 16	0.947 ± 0.087	$\mathbf{0.964 \pm 0.052}$	0.865 ± 0.166	0.954 ± 0.064
T = 20	0.964 ± 0.050	0.964 ± 0.057	0.961 ± 0.072	$\mathbf{0.972 \pm 0.048}$
Average	0.853	0.907	0.671	**0.913**

(d) Robustness comparison on Soybean

	Dataset: Soybean			
	Robustness Comparison			
	RS (A1)	CS (A2)	BS (A3)	HS (A4)
T = 4	A2$\succ$A1, A2$\succ$A3, A1$\succ$A3, A4$\succ$A1, A4$\succ$A3			
T = 8	A4$\succ$A1, A4$\succ$A2, A4$\succ$A3, A2$\succ$A1, A2$\succ$A3, A1$\succ$A3			
T = 12	A2$\succ$A1, A2$\succ$A3, A2$\succ$A4, A4$\succ$A1, A4$\succ$A3, A1$\succ$A3			
T = 16	A2$\succ$A1, A2$\succ$A3, A2$\succ$A4, A4$\succ$A1, A4$\succ$A3, A1$\succ$A3			
T = 20	A4$\succ$A1, A4$\succ$A2, A4$\succ$A3, A2$\succ$A1, A2$\succ$A3, A1$\succ$A3			
Total Order	**A4** (**10**) $>$ **A2** (**9**) $>$ **A1** ($-$**4**) $>$ **A3** ($-$**15**)			

(e) Accuracy comparison on breast-w

	Dataset: Breast Cancer			
	Classification Accuracy			
	R-S	C-S	B-S	H-S
T = 6	0.868 ± 0.106	$\mathbf{0.931 \pm 0.035}$	0.290 ± 0.177	0.922 ± 0.044
T = 10	0.908 ± 0.072	$\mathbf{0.924 \pm 0.039}$	0.341 ± 0.175	0.921 ± 0.040
T = 14	0.925 ± 0.052	0.929 ± 0.040	0.321 ± 0.211	$\mathbf{0.931 \pm 0.034}$
T = 20	0.923 ± 0.052	$\mathbf{0.934 \pm 0.035}$	0.371 ± 0.260	$\mathbf{0.934 \pm 0.032}$
Average	0.906	**0.930**	0.331	0.927

Table 5, continued

(f) Robustness comparison on breast-w

	Dataset: Breast Cancer			
	Robustness Comparison			
	R-S (A1)	C-S (A2)	B-S (A3)	H-S (A4)
T = 6	A2≻A1, A2≻A3, A2≻A4, A4≻A1, A4≻A3, A1≻A3			
T = 10	A2≻A1, A2≻A3, A2≻A4, A4≻A1, A4≻A3, A1≻A3			
T = 14	A4≻A1, A4≻A2, A4≻A3, A2≻A1, A2≻A3, A1≻A3			
T = 20	A2≻A1, A2≻A3, A4≻A1, A4≻A3, A1≻A3			
Total Order	**A2** (**9**) > **A4** (**7**) > **A1** (−**4**) >**A3** (−**12**)			

(g) Accuracy comparison on wine

	Dataset: Wine			
	Classification Accuracy			
	RS	CS	BS	HS
T = 6	0.704 ± 0.139	**0.838 ± 0.062**	0.756 ± 0.120	0.835 ± 0.071
T = 9	0.772 ± 0.116	0.829 ± 0.074	0.788 ± 0.112	**0.849 ± 0.065**
T = 12	0.823 ± 0.094	0.819 ± 0.068	0.844 ± 0.079	**0.857 ± 0.068**
T = 15	0.837 ± 0.078	0.821 ± 0.066	0.860 ± 0.062	**0.870 ± 0.052**
T = 18	0.863 ± 0.074	0.835 ± 0.053	**0.873 ± 0.049**	0.871 ± 0.056
T = 21	0.870 ± 0.061	0.847 ± 0.062	0.871 ± 0.060	**0.875 ± 0.052**
T = 30	0.871 ± 0.057	0.870 ± 0.055	0.881 ± 0.055	**0.884 ± 0.046**
Accuracy	0.820	0.837	0.839	**0.863**

(h) Robustness comparison on wine

	Dataset: Wine			
	Robustness Comparison			
	RS (A1)	CS (A2)	BS (A3)	HS (A4)
T = 6	A2≻A1, A2≻A3, A2≻A4, A4≻A1, A4≻A3, A3≻A1			
T = 9	A4≻A1, A4≻A2, A4≻A3, A2≻A1, A2≻A3, A3≻A1			
T = 12	A4≻A1, A4≻A2, A4≻A3, A3≻A2, A3≻A1, A1≻A2			
T = 15	A4≻A1, A4≻A2, A4≻A3, A3≻A2, A3≻A1, A1≻A2			
T = 18	A3≻A1, A3≻A2, A3≻A4, A4≻A1, A4≻A2, A1≻A2			
T = 21	A4≻A1, A4≻A2, A4≻A3, A3≻A1, A3≻A2, A1≻A2			
T = 30	A4≻A1, A4≻A2, A4≻A3, A3≻A1, A3≻A2, A1≻A2			
Total Order	**A4** (**17**) > **A3** (**5**) > **A1** (**-11**) = **A2** (**-11**)			

shown in the last row of Table 3, where A1 > A2. In this case, we say A1 is more robust than A2.

The experimental results of our proposed data selection mechanisms on different data sets are shown in the following tables. For each data set, experimental result includes two parts. One part is the average accuracy on different training data numbers. The other part is used to show the robustness of each mechanism. The value following "±"gives the standard deviation and the best result on each training data number is shown in bold face. We only give the experimental results on the four datasets, iris, soybean, breast-w and wine. Other results are given in Appendix. In the following tables, "T" refers to the number of training data. Note that for each data set, when the number of training data equals to the number of classes in it, hybrid selection is same as center-based selection.

Table 6 exhibits that, on iris, soybean, wine, glass, echo, heart2, horse, and german, HS is best both on average accuracy and robustness. Concretely its average accuracy is 5.7, 6, 4.3, 3.4, 2.6, 2.7, 0.1, and 1 percent better than RS on these datasets respectively. On breast-w and heart1, CS is best both on average accuracy and robustness. And its average accuracy is 2.4 and 3.3 percent better than RS on these two datasets. Note that even in these two datasets, HS is also better than RS. Its average accuracy is 2.1 and 0.3 percent better than RS on them. On the wdbc dataset, HS and RS have the same accuracy. However, HS is more robust than CS.

In summary, the observations reported in this section suggest that (performance of random selection is regarded as baseline):

(1) Center-based selection shows better performance as compared to random selection in certain datasets.
(2) Border-based selection does not show significant improvement over random selection.

Table 6
Performance comparison of RS, CS, BS and HS on learning

Dataset	Accuracy				Robustness			
	RS	CS	BS	HS	RS	CS	BS	HS
iris	0.776	0.793	0.525	**0.833**	1	0	−15	**14**
soybean	0.853	0.907	0.671	**0.913**	−4	9	−15	**10**
breast-w	0.906	**0.930**	0.331	0.927	−4	**9**	−12	7
wine	0.820	0.837	0.839	**0.863**	−11	−11	5	**17**
glass	0.558	0.521	0.513	**0.592**	5	−10	−10	**15**
echocardiogram	0.650	0.667	0.617	**0.676**	−4	6	−18	**16**
heart1	0.741	**0.774**	0.585	0.744	−3	**15**	−15	3
heart2	0.753	0.769	0.667	**0.780**	−3	7	−15	**11**
horse	0.749	0.708	0.640	**0.750**	**10**	−5	−15	**10**
german	0.656	0.651	0.591	**0.666**	3	1	−15	**11**
wdbc	**0.919**	0.884	0.739	**0.919**	7	−2	−9	**9**

(3) Hybrid selection outperforms random selection under all the selected datasets.

It is not difficult to understand this result. For center-based selection, it selects the samples with high degree of membership in each cluster. These samples are usually representative and valuable for learning, however, with the number of training data increasing, these samples might be redundant for learning. This is the reason why it cannot provide stable improvement compared to random selection. If we further restrict the number of training data to some extent, center-based selection will be superior to random selection. For border-based selection, it selects the samples around the borders between two classes. As they are quite likely to lie near the decision boundaries of classes, they can be regarded as "confusing samples". If these confusing samples are used solely for training, training might be overfitted to them to give bad generality for unseen samples. Hence border-based selection is always worst among these four methods. For hybrid selection, it inherits the advantages from both center-based and border-based selection. Center-based selection provides representative samples for learning, while border-based selection refines the performance of center-based learning.

In this work, we empirically evaluate our data selection methods in the case that there is no any prelabeled data available. In contract to it, if small prelabeled data exist, we would make use of them to boost the performance of clustering. This technique is called semi-supervised clustering [35][36]. In the future, we will continue our research to combine semi-supervised clustering with data selection.

5. Conclusions and future work

To achieve the best possible classifier with a small number of labeled data, in this paper, three training data mechanisms are proposed by using fuzzy clustering method. They are center-based selection, border-based selection and hybrid selection. Center-based selection chooses the samples with high degree of membership in each cluster. In border-based selection, the samples located at the borders between clusters are selected. Hybrid selection is the combination of them. Experimental results on a set of UCI data sets indicate that hybrid selection could effectively improve learning performance.

In current work, the samples around centers and borders are simply combined without considering their distributions and densities. Therefore, it is interesting to see whether the information of distribution and density could further improve the performance of hybrid-selection mechanism.

Acknowledgement

This research was supported by the MKE (Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Advancement) (IITA-2008-(C1090-0801-0002)). This study was also supported by a grant of the Korea Health 21 R&D Project, Ministry For Health, Welfare and Family Affairs, Republic of Korea (A020602).

References

[1] A.K. Jain, M.N. Murty and P.J. Flynn, Data clustering: a review, in: *ACM Computing Surveys*, 1999, 264–323.
[2] X.J. Zhu, Semi-supervised learning literature survey, Technical report 1530, Computer Science, University of Wisconsin-Madiso, 2006.

[3] D.D. Lewis and W.A. Gale, A sequential algorithm for training text classifiers, in: *Proceedings of 17th ACM International Conference on Research and Development in Information Retrieval*, 1994, 3–12.
[4] D. Mackay, Information-based objective functions for active data selection, *Neural Computation* **4**(4) (1992), 305–318.
[5] O. Chapelle, B. Scholkopf and A. Zien, *Semi-Supervised Learning*, MIT-Press, 2006.
[6] A.P. Dempster, N.M. Laird and D.B. Rubin, Maximum likelihood for incomplete data via the EM algorithm, *Journal of Royal Statistical Society* **B39**(1) (1977), 1–38.
[7] B.M. Shahshahani and D.A. Landgrebe, The effect of unlabeled samples in reducing the small sample size problem and mitigating the Hughes phenomenon, *IEEE Transaction on Geoscience and Remote Sensing* **32**(5) (1994), 1087–1095.
[8] D.J. Miller and H.S. Uyar, A mixture of experts classifier with learning based on both labeled and unlabelled data, *Advances in Neural Information Processing Systems* (1997), 571–577.
[9] K. Nigam, A.K. Mccallum, S. Thrun and T. Mitchell, Text classification from labeled and unlabeled documents using EM, in: *Proceedings of 17th International Conference on Machine Learning*, 2000, 103–134.
[10] K. Fukumizu, Stastical active learning in multilayer perceptrons, *IEEE Transactions on Neural Networks* **11**(1) (2000), 17–26.
[11] C. Campbell, N. Cristianini and A. Smola, Query learning with large margin classifiers, in: *Proceedings of 17th International Conference on Machine Learning*, 2000, 111–118.
[12] S. Tong and D. Koller, Support vector machine active learning with application to text classification, *Journal of Machine Learning Research* **2** (2001), 45–66.
[13] G. Schohn and D. Cohn, Less is more: active learning with support vector machines, in: *Proceedings of 17th International Conference on Machine Learning*, 2000, 839–846.
[14] D.D. Lewis and J. Catlett, Heterogeneous uncertainty sampling for supervised learning, in: *Proceedings of 11th International Conference on Machine Learning*, 1994, 148–156.
[15] C. Thompson, M.E. Califf and R.J. Mooney, Active learning for natural language parsing and information extraction, in: *Proceeding of 16th International Conference on Machine Learning*, 1999, 406–414.
[16] M. Tang, X. Luo and S. Roukos, Active learning for statistical natural language parsing, in: *Proceeding of 40th Anniversary Meeting of Association for Computational Linguistics*, 2002, 120–127.
[17] R. Hwa, On minimizing training corpus for parser acquisition, in: *Proceeding of 5th Computational Natural Language Learning Workshop*, 2001, 84–89.
[18] D. Cohn, L. Atlas and R. Ladner, Improving generalization with active learning, *Machine Learning Journal* **15** (1994), 210–221.
[19] I. Dagan and S.P. Engelson, Committee-based sampling for training probabilistic classifiers, in: *Proceeding of 12th International Conference on Machine Learning*, 1995, 150–157.
[20] I.A. Muslea, *Active learning with multiple views*, Ph.D. dissertation, Univ. Southern California, 2000.
[21] R. Liere, *Active learning with committees: An approach to efficient learning in text categorization using linear threshold algorithms*, Ph.D. dissertation, Oregon State Univ. 2000.
[22] N. Cebron and M.R. Berthold, Adaptive Fuzzy Clustering, Annual meeting of the North American Fuzzy Information Processing Society, NAFIPS 2006, Digital Object Identifier: 10.1109/NAFIPS.2006.365406, Publication Date: 3–6 June 2006, ISBN: 1-4244-0363-4.
[23] J. Kang, Kwang Ryel Ryu and Hyuk-Chul Kwon, Using Cluster-Based Sampling to Select Initial Training Set for Active Learning in Text Classification, *The 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2004, 384–388.
[24] H.T. Nguyen and A. Smeulders, Active learning using pre-clustering, in: *Proceedings of the twenty-first international conference on Machine learning*, 2004, 79–86.
[25] M. Li and I.K. Sethi, Confidence-based active learning, *IEEE Transaction on Pattern Analysis and Machine Intelligence* **28**(8) (2003), 1251–1261.
[26] K. Brinker, Incorporating diversity in active learning with support vector machines, in: *Proceedings of 20th International conference on machine learning*, 2003, 59–66.
[27] S. Hoi, R. Jin, J. Zhu and M. Lyu, Batch mode active learning and its application to medical image classification, in: *Proceedings of the 23th International Conference on Machine Learning*, 2006, 417–424.
[28] S. Hoi, R. Jin and M. Lyu, Large-scale text categorization by batch mode active learning, in: *Proceedings of the International World Wide Web Conference*, 2006, 633–642.
[29] G. Schohn and D. Cohn, Less is more: Active learning with support vector machines, in: *Proceedings of the 17th International Conference on Machine Learning*, 2000, 839–846.
[30] Z. Xu, K. Yu, V. Tresp and J. Wang, Representative sampling for text classification using support vector machines, in: *Proceedings of the 25th European Conference on Information Retrieval Research*, 2003, 393–407.
[31] J.C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York, 1981.
[32] L.A. Zadeh, Fuzzy sets as a basis for a theory of possibility, *Fuzzy Sets and Systems* **1**(1) (1978), 3–28.
[33] Yashil, Fuzzy C-Means Clustering MATLAB Toolbox, http://ce.sharif.edu/~m_amiri/project/yfcmc/index.htm.
[34] UCI Machine Learning Repository, http://www.ics.uci.edu/>>mlearn/MLRepository.html.
[35] S. Basu, *Semi-supervised Clustering: Probabilistic Models, Algorithms and Experiments*, Ph. D. Thesis, Department of Computer Science, University of Texas at Austin, 2005.
[36] B. Kulis and R.J. Mooney, Semi-supervised graph clustering: A kernel approach, in: *Proceedings of the 22nd International Conference on Machine Learning*, 2005, 457–464.

Appendix: Experimental results

Table A1. Performance comparison of RS, CS, BS and HS on seven datasets: glass, echo, heart1, heart2, horse, german and wdbc

(a) Accuracy comparison on glass

	Dataset: Glass			
	Classification Accuracy			
	RS	CS	BS	HS
T = 24	0.507 ± 0.098	0.481 ± 0.090	0.494 ± 0.099	$\mathbf{0.562 \pm 0.074}$
T = 30	0.536 ± 0.078	0.513 ± 0.082	0.513 ± 0.094	$\mathbf{0.567 \pm 0.070}$
T = 48	0.564 ± 0.074	0.514 ± 0.075	0.528 ± 0.078	$\mathbf{0.584 \pm 0.078}$
T = 60	0.604 ± 0.076	0.572 ± 0.074	0.514 ± 0.076	$\mathbf{0.615 \pm 0.067}$
T = 80	0.580 ± 0.074	0.525 ± 0.079	0.517 ± 0.088	$\mathbf{0.633 \pm 0.065}$
Average	0.558	0.521	0.513	**0.592**

(b) Robustness comparison on glass

	Dataset: Glass			
	Robustness Comparison			
	RS (A1)	CS (A2)	BS (A3)	HS (A4)
T = 24	A4≻A1, A4≻A2, A4≻A3, A1≻A2, A1≻A3, A3≻A2			
T = 30	A4≻A1, A4≻A2, A4≻A3, A1≻A2, A1≻A3			
T = 48	A4≻A1, A4≻A2, A4≻A3, A1≻A2, A1≻A3, A3≻A2			
T = 60	A4≻A1, A4≻A2, A4≻A3, A1≻A2, A1≻A3, A2≻A3			
T = 80	A4≻A1, A4≻A2, A4≻A3, A1≻A2, A1≻A3, A2≻A3			
Total Order	**A4** (**15**) > **A1** (**5**) > **A3** (**−10**) = **A2** (**−10**)			

(c) Accuracy comparison on echo

	Dataset: Echo			
	Classification Accuracy			
	RS	CS	BS	HS
T = 10	0.63 ± 0.120	0.621 ± 0.125	0.597 ± 0.119	$\mathbf{0.641 \pm 0.103}$
T = 20	0.641 ± 0.088	$\mathbf{0.671 \pm 0.092}$	0.564 ± 0.121	0.660 ± 0.092
T = 30	0.644 ± 0.086	0.686 ± 0.081	0.603 ± 0.093	$\mathbf{0.691 \pm 0.079}$
T = 40	0.655 ± 0.081	0.666 ± 0.086	0.637 ± 0.083	$\mathbf{0.689 \pm 0.070}$
T = 50	0.658 ± 0.074	0.682 ± 0.073	0.639 ± 0.087	$\mathbf{0.687 \pm 0.077}$
T = 60	0.672 ± 0.079	0.673 ± 0.088	0.660 ± 0.091	$\mathbf{0.686 \pm 0.081}$
Average	0.650	0.667	0.617	**0.676**

(d) Robustness comparison on echo

	Dataset: Echo			
	Robustness Comparison			
	RS (A1)	CS (A2)	BS (A3)	HS (A4)
T = 10	A4≻A1, A4≻A2, A4≻A3, A1≻A2, A1≻A3, A2≻A3			
T = 20	A2 ≻A1, A2≻A3, A2≻A4, A4≻ A1, A4 ≻A3, A1≻A3			
T = 30	A4 ≻A1, A4≻A2, A4≻A3, A2≻A1, A2≻A3, A1≻A3			
T = 40	A4 ≻A1, A4≻A2, A4≻A3, A2≻A1, A2≻A3, A1≻A3			
T = 50	A4 ≻A1, A4≻A2, A4≻A3, A2≻A1, A2≻A3, A1≻A3			
T = 60	A4 ≻A1, A4≻A2, A4≻A3, A2≻A1, A2≻A3, A1≻A3			
Total Order	**A4** (**16**) > **A2** (**6**) > **A1** (**-4**) > **A3** (**-18**)			

Table A1, continued

(e) Accuracy comparison on heart1

	Dataset: Heart1			
	Classification Accuracy			
	RS	CS	BS	HS
T = 10	0.707 ± 0.096	**0.756 ± 0.078**	0.592 ± 0.103	0.733 ± 0.056
T = 20	0.740 ± 0.071	**0.762 ± 0.067**	0.591 ± 0.108	0.743 ± 0.066
T = 30	0.740 ± 0.065	**0.801 ± 0.056**	0.591 ± 0.106	0.747 ± 0.061
T = 40	0.753 ± 0.072	**0.785 ± 0.062**	0.583 ± 0.087	0.754 ± 0.058
T = 50	0.766 ± 0.060	**0.768 ± 0.056**	0.567 ± 0.105	0.744 ± 0.065
Average	0.741	**0.774**	0.585	0.744

(f) Robustness comparison on heart1

	Dataset: Heart1			
	Robustness Comparison			
	RS (A1)	CS (A2)	BS (A3)	HS (A4)
T = 10	A2 ≻A1, A2≻A3, A2≻A4, A4≻A1, A4≻A3, A1≻A3			
T = 20	A2 ≻A1, A2≻A3, A2≻A4, A4≻A1, A4≻A3, A1≻A3			
T = 30	A2 ≻A1, A2≻A3, A2≻A4, A4≻A1, A4≻A3, A1≻A3			
T = 40	A2 ≻A1, A2≻A3, A2≻A4, A4≻A1, A4≻A3, A1≻A3			
T = 50	A2 ≻A1, A2≻A3, A2≻A4, A1≻A4, A1≻A3, A4≻A3			
Total Order	**A2 (15)** $>$ **A4 (3)** $>$ **A1 (-3)** $>$ **A3 (-15)**			

(g) Accuracy comparison on heart2

	Dataset: Heart2			
	Classification Accuracy			
	RS	CS	BS	HS
T = 10	0.754 ± 0.085	0.769 ± 0.063	0.515 ± 0.163	**0.789 ± 0.068**
T = 20	0.747 ± 0.071	**0.773 ± 0.061**	0.648 ± 0.127	0.766 ± 0.069
T = 30	0.749 ± 0.091	0.761 ± 0.075	0.715 ± 0.087	**0.787 ± 0.064**
T = 40	0.755 ± 0.069	0.750 ± 0.062	0.742 ± 0.078	**0.775 ± 0.060**
T = 50	0.762 ± 0.063	**0.792 ± 0.065**	0.716 ± 0.100	0.781 ± 0.063
Average	0.753	0.769	0.667	**0.780**

(h) Robustness comparison on heart2

	Dataset: Heart2			
	Robustness Comparison			
	RS (A1)	CS (A2)	BS (A3)	HS (A4)
T = 10	A4 ≻A1, A4≻A2, A4≻A3, A2≻A1, A2≻A3, A1≻A3			
T = 20	A2 ≻A1,A2≻A3.A2≻A4, A4≻A1, A4≻A3, A1≻A3			
T = 30	A4 ≻A1, A4≻A2, A4≻A3, A2≻A1, A2≻A3, A1≻A3			
T = 40	A4≻A1,A4≻A2,A4≻A3, A1≻A2,A1≻A3,A2≻A3			
T = 50	A2 ≻A1,A2≻A3.A2≻A4, A4≻A1, A4≻A3, A1≻A3			
Total Order	**A4 (11)** $>$ **A2 (7)** $>$ **A1 (−3)** $>$ **A3 (−15)**			

(i) Accuracy comparison on horse

	Dataset: Horse			
	Classification Accuracy			
	RS	CS	BS	HS
T = 10	**0.708 ± 0.107**	0.691 ± 0.072	0.611 ± 0.090	**0.708 ± 0.087**
T = 20	0.730 ± 0.078	0.709 ± 0.061	0.649 ± 0.079	**0.748 ± 0.064**
T = 30	**0.758 ± 0.071**	0.704 ± 0.060	0.643 ± 0.076	0.748 ± 0.064
T = 40	**0.775 ± 0.068**	0.720 ± 0.071	0.639 ± 0.080	0.767 ± 0.063
T = 50	0.774 ± 0.062	0.716 ± 0.066	0.656 ± 0.088	**0.777 ± 0.056**
Average	0.749	0.708	0.640	**0.750**

Table A1, continued

(j) Robustness comparison on horse

	Dataset: Horse			
	Robustness Comparison			
	RS (A1)	CS (A2)	BS (A3)	HS (A4)
T = 10	A1≻A2, A1≻A3, A4≻A2,A4≻A3,A2≻A3			
T = 20	A4≻A1,A4≻A2,A4≻A3,A1≻A2,A1≻A3,A2≻A3			
T = 30	A1≻A2,A1≻A3,A1≻A4,A4≻A2,A4≻A3,A2≻A3			
T = 40	A1≻A2,A1≻A3,A1≻A4,A4≻A2,A4≻A3,A2≻A3			
T = 50	A4≻A1,A4≻A2,A4≻A3,A1≻A2,A1≻A3,A2≻A3			
Total Order	**A1** (**10**) = **A4** (**10**) > **A2** (−**5**) > **A3** (−**15**)			

(k) Accuracy comparison on german

	Dataset: German			
	Classification Accuracy			
	R-S	C-S	B-S	H-S
T = 10	0.633 ± 0.072	**0.665 ± 0.049**	0.532 ± 0.102	0.656 ± 0.050
T = 20	0.657 ± 0.065	0.660 ± 0.066	0.618 ± 0.076	**0.671 ± 0.052**
T = 30	0.653 ± 0.063	0.639 ± 0.057	0.565 ± 0.086	**0.673 ± 0.055**
T = 40	0.667 ± 0.060	0.642 ± 0.053	0.593 ± 0.079	**0.670 ± 0.052**
T = 50	**0.669 ± 0.058**	0.648 ± 0.050	0.648 ± 0.067	0.660 ± 0.058
Average	0.656	0.651	0.591	**0.666**

(l) Robustness comparison on german

	Dataset: German			
	Robustness Comparison			
	RS (A1)	CS (A2)	BS (A3)	HS (A4)
T = 10	A2≻A1,A2≻A3,A2≻A4,A4≻A1,A4≻A3,A1≻A3			
T = 20	A4≻A1,A4≻A2,A4≻A3,A2≻A1,A2≻A3,A1≻A3			
T = 30	A4≻A1,A4≻A2,A4≻A3,A1≻A2,A1≻A3,A2≻A3			
T = 40	A4≻A1,A4≻A2,A4≻A3,A1≻A2,A1≻A3,A2≻A3			
T = 50	A1≻A2,A1≻A3,A1≻A4,A4≻A2,A4≻A3			
Total Order	**A4** (**11**) > **A1** (**3**) > **A2** (**1**) > **A3** (−**15**)			

(m) Accuracy Comparison on wdbc

	Dataset: Wdbc			
	Classification Accuracy			
	R-S	C-S	B-S	H-S
T = 10	0.885 ± 0.058	0.872 ± 0.050	0.578 ± 0.187	**0.906 ± +0.035**
T = 20	0.915 ± 0.039	0.883 ± 0.051	0.652 ± 0.188	**0.917+0.034**
T = 30	0.920 ± 0.032	0.887 ± 0.035	0.696 ± 0.195	**0.927 ± 0.035**
T = 40	**0.938 ± 0.033**	0.891 ± 0.039	0.818 ± 0.131	0.927 ± 0.029
T = 50	0.939 ± 0.025	0.889 ± 0.039	**0.953 ± 0.054**	0.919 ± 0.038
Average	**0.919**	0.884	0.739	**0.919**

(n) Robustness comparison on wdbc

	Dataset: Wdbc			
	Robustness Comparison			
	RS (A1)	CS (A2)	BS (A3)	HS (A4)
T = 10	A4≻A1,A4≻A2,A4≻A3,A1≻A2,A1≻A3,A2≻A3			
T = 20	A4≻A1,A4≻A2,A4≻A3,A1≻A2,A1≻A3,A2≻A3			
T = 30	A4≻A1,A4≻A2,A4≻A3,A1≻A2,A1≻A3,A2≻A3			
T = 40	A1≻A2,A1≻A3,A1≻A4,A4≻A2,A4≻A3,A2≻A3			
T = 50	A3≻A1,A3≻A2,A3≻A4,A1≻A2,A1≻A4,A4≻A2			
Total Order	**A4** (**9**) > **A1** (**7**) > **A2** (−**7**) > **A3** (−**9**)			