Journal of AI and Data Mining
Vol 3, No 2, 2015, 121-130. 10.5829/idosi.JAIDM.2015.03.02.01

IRDDS: Instance reduction based on Distance-based decision surface

J. Hamidzadeh

Faculty of Computer Engineering & Information Technology, Sadjad University of Technology, Mashhad, Iran.

Received 23 April 2015; Accepted 23 August 2015
Corresponding author: j_hamidzadeh@sadjad.ac.ir (J. Hamidzadeh).

Abstract

In instance-based learning, a training set is given to a classifier for classifying new instances. In practice, not all information in the training set is useful for classifiers. Therefore, it is convenient to discard irrelevant instances from the training set. This process is known as instance reduction, which is an important task for classifiers since through this process the time for classification or training could be reduced. Instance-based learning methods are often confronted with the difficulty of choosing the instances, which must be stored to be used during an actual test. Storing too many instances may result in large memory requirements and slow execution speed. In this paper, first, a Distance-based Decision Surface (DDS) is proposed and is used as a separate surface between the classes, and then an instance reduction method, which is based on the DDS is proposed, namely IRDDS (Instance Reduction based on Distance-based Decision Surface). Using the DDS with Genetic algorithm selects a reference set for classification. IRDDS selects the most representative instances, satisfying both of the following objectives: high accuracy and reduction rates. The performance of IRDDS is evaluated on real world data sets from UCI repository by the 10-fold cross-validation method. The results of the experiments are compared with some state-of-the-art methods, which show the superiority of the proposed method, in terms of both classification accuracy and reduction percentage.

Keywords: *Instance Reduction, Distance-based Decision Surface, Support Vector Machine, Genetic Algorithm.*

1. Introduction

In the pattern recognition, supervised classification is a procedure that assigns a label to an unclassified sample, trained by a set of previously classified samples. Classification is one of the most important goals of pattern recognition [1].

In the literature of data classification, there are some methods that classify data based on distance between new data and training samples. Instance reduction is a crucial task in some instance-based learning methods. Instance-based learning methods are often confronted with the difficulty of choosing the instances, which must be stored to be used during an actual test. Storing too many instances may result in a large memory occupation and decrement in execution speed. Actually, some training sets may contain non- or a little information, which can be either noisy or redundant. Therefore, a process will be needed to discard unwanted information from the training set. In literature, this discarding process is known as the instance reduction process. As a result, after some of the instances have been removed from the training set, the amount of memory needed for storage and the time required for an actual test are reduced.

One main challenge in designing an instance reduction algorithm is the maintenance of border, central, or other sets of points. In this research, we have decided to maintain the border points because the internal ones do not affect the decision boundaries as much as the non-boundaries do [2-5]. Hence, the internal points can be removed with a low impact on classification accuracy. A large number of border points may be needed to define a border completely, so some methods maintain central points in order to use those instances which are the most typical of a particular class to classify instances close to them. This can affect the decision boundaries, because the decision boundaries depend on not only where the instances of one class lie, but also where those of the other classes lie.

In instance reduction, we often face a trade-off between the sample size and the classification accuracy [6]. A successful algorithm often reduces the size of the training set significantly without a significant reduction on classification accuracy. In some cases, classification accuracy can be even increased with the reduction of instances, when noisy instances are removed or decision boundaries are smoothed. A recently survey of different methods for data reduction can be seen in [7].

This paper focuses on the problem of reducing the size of the stored set of instances while trying to maintain classification accuracy. This is first accomplished by providing a survey of methods which have been employed to reduce the number of instances (that are needed in learning methods) and then by proposing an instance reduction technique based on the Distance-based Decision Surface (DDS) [8], namely IRDDS (Instance Reduction based on Distance-based Decision Surface). In [8], a weighted quadratic decision surface is derived. In this paper, we have derived an unweighted decision surface of order one.

The remainder of the paper is organized as follows: In section 2, a survey of instance reduction methods is presented. In section 3, the proposed Distance-based Decision Surface is introduced in subsection 3.1. The proposed instance reduction method (IRDDS) is introduced in subsection 3.2. Finally, the statistical stability of the proposed method is proved in subsection 3.3. An evaluation of the proposed method is presented in section 4 and its performance is compared to some state-of-the-art methods. Finally, in section 5, conclusions and future research direction are presented.

2. Survey of instance reduction methods and distance-based classifiers

Several methods have been proposed for instance reduction, some of which are surveyed in this section. Most of the methods discussed here use T as original instances in the training set and S ⊆ T (S as a subset of T) as their representatives.

The Condensed Nearest Neighbor (CNN) [9] and the Edited Nearest Neighbor (ENN) [10] rules are the first two methods of instance reduction. The CNN begins by selecting one instance which belongs to any class from T randomly and puts it in S, and then each instance in T is classified by using only the instances in S. If an instance is misclassified, it will be added to S in order to ensure that it will be classified correctly. This process will be repeated until there are no misclassified instances in T. Instance Based Learning Algorithms (IBn), introduced in [11], can be considered as editing methods. IB2 is an online learning method, similar to CNN, IB2 works by adding to an initially empty set (S) those instances that are not correctly classified by the edited set (S). Within this setting, a newly available instance that is not added to the edited set does not need to be stored. Since noisy instances are very likely to be misclassified, they are almost always maintained in the edited set. In order to overcome this weakness, the IB3 method uses a wait and see evidence gathering method to determine which of the kept instances are expected to perform well during classification.

The Reduced Nearest Neighbor (RNN) rule was introduced by Gates [12]. RNN algorithm starts with $S = T$ and removes each instance from S until further removal does not cause any other instances in T to be misclassified. RNN is computationally more expensive than Hart's Condensed Nearest Neighbor (CNN) rule, but it always produces a subset of CNN. Thus, RNN is less expensive in terms of computation and storage during the classification stage. Since the removed instance is not guaranteed to be classified correctly, this algorithm is able to remove noisy and internal instances while maintaining border points.

Another variant of CNN is the Generalized Condensed Nearest Neighbor (GCNN) [13], which is similar to CNN. However, GCNN assigns instances that satisfy an absorption criterion to S. The absorption is calculated in terms of the nearest neighbors and its rivals (the nearest instances of the other classes). An instance is absorbed or included in S if its distance compared to its nearest neighbor and its nearest rivals are not more than a threshold.

In ENN algorithm, S starts out the same as T, then any instance in S which does not agree with the majority of its k nearest neighbors is removed. This removes noisy instances as well as close border cases. It also maintains all internal points, which keep it from reducing the storage requirements as much as most of the other reduction methods do. A variant of this method is the Repeated ENN (RENN). The RENN applies the ENN algorithm repeatedly until all remaining instances have the majority of their neighbors with the same class. Another extension of ENN is all k-NN method [14]. This algorithm works as follows: for i = 1 to k, flag any instances which are not classified correctly by its i nearest neighbors, as bad. When the loop is repeated for k times, any instances which are flagged as bad are removed.

In order to reduce storage requirements and remove noisy instances, an instance *t* should be removed if all *k* of its neighbors are from the same class or even from a class other than *t*. This process removes noisy instances as well as internal ones, while maintaining border ones. Unlike most previous methods, there are some methods such as DROP1 to DROP5 which pay more attention to the order according to which instances have been removed [2]. In these methods, each instance t has k nearest neighbors (ordered from the nearest to the farthest) and those instances that have t as one of their k nearest neighbors are called the associates of t (sorted from the nearest to the farthest). The Iterative Case Filtering algorithm (ICF) was proposed in [15]. ICF is based on the Coverage and Reachable sets which are the neighboring set and associate set, respectively. The neighborhood set for an instance such as t is all the instances between t and the nearest enemy of t. The nearest enemy of t is the nearest instance from the other classes. Those instances that have t as one of their k nearest neighbors are called the associates set of t, where t is the training set. In this method, an instance t is flagged for removal if |Reachable(t)| > |Coverage(t)|, which means that more cases can solve t than t can solve itself, then all the instances flagged for removal will be deleted. Another method that finds border instances is proposed in [3], namely Prototype Selection by Clustering (PSC), which applies clustering algorithm. Two types of clustered regions are in PSC method, namely homogeneous and heterogeneous clusters. In homogeneous cluster, all instances are from the same class, whereas, in heterogeneous clusters, they are from different classes. Thus, two types of instances are in PSC, one of which is the mean of the instances in each homogeneous cluster and the other is from heterogeneous clusters as a border instance.

Evolutionary algorithms have been used for instance reduction, with promising results. The basic idea is to maintain a population of chromosomes, which represents solutions to the problem and evolves over time through a process of competition. In this evaluation, both data reduction and classification accuracy are considered. The examples of application of genetic algorithm and other evolutionary algorithms for instance reduction can be found in [16-18]. The CHC evolutionary algorithm [16] and Steady-State Memetic Algorithm (SSMA) [17] are the most known evolutionary algorithms. In terms of instance selection, the SVM (Support Vector Machines) not only is a classifier but also an instance selection method. SVBPS (Support Vector Based Prototype Selection) is a wrapper method which is based on SVM [19]. It works doing a double selection; the first one applies SVM to obtain the support vectors and the second one applies DROP2 over the support vectors. FRPS (Fuzzy Rough Prototype Selection) is a fuzzy-based method which is introduced in [20]. It uses fuzzy rough set theory to express the quality of the instances and uses a wrapper method to determine which instances to be removed.

Nikolaidis and et. al. introduced a multi-stage method for instance reduction and abstraction in [5], namely Class Boundary Preserving (CBP) algorithm. CBP is a hybrid method which selects and abstracts instances from training set that are close to the class boundaries. In the first stage of CBP, using ENN algorithm smooths the class boundaries. In the second stage, it tries to distinguish between border and non-border instances by using the geometric characteristics of the instance distribution. In the third stage, border instances are pruned by using the concept of mutual neighborhood, and in the last stage, the non-border instances are clustered.

Hamidzadeh and et. al. introduced a Large Symmetric Margin Instance Selection method, namely LAMIS [21]. LAMIS removes non-border instances and keeps border ones. This method presents an instance reduction process through formulating it as a constrained binary optimization problem and solves it by employment filled function algorithm. In LAMIS, the core of instance selection process is based on keeping the hyperplane that separates a two-class data, to provide large margin separation. These authors introduced another instance reduction method in [4]. This method is based on hyperrectangle clustering, called Instance Reduction Algorithm using Hyperrectangle Clustering (IRAHC). IRAHC removes interior instances and keeps border and near border ones. This method presents an instance reduction process based on hyperrectangle clustering. A hyperrectangle is an n-dimensional rectangle with axes aligned sides, which is defined by min and max points and a corresponding distance function. The min-max points are determined by using the hyperrectangle clustering algorithm. In IRAHC, the core of instance reduction process is based on the set of hyperrectangles.

A survey of the related classifiers is given below. Classification can be done based on sample properties, one of which is distance. Distance is a numerical description of how much objects are departed. In the Euclidean space $\Re^n$, the distance between two points is usually given by the Euclidean distance (2-norm distance). Based on other norms, different distances are used such as 1-, p- and infinity-norm. In classification, various distances can be employed to measure the closeness, such as the Euclidean, Mahalanobis [22] or bands distance [23].

In literature of data classification, there are some methods that classify data based on the distance between new unseen data and training samples. One of the classifiers is Minimum Distance Classifier (MDC) [1]. It classifies an unknown sample into a category to which the nearest prototype pattern belongs. In this classifier a Euclidean distance is used as the metric. Senda et al. based on karhunen-loeve expansion omit the redundant calculations of MDC [24].

3. The proposed method

In this section, first we propose a distance-based decision surface and then a new method for instance reduction, namely IRDDS (Instance Reduction based on Distance-based Decision Surface) is introduced. IRDDS is based on the proposed DDS. For two given classes, we calculate the average distances of all the training samples. Unclassified samples are classified as a class that has the smaller average distance. Applying such a rule leads to derive a formula to be used as the decision surface. Afterwards, we present an instance reduction method based on DDS by employing genetic algorithm.

The proposed Distance-based Decision Surface is introduced in subsection 3.1. Also, a kernel extension of DDS is introduced in this subsection. The proposed instance reduction method (IRDDS) is introduced in subsection 3.2. Finally, the statistical stability of the proposed method is proved in subsection 3.3.

The steps of IRDDS are described as follow: In the first step of IRDDS, the original training sample is considered. In the second step of IRDDS, a proper distance-based surface, namely DDS, is obtained by using parameter tuning which is described in subsections 3.1 and 3.2. In the next two steps, the instance reduction is done based on DDS using Genetic Algorithm. This step of IRDDS is described in subsection 3.2. Finally, the reduced training sample is obtained through IRDDS.

3.1. Distance-based Decision Surface (DDS)

In this subsection, we derive a formula to determine the decision surface between two classes of samples. Let $d_1(x,x_i)$ denote distance of the test sample x to the training sample of the first class and denote distance of the test sample x to the training sample of the second class. DDS is based on the distance between unclassified sample x and samples of two classes. The goal is to determine decision surface in a way that the average distances from the two classes are equal. Hence, (1) is employed to determine the decision surface.

$$\frac{1}{n_1}\sum_{i=1}^{n_1} d_1(x,x_i)=\frac{1}{n_2}\sum_{j=1}^{n_2} d_2(x,x_j) \tag{1}$$

The average distance of x from all samples in each class is shown in (1), so a decision surface can be derived. In (1), and are used as the number of training samples for the first and second classes respectively. It should be noted that d(*,*) calculates the distance between two points and the same distance function is used in (1). We can derive a linear equation as a decision surface from (1). Therefore, to classify a new unclassified sample, we have presented a formula as the decision surface.

Here, we use the Euclidean distance (2-norm). Thus, substituting distances with this norm gives (2).

$$\begin{aligned} &x^T x\,(n_2 n_1) - 2x^T\left(n_2\sum_{i=1}^{n_1} x_i\right) + n_2\sum_{i=1}^{n_1} x_i^T x_i = \\ &x^T x\,(n_1 n_2) - 2x^T\left(n_1\sum_{j=1}^{n_2} x_j\right) + n_1\sum_{j=1}^{n_2} x_j^T x_j \end{aligned} \tag{2}$$

The proposed decision surface is derived where a and b are defined as in (3) and (4), respectively,

$$a = 2n_1 n_2\ \left(\left(\sum_{j=1}^{n_2} x_j\right)/n_2 - \left(\sum_{i=1}^{n_1} x_i\right)/n_1\right) \tag{3}$$

$$b = n_2\sum_{i=1}^{n_1} x_i^T x_i - n_1\sum_{j=1}^{n_2} x_j^T x_j \tag{4}$$

$$\mathrm{F(x)} = ax + b = 0 \tag{5}$$

A linear decision surface is shown in (5), which is called Distance-based Decision Surface (DDS). To classify a test sample such as x, it is sufficient to determine the sign of (5). Input test sample cannot properly be classified if the sign of decision surface is neither positive nor negative. In this situation the label of the sample is randomly assigned.

Kernel methods are powerful statistical learning techniques, which are widely applied to various learning algorithms [25]. Kernel methods can be employed to transform samples into a high dimensional space. In the high dimensional space, various methods can be employed to separate samples linearly. A mapping function denoted can be employed to transform samples into a high dimensional space. By using kernel function, the inner products between the images of the data can be substituted in the feature space. As a result, we have (6).

$$K(x,y) = <\varphi(x),\varphi(y)> = \varphi(x)^T \varphi(y) \qquad (6)$$

Thus, using (1) and Euclidean distance in the high dimensional feature space gives (7).

$$\begin{aligned} &n_2 \times \sum_{i=1}^{n_1}\left[(\varphi(x)-\varphi(x_i))^T\,(\varphi(x)-\varphi(x_i))\right] = \\ &n_1 \times \sum_{j=1}^{n_2}\left[(\varphi(x)-\varphi(x_j))^T\,(\varphi(x)-\varphi(x_j))\right] \end{aligned} \qquad (7)$$

Using (6) in (7) gives:

$$\begin{aligned} &n_2 \times \sum_{i=1}^{n_1}\left[k(x,x) - 2k(x,x_i) + k(x_i,x_i)\right] = \\ &n_1 \times \sum_{j=1}^{n_2}[k(x,x) - 2k(x,x_j) + k(x_j,x_j)] \end{aligned} \qquad (8)$$

Using Radial Basis Function (RBF) kernel in (8) gives:

$$n_1\sum_{j=1}^{n_2}\exp(-\frac{\|x-x_j\|^2}{2\sigma^2}) - n_2\sum_{i=1}^{n_1}\exp(-\frac{\|x-x_i\|^2}{2\sigma^2}) = 0 \qquad (9)$$

As a result of using this kernel function, we have a nonlinear decision surface as shown in (9).

3.2. Reduction based on DDS using genetic algorithm

In the context of data reduction, we are required to reduce instances while maintaining the data classification accuracy. Hence, in the reduction of instances, we often face a trade-off between the sample size and the classification accuracy. Therefore, instance reduction is a multi-objective optimization problem that attempts to maximize the classification accuracy and, at the same time, minimizes the sample size. The issue of measuring the classification accuracy is associated with the sample quality issue, in which we aim to achieve the best representative sample with the minimum size. In DDS algorithm, it is better to use the stored instances during generalization in order to avoid excessive storage, time complexity, and possibly to improve generalization by avoiding noise and overfitting. DDS maintains all the training instances. It learns very quickly, because it needs only to read the training set without any further processing. Hence, it generalizes better for many applications. However, it has relatively large memory requirements because it stores all the training instances. It must make computation through all available instances in order to classify a new input sample, so it is slow during the classification task. Moreover, since it stores all instances in the training set, noisy instances are stored as well, and can degrade the classification accuracy. On the other hand, when some of the instances are removed from the training set, the storage requirements and time needed for generalization are correspondingly reduced. This subsection focuses on the problem of reducing the size of the stored set of instances while trying to maintain or even improve the classification accuracy. In this subsection, we propose a new method for instance reduction that is based on DDS, namely IRDDS (Instance Reduction based on Distance-based Decision Surface).

Genetic Algorithms (GA) are optimization processes inspired in natural evolution laws. In this paper, a binary genetic algorithm is applied to optimize the bit-stream in order to implement instance reduction based on DDS. Hence, the process is carried out for instance reduction (for two classes) by applying a binary GA as follows:

a) The binary chromosome consists of n1+n2 gens, where n1 and n2 are the number of instances in the first and second classes, respectively. Each gene can have two possible states: 0 or 1. If the gene is 1, then its associated instance will be maintained. If it is 0, then its associated instance will be discarded.

b) The cost function (fitness function) that must be minimized can be defined as follows:

The cost function is the combination of the two following terms: the reverse of sample reduction rate and the classification error rate. The cost function = A + B, where A equals weight multiplied by inverse of RedRate, and B equals (1-weight) multiplied by error_rate. The sample reduction rate (RedRate) is defined as the discarded training sample size divided by the original training sample size.

c) GA is applied in order to obtain a subset of the training samples that minimizes the cost function.

The weight parameter that is used in the cost function is found to obtain a quite good trade-off between storage requirements and achieved error rate for the experiments, which have been conducted in this paper. The weight parameter takes values in interval [0, 1]. Regarding the parameters, we preserved the value of weight=0.5 as the best choice, due to the fact that it was analysed in the previous work related to instance selection [26]. During the training process, the cost function over the validation set is calculated for the best individual of each population. The individual that achieves the lowest cost value over the validation set is selected as the final individual. The other GA parameters that have been used in the experiments are reported in the next section.

3.3. Statistical stability of the proposed method

Statistical stability is the most fundamental, which means the patterns, which are identified by the algorithm, and really genuine patterns of the data source but not just an accidental relation occurring in the finite training set. This property can be considered as the statistical robustness of the output in the sense that if we rerun the algorithm on a new sample from the same source, it should identify a similar pattern. Proving that a given pattern is indeed significant and is the concern of ‘learning theory’, a body of principles and methods that estimate the reliability of pattern functions under appropriate assumptions about the way in which the data was generated. The Rademacher complexity measures the capacity of a class. It assesses the capacity of the class by its ability to fit random noise. The difference between empirical and true estimation over the pattern class can be bounded in terms of its Rademacher complexity [27].

The following theorem can be used to derive an upper-bound of the generalization error in terms of Rademacher complexity for the proposed method [27].

Theorem: Let P be a probability distribution on Z, where Z be training samples drawn according to P with probability at least $1-\delta$, the proposed function f (or DDS classifier) satisfies:

$$P(y \neq f(x)) \leq \hat{P}_n(y \neq f(x)) + 2\sqrt{\frac{E(k(X,X))}{n}} + \sqrt{\frac{\ln(\frac{1}{\delta})}{2n}} \tag{10}$$

where, function *f* is as:

$$f(x)=\sum_{i=1}^{n}\mu_i k(x,x_i) \tag{11}$$

For obtaining the upper bound in theorem 1, at first, we prove (12). This inequality shows the empirical complexity measure of the proposed function *f*.

$$\hat{G}_n(f) \leq \frac{2}{n}\left[\sum_{i=1}^{n} k(x_i,x_i)\right]^{\frac{1}{2}} \tag{12}$$

$$\begin{aligned} F = &\left\{\sum_{i=1}^{n}\mu_i k(x,x_i) : n \in N, x_i \in X\right\} \subseteq \\ &\left\{< w,\Phi(x) >: \|w\| \leq 1\right\} \end{aligned} \tag{13}$$

According to the Sup properties, Cauchy Schwarz and Jensen’s inequality, we have

$$\begin{aligned} \hat{G}_n(F) &\leq \frac{2}{n} E\left[(\sum_{i=1}^{n}\sum_{j=1}^{n} g_i g_j k(x_i,x_j))^{\frac{1}{2}} |X_i \right] \\ &\leq \frac{2}{n}\left(\sum_{i=1}^{n}\sum_{j=1}^{n} E\left[g_i g_j k(x_i,x_j)|X_i\right]\right)^{\frac{1}{2}} \end{aligned} \tag{14}$$

Finally, we obtain (15) and the theorem proved.

$$\begin{aligned} &G_n(F)=E(\hat{G}_n(F)) \leq E\left(\frac{2}{n}\left(\sum_{i=1}^{n} k(x_i,x_i)\right)^{\frac{1}{2}}\right) = \\ &\frac{2}{n}E(\sqrt{\sum_{i=1}^{n} k(x_i,x_i)})=\frac{2}{n}E(\sqrt{n \times \|\varphi(x)\|^2}) = \\ &\frac{2}{n}E(\|\varphi(x)\| \times \sqrt{n})=\frac{2}{\sqrt{n}}R \end{aligned} \tag{15}$$

4. Experimental results

In this section, at first, SVM and DDS classifiers have been compared, and then IRDDS is compared with the some state-of-the-art instance reduction methods. Extensive experiments have been conducted to evaluate the performance of the proposed method against five state-of-the-art instance selection methods using some real world data sets.

In order to validate IRDDS, experiments have been conducted over the real world data sets which have been taken from the UCI data set repository [28]. The selected data sets and their characteristics are shown in table 1. In this table, #samples, #features, and #classes denote the number of instances, the number of attributes, and the number of classes, respectively.

The data sets are grouped into two categories depending on the size they have (a horizontal line divides them in the table). The small data sets have less than 1000 instances and the larger data sets have more than 1000 instances. In each group, the data sets have been sorted increasingly depending on their #classes.

In the first experiment, SVM and DDS classifiers have been compared. To this end, the classifiers have been tested on the first group of the data sets (the small size data sets).

Table 1. Selected data sets of UCI data repository [28].

No.	Data Set	#samples	#features	#classes
1	Ionosphere	351	34	2
2	Sonar	208	60	2
3	Wdbc	569	30	2
4	Liver	345	6	2
5	Haberman	306	3	2
6	Heart	267	44	2
7	Pima	768	8	2
8	Musk	476	167	2
9	Trans	748	4	2
10	Iris	150	4	3
11	Vehicle	846	18	4
12	Glass	214	9	7
13	Ecoli	336	7	8
14	Vowel	990	10	11
15	Census	299,285	41	2
16	Satimage	4435	36	6
17	Segment	2310	19	7
18	Shuttle	43500	9	7
19	Yeast	1484	8	10
20	Pendigits	3498	16	10
21	Poker	350,000	10	10
22	Letter	15000	16	26

The experimental results (without instance reduction for the DDS) are summarized in table 2. Note that the values in table 2 denote the error classification rates in percentage. For multiclass classification, we can use classification by pairwise coupling. In this paper, we have used the voting approach in comparison to the classifier output approaches for the extension of two-class classifier as multiclass classifier.

In order to show the performance of IRDDS, it was compared against the other reduction methods, which include BEPS, LAMIS, DROP3, IRAHC and PSC, using k-NN with k=3 and the Euclidean distance. All the methods have been tested on all the data sets in terms of classification error rate and reduction percentage performance measures. Table 3 shows the results obtained using the competing methods. For each method, the average classification error rates (Err) and the reduction percentages (Red) are shown. As already mentioned, instance reduction is a multi-objective optimization problem.

Therefore, table 5 presents both classification error rates and instance reduction percentages for all the competing methods and all the data sets. The reduction percentages are the ratio of the number of discarded instances to the number of instances in the original data set multiplied by 100.

Table 2. The classification error rates (percent) and their parameters.

Data Set	SVM			DDS	
	C	б	Rate	б	Rate
Ionosphere	2	4	**3.99**	0.1	5.12
Sonar	0.8	8	14.42	0.4	**11.06**
Wdbc	4	2	4.92	1	**2.99**
Liver	2	4	**26.96**	8	32.46
Haberman	2	2	24.84	12	**24.18**
Heart	1	4	**16.85**	0.8	17.60
Pima	4	4	**27.08**	4	29.43
Musk	0.2	16	17.15	0.2	**14.20**
Trans	8	2	**18.45**	8	19.10
Iris	0.2	2	3.33	0.2	**2.67**
Vehicle	2	8	23.95	1	**19.10**
Glass	4	2	28.50	0.1	28.50
Ecoli	4	4	**12.80**	0.1	13.39
Vowel	1	2	24.15	0.4	**19.84**

In table 5, for each data set, the best methods have been highlighted in terms of its instance reduction percentage or error rate. As can be seen, in the cases of problems, which IRDDS could not attain the highest reduction percentage, this method has almost the lowest error rate.

Overall, among the competing methods, IRDDS exhibits the best average of instance reduction percentage.

As already mentioned, instance reduction is a two-objective optimization problem and a gain in one objective becomes unpleasant with the other objective. Therefore, table 3 presents both classification error rates and reduction percentages for all the competing methods and all the data sets. There are some major factors that affect different performances of IRDDS and the other methods used for instance reduction in each problem or data set. First of all, the complexity of classes in each problem is a key factor in instance reduction percentage. Secondly, there are noises and outliers in each problem differently. In table 4, average ranks in terms of error rate and reduction percentage are shown separately.

Table 4 indicates that, it is clear that IRDDS exhibits good average rank in terms of both error rate and reduction percentage. IRDDS obtains the best classification error rate and the second best instance reduction percentage ranks among the competing methods. Overall, IRDDS exhibits good results in both classification error rates and instance reduction percentages.

In order to compare the execution speed of the competing methods, the average computation times of IRDDS and the other methods measured in seconds per run are reported in table 5. It includes training time. Time required to reduce the training set averaged over the training sub sets in 10-fold cross validation.

Table 3. Average error rates (Err) and reduction percentages (Red) of IRDDS and the other methods.

Data Set	BEPS		LAMIS		DROP3		IRAHC		PSC		IRDDS	
	Err	Red	Err	Red	Err	Red	Err	Red	Err	Red	Err	Red
Ionosphere	15.20	68.24	15.13	79.62	17.32	55.70	**14.86**	89.87	18.94	43.79	**14.86**	**96.90**
Sonar	14.09	80.12	12.23	**90.35**	27.55	72.11	**10.23**	90.01	20.24	64.59	15.23	75.87
Wdbc	14.90	45.87	17.31	66.93	18.93	28.12	16.89	66.72	21.62	43.59	**4.85**	**89.75**
Liver	34.68	54.13	32.79	**77.92**	35.02	66.84	32.45	76.36	44.64	54.13	**25.30**	60.55
Haberman	26.70	80.30	23.54	**83.51**	33.40	80.90	**22.62**	84.85	36.38	64.70	23.74	48.28
Heart	**21.38**	27.85	32.13	88.95	36.22	88.11	31.72	**88.15**	35.59	46.92	22.72	85.07
Pima	26.03	58.43	24.29	89.21	30.30	**94.46**	**23.53**	89.06	29.83	56.94	25.13	67.15
Musk	16.03	81.60	15.01	48.38	28.72	47.49	14.63	48.50	27.84	31.59	**13.80**	**82.50**
Trans	25.80	71.24	24.93	89.47	30.19	82.07	25.78	89.02	28.73	43.76	**25.19**	**91.10**
Iris	4.31	69.32	3.47	73.87	4.27	68.10	3.87	75.68	5.34	**79.55**	**3.0**	71.26
Vehicle	35.19	90.32	32.69	91.40	47.07	90.10	**29.95**	90.13	38.75	67.45	32.92	**92.18**
Glass	23.78	87.35	20.85	**89.63**	27.68	77.46	**19.85**	88.45	40.91	61.55	24.95	60.23
Ecoli	13.87	71.40	**15.12**	**88.71**	19.45	74.16	15.49	87.03	18.62	73.84	18.29	81.38
Vowel	**8.50**	76.94	9.14	84.96	9.31	65.41	8.54	85.14	15.75	54.76	8.91	**87.19**
Census	27.43	54.10	26.18	73.90	22.65	65.98	28.70	54.60	24.65	**77.78**	**19.32**	66.50
Satimage	6.04	82.35	**5.13**	75.87	7.07	74.80	11.25	54.10	7.33	62.85	6.50	**84.02**
Segment	7.01	68.02	**6.12**	56.29	9.60	**80.36**	18.13	60.20	13.45	44.29	10.75	48.12
Shuttle	19.10	58.97	17.84	62.54	26.81	49.31	21.16	46.10	13.67	69.10	**13.08**	**70.23**
Yeast	44.60	78.06	43.01	82.97	43.82	82.86	42.67	83.70	46.96	56.80	**42.11**	**84.36**
Pendigits	6.01	75.23	5.19	90.04	3.46	85.18	4.28	90.63	5.68	**93.77**	**2.15**	31.52
Poker	28.34	62.95	32.14	65.10	29.73	73.56	27.98	48.20	42.93	73.96	**21.36**	**76.10**
Letter	7.01	67.30	6.14	82.59	15.76	48.38	5.62	83.97	20.65	81.67	**4.18**	**85.10**

Table 4. Average ranks in terms of error rate and reduction percentage ranks.

Average rank	BEPS	LAMIS	DROP3	IRAHC	PSC	IRDDS
Error rate (Err)	3.43 (4)	3.24 (3)	4.65 (5)	2.80 (2)	5.26 (6)	2.11 (1)
Reduction percentage (IR)	4.33 (5)	2.48 (1)	3.96 (4)	3.04 (3)	4.45 (6)	2.73 (2)

Table 5: Average computational times of IRDDS and the other methods measured in seconds per run.

Data set	BEPS	LAMIS	DROP3	IRAHC	PSC	IRDDS
Ionosphere	6.1(5)	3.8(2.5)	7.5 (6)	3.2(1)	4.1 (4)	3.8 (2.5)
Sonar	0.98(1)	1.43(5)	1.3 (3)	1.3(3)	1.7 (6)	1.3 (3)
Wdbc	37.1(5)	5.28(2)	87.34 (6)	4.8(1)	5.8 (3)	19.4 (4)
Liver	3.8(1)	15.18(5)	5.0 (2)	14.8(3)	16.8 (6)	15.03 (4)
Haberman	4.0(5)	2.30(2)	4.1 (6)	2.1(1)	2.6 (3)	3.5 (4)
Heart	1.12(1)	2.46(6)	2.2 (2)	2.4(4.5)	2.3 (3)	2.4 (4.5)
Pima	38.6(5)	5.6(3)	83.5 (6)	4.7(1)	6.5 (4)	5.1 (2)
Musk	19.8(5)	4.15(3)	75.10(6)	3.5(2)	4.3(4)	1.2(1)
Trans	37.9(5)	5.8(3)	82.06(6)	5.3(2)	6.4(4)	5.2(1)
Iris	0.7(1)	3.04(5)	0.8 (2)	2.9(3.5)	3.16 (6)	2.9 (3.5)
Vehicle	26.6(5)	6.2(2)	86.38(6)	5.3(1)	7.10(4)	6.34(3)
Glass	1.6(4)	1.16(1)	1.5 (2)	1.6(4)	1.7 (6)	1.6 (4)
Ecoli	7.5(5)	2.91(3)	8.2 (6)	2.8(1.5)	3.2 (4)	2.8 (1.5)
Yeast	71.8(5)	6.8(2)	76.90(6)	5.8(1)	7.13(4)	6.95(3)
Vowel	51.6(5)	6.4(2)	58.12(6)	6.2(1)	8.39(3)	8.40(4)
Census	487,131.12(5)	32,452.06(3)	500,260.8(6)	28,325.87(1)	73,629.1 (2)	85,201.0 (4)
Satimage	119.6(1)	154.42(4)	130.92(2)	149.18(3)	184.36(5)	192.10(6)
Segment	89.6(3)	83.41(2)	7624.9 (6)	79.80(1)	107.67 (5)	96.05 (4)
Shutlle	8943.7(2)	613.34(1)	9938.12(3)	10276.38(4)	11083.64(5)	12541.18(6)
Pendigits	97.4(1)	123.62(4)	11587.3 (6)	98.70(2)	158.5 (3)	146.93 (5)
Poker	13,350.11(3)	6430.24(2)	728,851.6(6)	852.23(1)	93,436.8 (4)	142,106.4(5)
Letter	492.6(1)	517.17(3)	504.02(2)	601.4(4)	716.84(6)	698.76(5)
Average rank	3.39	3.06	4.70	2.07	4.22	3.57

5. Conclusions and future research

In this paper, a new instance reduction procedure is presented. The introduced method employs a Distance-based Decision Surface (DDS) and Genetic algorithm, called IRDDS. In IRDDS, we proposed a decision surface for a binary classification process. An input test sample with respect to the decision surface can be assigned to one of the two classes.

The experiment results show IRDDS is the best among all the other methods. IRDDS obtains the best classification error rate and the second best instance reduction percentage ranks among the competing methods. Overall, IRDDS exhibits good results in both classification error rates and instance reduction percentages. The performance of IRDDS is demonstrated through the experiments, and the results show that IRDDS has good robustness in noisy cases. The results clearly demonstrate that IRDDS often yields the most robust performance over the data sets.

IRDDS is an offline method. As future research, we are interested in extending it to use for online applications, such as data stream.

References

[1] Duda, R. O., et al. (2001). Pattern classification. New York: Wiley Interscience Publication.

[2] Wilson, D. R. & Martinez, T. R. (2000). Reduction techniques for instance-based learning algorithms. Machine Learning, vol. 38, pp. 257–286.

[3] Olvera-Lo´pez, J. A., Carrasco-Ochoa, J. A. & Martı´nez-Trinidad, J. F. (2010). A new fast prototype selection method based on clustering. Pattern Analysis and Applications, vol. 13, no. 2, pp. 131-141.

[4] Hamidzadeh, J., Monsefi, R., & Yazdi, H. S. (2015). IRAHC: Instance Reduction Algorithm using Hyperrectangle Clustering. Pattern Recognition, vol. 48, pp. 1878–1889.

[5] Nikolaidis, K., Goulermasn, J. Y. & Wu, Q. H. (2011). A class boundary preserving algorithm for data condensation. Pattern Recognition, vol. 44, pp. 704–715.

[6] Liu, H. & Motoda, H. (2002). On Issues of Instance Selection. Data Mining Knowledge Discovery, vol. 6, pp. 115-130.

[7] Garcı´a, S., Derrac, J., Cano, J. R. & Herrera, F. (2012). Prototype selection for nearest neighbor classification: taxonomy and empirical study. IEEE Transactions on Pattern and Machine Intelligence, vol. 34, no. 3, pp. 417 – 435.

[8] Hamidzadeh, J., Monsefi, R. & Sadoghi Yazdi, H. (2012). DDC: distance-based decision classifier. Neural Computation & Application, no. 21, pp. 1697–1707.

[9] Hart, P. (1968). The condensed nearest neighbor rule. IEEE Transaction on Information Theory, vol. 14, pp. 515–516.

[10] Wilson, D. (1972). Asymptotic properties of nearest neighbor rules using edited data. IEEE Transactions on Systems, Man and Cybernetics, vol. 2, pp. 408–421.

[11] Aha, D. W. & Kibler D. (1991). Instance-based learning algorithms. Machine Learning, vol. 6, no. 1, pp. 37–66.

[12] Gates, G. W. (1972). The Reduced Nearest Neighbor Rule. IEEE Transactions on Information Theory, vol. 18, no. 3, pp. 431-433.

[13] Chien-Hsing, C., Bo-Han, K. & Fu, C. (2006). The generalized condensed nearest neighbor rule as a data reduction method. Proceedings of the 18th international conference on pattern recognition, IEEE Computer Society, Hong-Kong, pp. 556–559, 2006.

[14] Tomek I. (1976). An experiment with the edited nearest-neighbor rule. IEEE Transactions on Systems, Man and Cybernetics, vol. 6, no. 6, pp. 448–452.

[15] Brighton, H. & Mellish, C. (2002). Advances in instance selection for instance-based learning algorithms. Data Mining and Knowledge Discovery, vol. 6, no. 2, pp. 153–172.

[16] Cano, J. R., Herrera, F. & Lozano, M. (2003). Using evolutionary algorithms as instance selection for data reduction in KDD: an experimental study. IEEE Transactions on Evolutionary Computation, vol. 7, no. 6, pp. 561–575.

[17] Garcia, S., Cano, J. R. & Herera, F. (2008). A memetic algorithm for evolutionary prototype selection: a scaling up Approach. Pattern Recognition, vol. 41, pp. 2693–2709.

[18] Olvera-Lopez, A. J., Carrasco-Ochoa, J. F., Martinez-Trinidad, J. A. & Kittler, J. (2010). A review of instance selection methods. Artificial Intelligence Review, vol. 34, pp. 133–143.

[19] Li, Y., Hu, Z., Cai, Y. & Zhang, W. (2005). Support vector based prototype selection method for nearest neighbor rules. Lecture Notes in Computer Science, vol. 3610, pp. 528-535.

[20] Verbiest, N., Cornelis, C. & Herrera, F. (2013). FRPS: A Fuzzy Rough Prototype Selection method, Pattern Recognition, vol. 46, no. 10, pp. 2770-2782.

[21] Hamidzadeh, J., Monsefi, R., & Yazdi, H. S. (2014). Large symmetric margin instance selection algorithm. International Journal of Machine Learning and Cybernetics, doi:10.1007/s13042-014-0239-z.

[22] Cevikalp, H., Larlus, D., Douze, M. & Jurie, F. (2007). Local subspace classifiers: linear and nonlinear approaches. IEEE Workshop on Machine Learning for Signal Processing, 2007.

[23] Laguia, M. & Castro, J. L. (2008). Local distance-based classification. Knowledge-Based Systems, vol. 21, pp. 692-703.

[24] Senda, S., Minoh, M. & Ikeda, K. (1995). A fast algorithm for the minimum distance classifier and its application to kanji character recognition. Proceedings of the Third International Conference on Document Analysis and Recognition, vol. 1, pp. 283-286, 1995.

[25] Pekalska, E. & Hassdonk, B. (2009). Kernel discriminant analysis for positive definite and indefinite kernels. IEEE Transactions on Pattern and Machine Intelligence, vol. 31, no. 6.

[26] Garcı´a, S., Derrac, J., Luengo, J., Carmona, C. J. & Herrera, F. (2011). Evolutionary Selection of Hyperrectangles in Nested Generalized Exemplar Learning. Applied Soft Computing, vol. 11, pp. 3032-3045.

[27] Bartlett, P. L. & Mendelson, S. (2002). Rademacher and Gaussian Complexities: Risk Bounds and Structural Results. Journal of Machine Learning Research, vol. 3, pp. 463-482.

[28] UCI Machine Learning Repository (2013). Available: http://archive.ics.uci.edu/ml.

نشریه هوش مصنوعی و داده کاوی

دانشگاه شاهرود

کاهش نمونه بر اساس سطح تصمیم‌گیری مبتنی بر فاصله

جواد حمیدزاده

دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی سجاد، مشهد، ایران.

ارسال ۲۰۱۵/۰۴/۲۳؛ پذیرش۲۰۱۵/۰۸/۲۳

چکیده:

در یادگیری مبتنی برنمونه، از یک مجموعه داده برای آموزش طبقه‌بند استفاده می‌شود. در واقع، تمام نمونه‌های موجود در این مجموعه، برای آموزش طبقه‌بند ضروری نیست. بنابراین لازم است که نمونه‌های غیرضروری از مجموعه آموزشی حذف گردند. به این رویه حذف، رویه کاهش نمونه گویند. رویه کاهش نمونه، یک مرحله حیاتی برای طبقه‌بندها به شمار می‌آید زیرا در نتیجه آن، زمان آموزش و آزمایش طبقه‌بند کاهش می‌یابد. یک چالش مهم در این رویه، حفظ و نگهداری نمونه‌های مفید است زیرا نگهداری نمونه‌های زیاد باعث افزایش مصرف حافظه و زمان اجرای طبقه‌بند می‌گردد. در این مقاله، ابتدا یک سطح تصمیم مبتنی برفاصله برای جداسازی داده‌های دو کلاسه پیشنهاد شده است. سپس یک رویه کاهش نمونه براساس سطح تصمیم مزبور و الگوریتم ژنتیک ارایه شده است. به کمک سطح تصمیم فوق و الگوریتم ژنتیک، نمونه‌های مفید حفظ و نمونه‌های غیرضرور نیز به گونه‌ای حذف می‌گردند که نرخ صحت طبقه‌بند کاهش نیافته و نرخ کاهش نمونه نیز بالا باشد. در ارزیابی روش پیشنهادی، از داده‌های واقعی برگرفته شده از سایت UCI استفاده شده است. در این ارزیابی از روش اعتبارسنجی ضربدری ده تایی بهره گرفته شده است. نتایج حاصله با روش‌های مرز دانش مقایسه شده است. نتایج نشان دهنده برتری روش پیشنهادی نسبت به سایر روش‌های مرز دانش با توجه به دو معیار نرخ صحت و نرخ کاهش نمونه است.

کلمات کلیدی: کاهش نمونه، سطح تصمیم‌گیری مبتنی بر فاصله، ماشین بردار پشتیبان، الگوریتم ژنتیک.