The Journal of Systems and Software 106 (2015) 1–8

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

Learning to detect representative data for large scale instance selection

Wei-Chao Lin [a], Chih-Fong Tsai [b,*], Shih-Wen Ke [c], Chia-Wen Hung [a], William Eberle [d]

[a] Department of Computer Science and Information Engineering, Hwa Hsia University of Technology, Taiwan
[b] Department of Information Management, National Central University, Taiwan
[c] Department of Information and Computer Engineering, Chung Yuan Christian University, Taiwan
[d] Department of Computer Science, Tennessee Technological University, USA

ARTICLE INFO

Article history:
Received 27 September 2014
Revised 19 February 2015
Accepted 9 April 2015
Available online 14 April 2015

Keywords:
Instance selection
Data reduction
Data mining

ABSTRACT

Instance selection is an important data pre-processing step in the knowledge discovery process. However, the dataset sizes of various domain problems are usually very large, and some are even non-stationary, composed of both old data and a large amount of new data samples. Current algorithms for solving this type of scalability problem have certain limitations, meaning they require a very high computational cost over very large scale datasets during instance selection. To this end, we introduce the ReDD (**Re**presentative **D**ata **D**etection) approach, which is based on outlier pattern analysis and prediction. First, a machine learning model, or detector, is used to learn the patterns of (un)representative data selected by a specific instance selection method from a small amount of training data. Then, the detector can be used to detect the rest of the large amount of training data, or newly added data. We empirically evaluate ReDD over 50 domain datasets to examine the effectiveness of the learned detector, using four very large scale datasets for validation. The experimental results show that ReDD not only reduces the computational cost nearly two or three times by three baselines, but also maintains the final classification accuracy.

1. Introduction

1.1. Background

The large size of today's data collections often makes them very difficult for the current data mining algorithms to handle properly. As a consequence, data pre-processing has become one of the most important steps in KDD (knowledge discovery in databases) for good quality data mining. In other words, if the chosen dataset contains too many instances (i.e., data samples), it can result in large memory requirements, slow execution speed, and over-sensitivity to noise. Another problem with using the original data points is that there may not be any located at the precise points that would make for the most accurate and concise concept description (Pyle, 1999).

Data pre-processing is often implemented using instance selection, or data reduction. The aim of instance selection is to reduce the dataset size by filtering out data from a given dataset that are noisy, redundant or both, and so likely to degrade the mining performance (Wilson and Martinez, 2000; Li and Jacob, 2008). More specifically, instance selection is used to shrink the amount of data, after which data mining algorithms can be applied to the reduced dataset. Sufficient results are achieved if the selection strategy is appropriate (Reinartz, 2002).

This task is similar to outlier detection (Hodge and Austin, 2004) or anomaly detection (Chandola et al., 2009) where the aim is to discover observations that lie an abnormal distance from other values in a population. Simply, outliers are the unusual observations (or bad data points) that are far removed from the mass of data. In other words, they are further away from the sample mean than what is deemed reasonable. Consequently, outliers could lead to significant performance degradation (Aggarwal and Yu, 2001; Barnett and Lewis, 1994).

Filtering out the detected outliers is very useful for discovering the normative patterns in the data (Knorr et al., 2000). Therefore, from the data mining perspective, the aim of instance selection can be thought of as the same as outlier detection (Liu and Motoda, 2001). In other words, performing instance selection and outlier detection over a given dataset can reduce the size of datasets and ensure that they contain higher proportions of representative data.

1.2. Motivation

Defining whether outliers are lying an abnormal distance from other samples or not is a subjective process and defining what constitutes an outlier or determining whether or not an observation is an outlier is a difficult problem. Many instance selection and outlier

* Corresponding author. Tel.: +886 3 422 7151; fax: +886 3 4254604.
E-mail address: cftsai@mgt.ncu.edu.tw (C.-F. Tsai).

http://dx.doi.org/10.1016/j.jss.2015.04.038
0164-1212/

detection methods have been proposed to detect and remove unrepresentative data from a given dataset, and they have shown some promising results (García et al., 2012; Hodge and Austin, 2004; Chandola et al., 2009).

However, since we live in a non-stationary environment, datasets in many domains do not always contain fixed numbers of data samples. In other words, new data samples are continually being added to the database for data mining, which causes the dataset size to become larger and larger. As a consequence, wrong decisions could be made if mining results are discovered from 'out of date' datasets.

For example, instance selection or outlier detection performed over a given dataset D_1 containing 10,000 examples collected at a specific time T_1, results in a reduced dataset $D_{1\_reduced}$ for the later mining stage (where the size of $D_{1\_reduced}$ is smaller than D_1). However, after some time, the size of D_1 becomes larger as new data samples, D_{new}, are stored. As a result, a new larger dataset D_2, composed of D_1 and D_{new}, is created at time T_2. At this point, there are two possible strategies for performing instance selection or outlier detection. The first one, the common strategy, is usually employed over D_2. This can be regarded as the static environment problem without considering the growing dataset. The second one involves performing instance selection over D_{new} resulting in $D_{new\_reduced}$ where the reduced dataset of D_2 is the combination of $D_{1\_reduced}$ and $D_{new\_reduced}$.

In these two cases, the computational cost of performing this data-processing task becomes higher and higher as the new dataset becomes larger and larger in size. This creates the problem of a very high computational cost which is required for performing instance selection over D_2 or D_{new}.

To this end, we introduce a novel process, namely ReDD (**Re**presentative **D**ata **D**etection) which is based on analyzing (or learning) the patterns of unrepresentative data that are identified in the instance selection step. These patterns are then used as guidelines to predict whether a new data sample is representative or not. This prediction task can be accomplished by training a supervised machine learning model. The hypothesis behind ReDD is that if (un)representative data can be well predicted over a set of new data samples, there is no need to perform instance selection again over a new, larger dataset. For the previous example, we only need to train a specific classifier over a two-class training set composed of the representative group (i.e., $D_{1\_reduced}$) and the unrepresentative group (i.e., $D_1 - D_{1\_reduced}$). The classifier can then be used to distinguish between representative and unrepresentative data over D_{new}.

Consequently, the time cost of ReDD over D_{new} can be much smaller than that of performing instance selection over D_2 or D_{new} since the time for performing on-line classification as testing is usually much shorter than off-line learning as training (Chang et al., 2010; Edakunni and Vijayakumar, 2009). In our case, the total time for training a classifier over D_1, and testing the classifier to perform representative data detection as the on-line classification task over D_{new}, is smaller than directly performing instance selection over D_2 or D_{new}, especially when D_2 or D_{new} is certainly larger than D_1.

Note that detecting (un)representative data using ReDD is different from the existing outlier detection methodology used to detect (non)outliers. First, outlier detection aims to detect whether a new exemplar lies in a region of normality, but ReDD focuses on training a classifier to classify a new exemplar into one of two pre-defined classes (i.e., representative and unrepresentative classes) without considering the 'normal' and 'abnormal' data distributions. Second, for supervised learning based upon outlier detection approaches, the number of outliers in the training dataset is usually very small. In addition, the training dataset is typically based on manually labeling normal and abnormal data. On the other hand, the generation of the training dataset in ReDD is based on instance selection, which usually contains a large number of unrepresentative data and a small number of representative data,[1] with the labeling for the two groups of data being fully automatic.

The rest of this paper is organized as follows. Section 2 briefly reviews related literature of instance selection and outlier detection. Section 3 introduces the proposed ReDD process for (un)representative data analysis and prediction. Section 4 presents the experimental results and the conclusion is provided in Section 5.

2. Literature review

2.1. Instance selection

Instance selection can be defined as follows. Given a dataset D composed of training set T and testing set U, let X_i be the ith instance in D, where $X_i = (X_1, X_2, \ldots, X_m)$ which contains m different features. Let $S \subset T$ be the subset of selected instances that result from the execution of an instance selection algorithm. Then, U is used to test a classification technique trained by S (Cano et al., 2003; García et al., 2012).

In the literature, there are a number of related studies proposing instance selection methods for obtaining better mining quality. Specifically, Pradhan and Wu (1999) and Jankowski and Grochowski (2004) surveyed several relevant selection techniques, which can be divided into three application-type groups: noise filters, condensation algorithms, and prototype searching algorithms. In addition, extensive comparative experiments were conducted by Wilson and Martinez (2000), García-Pedrajas et al. (2010), and García et al. (2012). Some cutting-edge instance selection algorithms have been identified, such as Decremental Reduction Optimization Procedure 3 (DROP3), and Genetic Algorithms (GA), which make the k-NN classifiers provide better performance over other instance selection methods.

The noise-filtering algorithms are usually based on the nearest neighbor principle to remove data points which do not agree with the majority of its k nearest neighbor. For condensation algorithms, IB3 (Aha et al., 1991) and DROP3 (Wilson and Martinez, 2000) are two representative algorithms. In IB3, instance x from the training set T is added to a new set S if the nearest acceptable instance in S has different class than x, in which acceptability is defined by a confidence interval

$$\frac{p + \frac{z^2}{2n} \pm z\sqrt{\frac{p(p-1)}{n} + \frac{z^2}{2n^2}}}{1 + \frac{z^2}{n}} \quad (1)$$

where z is a confidence factor, p is the classification accuracy of a given instance (while added to S), and n is equal to the number of classification-trials for the given instance (while added to S).

On the other hand, the Decremental Reduction Optimization Procedure 1 (DROP1) uses the following basic rule to decide if it is safe to remove an instance from the instance set S (where $S = T$ originally):

> Remove P if at least as many of its associates in S would be classified correctly without P. (2)

DROP2 starts the process from sorting instances according to their distances from the nearest opposite class instance. The DROP3 algorithm additionally performs the noise filtering approach before starting the DROP2 algorithm.

Finally, the genetic algorithm (GA) (Cano et al., 2003) is one type of prototype searching algorithm. In general, it uses a population of strings (called chromosomes), which encode candidate solutions

[1] In García-Pedrajas et al. (2010) and García et al. (2012), the reduction rates for instance selection by state-of-the-art algorithms over various domain datasets are very high, i.e. about 80% on average. This means that a large amount of data in each dataset is filtered out.

(called individuals) to an optimization problem. Particularly, the genetic information (i.e., chromosome) is represented by a bit string (such as binary strings of 0s and 1s) and sets of bits encode the solution. That is, a chromosome consists of m genes with two possible states: 1 and 0. If the gene is 1, then its associated instance is included in the subset of T represented by the chromosome. S is then evaluated and coded by a chromosome through a chosen fitness function, which is usually based on the 1-nearest neighbor classifier, to measure the classification rate associated with S. The objective of GA is to maximize the classification rate and minimize the number of instances obtained.

As discussed previously, one major limitation of instance selection is the difficulty of processing very large scale (non-stationary) datasets. When the dataset size grows to a certain level, instance selection should be executed again over the larger dataset even though it has already been performed over the previous smaller dataset. Consequently, the computational cost of performing instance selection rapidly increases as the dataset size becomes larger and larger.

2.2. Outlier detection

By definition, outliers or anomalies are patterns in the data that do not conform to a well defined notion of normal behavior. Since the aim of outlier detection is to identify outliers, it is related to the instance selection task where a set of outliers represented by O can be simply obtained, which is based on $T - S$.

In general, there are three fundamental outlier detection approaches. The first one is to determine the outliers with no prior knowledge of the data, which is analogous to unsupervised clustering. The second approach is to model both normality and abnormality, which is analogous to supervised classification where pre-labeled data illustrating what constitutes normal or abnormal are required. The final approach is to model normality only, or in a very few cases to model abnormality, which is analogous to a semi-supervised recognition task (Hodge and Austin, 2004). Outlier detection is usually based on analyzing the boundary between normal and abnormal regions where outliers can be detected if they were outside the normal region (i.e., the first approach). Some supervised learning based detection algorithms can be employed for this task (i.e., the second and third approaches). In particular, predictive models are built for normal vs. anomalous classes. However, the current problem is that the anomalous instances in the training dataset are far fewer than normal ones. In addition, it is usually difficult to obtain accurate and representative labels, especially for the anomalous class. Typically, labeling is often done manually by a human expert, which is very time-consuming (Chandola et al., 2009). Similar to instance selection, performing outlier detection requires very high computational costs for very large scale datasets.

3. ReDD: representative data detection

3.1. The ReDD process

In order to reduce the computational cost for instance selection on continually growing or very large datasets (cf. Section 2.1), where the number of anomalous instances in the training set via manual labeling for (supervised learning based) outlier detection is small (cf. Section 2.2), the ReDD (**Re**presentative **D**ata **D**etection) process is proposed, as shown in Fig. 1.

Given a dataset D, divided into D1 and D2, D1∪D2 = D, where D1 is simply defined as 50% of D. ReDD is composed of two stages: the training dataset generation stage, and the representative data detection stage. The solid lines indicate the first stage which is to generate the training dataset, based on the common instance selection process. That is, given a dataset (D1) the results of performing instance selection include a representative data (RD) dataset and unrepresentative data (URD) dataset. In the literature, the RD dataset has generally been used for the data mining process, while the unrepresentative data has not been further analyzed.

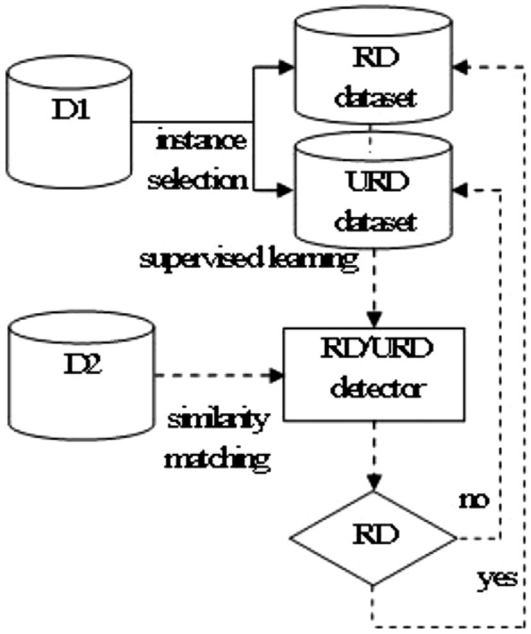

Fig. 1. The ReDD process.

The dotted lines indicate the representative data detection stage comprised of two steps. The first step is based on constructing a classifier by using a training dataset composed of RD and URD datasets produced in the first stage. In other words, the patterns of RD and URD are identified and learned. As a result, a classifier, the RD/URD detector, is developed. However, it should be noted that effectively identifying and learning the RD and URD patterns is heavily dependent on the instance selection and classification techniques used. Therefore, several different instance selection and classification methods will be compared in this paper.

The RD/URD detector used in the second step utilizes a form of similarity matching to classify the 'new' dataset (D2) into one of the RD and URD classes. This step is much more efficient than the conventional approach of directly performing instance selection over the new and larger dataset composed of the old and new data samples, i.e. D.

One simple way to accomplish similarity matching is to apply the k-nearest neighbor approach (Jain et al., 2000) to measure the distances between each new data sample and the training set. The shortest distance between new data and a specific piece of training data determines the class to which the new data belongs. Therefore, if the new data is classified into the RD class, then it is stored into the RD dataset, which is used for the mining purpose as its size continuously increases; otherwise it is stored into the URD dataset. However, in this paper, several different techniques will be discussed to develop the RD/URD detector for comparison.

It should be noted that ReDD is not proposed to compete with existing instance selection algorithms. Instead, ReDD is designed to speed up the instance selection procedure no matter which instance selection algorithm is used. In other words, when a specific instance selection algorithm is chosen as the best or the optimal solution for the instance selection purpose, it can be integrated within ReDD to reduce the computational time when a very large scale dataset is used. In addition, ReDD is specifically designed for large scale datasets. For performing instance selection over small scale datasets, it is unlikely to require large computational time by using existing instance selection algorithms. In this case, to conduct the ReDD procedure based on some instance selection algorithm is unnecessary.

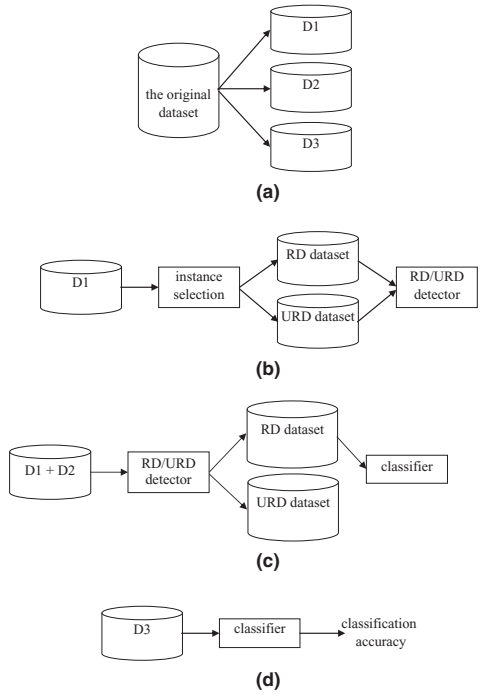

Fig. 2. The operating process of ReDD. (a) Divide the original dataset into training (D1 and D2) and testing (D3) sets, (b) Training the RD/URD detector, (c) Training the classifier using the RD dataset, (d) Testing the classifier.

3.2. The operating process

The operating processes of ReDD are shown in Fig. 2. To simulate a real world problem, where instance selection is inevitably performed over large scale datasets, we divide each dataset into three subsets (Fig. 2(a)), namely D1, D2, and D3, where D1 and D2 represent the data collected in T1 and T2, respectively, and are used as the training sets (T2 is more recent than T1), and D3 is the testing set used to test the classifier's performance. Note that one can also regard this problem under a static environment meaning that D1 + D2 represent the training set and D3 is the testing set. In this study, we randomly select 40% of the original dataset for D1 and D2 individually and the final 20% for D3. Moreover, this partitioning strategy was run five times in order to avoid bias in the classification result. Therefore, the final classification accuracy reported in this paper is based on averaging the five results produced by five different training and testing sets, respectively.

First of all, instance selection is performed to identify RD and URD datasets over D1 (Fig. 2(b)). Similar to the previous study, IB3, DROP3, and GA are used for the task of instance selection. Then, the RD and URD datasets are used to construct the RD/URD detector with some classification technique.

Next, the RD/URD detector is used to replace conventional instance selection and to identify RD and URD datasets over D1 and D2, respectively (Fig. 2(c)). Note that the RD and URD datasets in Fig. 2(b) and (c) should be different. The resultant RD datasets identified by the RD/URD detector from D1 and D2 are used as the training set to construct the classifier. Finally, D3 is used to test the classification accuracy of the classifier. In this paper, the support vector machine (SVM) classifier is used. Therefore, the following classification results presented in Sections 4.1 and 4.2 mean the classification performance of SVM over D3.

To find the baseline, which is different from our ReDD approach (shown in Fig. 3), a specific instance selection algorithm is utilized over D1 + D2 for each very large scale dataset to identify the RD dataset which is used as the training set to train the classifier. Consequently, the computational times to find the baseline and ReDD can be compared in terms of instance selection.

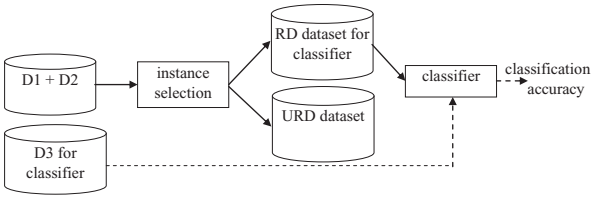

Fig. 3. The baseline operating process.

Table 1
Computational complexities of ReDD and the baselines.

	ReDD	Baselines
GA	$O(\frac{n}{2})^2 + O(n(\log n)^2) + O(n \log n)$	$O(n^2)$
IB3	$O((\frac{n}{2})^2 \log_2 \frac{n}{2}) + O(n(\log n)^2) + O(n \log n)$	$O(n^2 \log_2 n)$
DROP3	$O(\frac{n}{2})^3 + O(n(\log n)^2) + O(n \log n)$	$O(n^3)$

Our aim is to not only to reduce the computational time needed to identify the RD datasets over very large scale datasets, but also ensure that the performance of the classifier based on the ReDD approach is comparable with or even better than the one based on the baseline instance selection approach.

3.3. Computational complexity

The computational complexities of three state-of-the-art algorithms (including GA, IB3, and DROP3) for instance selection are $O(n^2)$, $O(n^2 \log_2 n)$, and $O(n^3)$ respectively (Jankowski and Grochowski, 2004). ReDD additionally includes the processes of training and testing the RD/URD detector. If the detector is based on a CART classifier, the computational complexities of training and testing CART are $O(n(\log n)^2)$ and $O(n \log n)$ respectively. Table 1 shows a comparison of the computational complexities of ReDD and the baselines (i.e., GA, IB3, and DROP3). Note that in this study, the 'n' in ReDD is only half of the 'n' in the baselines and is therefore denoted by $\frac{n}{2}$. In other words, the 'n' can be regarded as the number of data samples in the chosen dataset (D).

Since the level of computational complexity of the training and testing CART in ReDD is very low, the complexities of $O(n(\log n)^2)$ and $O(n \log n)$ can be omitted. In this case, the time complexity of ReDD is less than the baselines.

For example, if n is 100 (i.e., 100 data samples), the computational complexities of DROP3 and ReDD are $(100)^3$ and $(50)^3$, which are 1,000,000 and 125,000 respectively. As 'n' increases, the computational effort required becomes larger and larger. This indicates that theoretically, ReDD can save more time for instance selection when the number of data samples is certainly large.

3.4. Research questions

The proposed ReDD process raises two questions or research objectives. Since using different instance selection algorithms can produce different training sets and different classification techniques have different learning capabilities leading to different detection performances, our first research question is: Can we construct a classifier based on a training set generated by the instance selection result, which is able to distinguish reasonably well between representative and unrepresentative data over a given testing set? In this paper, the detector's performance is measured by the rate of classification accuracy in order to fairly compare with the classification accuracy obtained by the baseline instance selection procedure (cf. Section 4.1).

Table 2
Average classification of SVM based on the BPNN, CART, *k*-NN, NB, and SVM detectors.

	BPNN	CART	*k*-NN	NB	SVM
GA (69.61%)	64.057% (3)	69.326% (1)	61.235% (5)	62.810% (4)	69.135% (2)
IB3 (23.56%)	78.915% (3)	80.389% (1)	77.755% (4)	76.835% (5)	80.056% (2)
DROP3 (46.43%)	79.871% (3)	80.581% (1)	79.436% (4)	72.083% (5)	80.143% (2)

Table 3
Average classification of BPNN, CART, *k*-NN, NB, and SVM over 15 larger datasets.

	BPNN	CART	*k*-NN	NB	SVM
GA	54.805% (3)	57.838% (1)	52.707% (5)	54.210% (4)	57.250% (2)
IB3	81.258% (3)	82.955% (2)	77.774% (4)	76.563% (5)	82.976% (1)
DROP3	85.562% (4)	86.816% (1)	86.027% (3)	80.253% (5)	86.561% (2)

Table 4
Average classification of BPNN, CART, *k*-NN, NB, and SVM over 2-class datasets.

	BPNN	CART	*k*-NN	NB	SVM
GA	65.475% (3)	71.908% (1)	62.784% (5)	63.837% (4)	71.682% (2)
IB3	77.460% (4)	79.285% (1)	76.803% (5)	78.550% (3)	79.209% (2)
DROP3	76.745% (3)	77.914% (2)	75.213% (4)	71.023% (5)	78.220% (1)

Furthermore, the second research question is: If some existing classification technique(s) used as the detector(s) in the ReDD process could perform similar or better than the baseline instance selection approach, can ReDD based on the better detector(s) provide higher classification accuracy than the baseline over larger scale datasets (cf. Section 4.2)?

4. Experiments

4.1. Study one: Research Question 1

4.1.1. Experimental setup

In study one, 50 different domain datasets from the UCI Machine Learning Repository[2] were selected for the experiments (García-Pedrajas et al., 2010). The numbers of data samples and input variables of these datasets ranged from 57 to 1099 and from 4 to 101, respectively. In addition, the number of classes to be classified ranged from 2 to 29.

For each dataset, three well-known instance selection algorithms are used to generate the RD and URD datasets: IB3 (Aha et al., 1991), DROP3 (Wilson and Martinez, 2000), and a genetic algorithm (GA) (Cano et al., 2003).

Next, five popular classification techniques are used to construct the classifiers for the RD/URD detectors, namely the back-propagation neural network (BPNN), CART decision tree, *k*-nearest neighbor (*k*-NN), naïve Bayes (NB), and support vector machine (SVM) (Wu et al., 2008) techniques. Note that these classifiers are constructed with the WEKA data mining software (Witten and Frank, 2005) and WEKA's default settings are used for all parameters.

4.1.2. Experimental results

Table 2 shows the average classification accuracy of SVM based on the five constructed classifiers as the RD/URD detectors over the 50 datasets generated by GA, IB3, and DROP3. Note that the values followed by the instance selection algorithms mean the average reduction rates.[3] As one can see, the CART and SVM classifiers perform best and second best for detecting (un)representative data. On the other hand, since GA filters out the most data samples, this indicates that over-selection occurs in performing GA, and this leads to lower classification accuracy.

According to García-Pedrajas et al. (2010), the average classification accuracy of the 1-NN classifier without performing instance classification is 79.47% over the 50 datasets. This shows that CART and SVM perform reasonably well and have the potential to be used as the (un)representative data detectors. However, we believe that CART is much more suitable for the ReDD process since it can not only make SVM provide the highest classification rate on the basis of these three well-known instance selection algorithms, but also produce some decision rules for the patterns of (un)representation data, which are easily interpreted by a knowledge-domain expert (i.e., "IF-THEN" rules).

To demonstrate the suitability of using CART as the (un)representative data detector, we further examine some specific classification results over larger datasets among the 50 datasets (i.e., where the number of data samples is larger than 1000) and 2-class datasets (i.e., binary classification), which are shown in Tables 3 and 4, respectively. As we can see, CART can provide the best and second best classification accuracy no matter which instance selection algorithm is used in the ReDD process.

In short, we can identify the best classifier (i.e., CART) as the detector for the ReDD process for distinguishing between representative and unrepresentative data over different datasets. However, the classification rates of the classifiers depend on the training dataset generated from the used instance selection algorithm.

4.2. Study two: Research Question 2

4.2.1. Experimental setup

To answer the second research question in the second experimental study, four very large scale datasets are used. They are the KDD Cup[4] 2004 (Protein prediction) and 2008 (Breast cancer), Person activity,[5] and Covertype[6] datasets. Table 5 lists the basic information for these four datasets.

As identified in Section 4.1.2, CART performs the best as the RD/URD detector over the small scale datasets. Two other detectors, which are *k*-NN and SVM are also used for comparison.

[2] http://archive.ics.uci.edu/ml/.

[3] The reduction rate represents the percentage of the data sample, which are filtered out from a given training set.

[4] http://www.sigkdd.org/kddcup/.

[5] http://archive.ics.uci.edu/ml/datasets/Localization+Data+for+Person+Activity.

[6] http://archive.ics.uci.edu/ml/datasets/Covertype.

Table 5
Basic information for the four datasets.

Datasets	No. of features	No. of samples	No. of classes
Breast cancer	117	102,294	2
Covertype	54	581,012	7
Person activity	8	164,860	11
Protein prediction	74	145,751	2

4.2.2. Experimental results

Table 6 shows the average classification accuracy of CART, *k*-NN, and SVM over the four very large scale datasets created by the ReDD and baseline approaches based on IB3, DROP3, and GA respectively. Note that the underlined numbers mean the best performance in each dataset. Similar to Table 2, the values followed by the instance selection algorithms represent the reduction rates over the four datasets, respectively.

We can see that the final classification accuracies of the baseline instance selection methods and ReDD are very similar. In most cases, the performance difference in average classification accuracy obtained with the baseline and ReDD approaches are less than 1%. This demonstrates the robustness of the ReDD approach, which does not need to take the full training set into account, but can still maintain the final classification accuracy with the three different baseline instance selection methods.

Furthermore, Table 7 shows the average processing times for the baselines and ReDD, including the times for training and testing the detector. Note that detector training and testing in ReDD is based on CART. As we can see, on average, it is nearly two or three times faster to use the ReDD approach than the baselines.

In summary, our experimental results demonstrate the effectiveness and efficiency of the proposed ReDD approach for large scale instance selection. More specifically, we show that supervised learning techniques can be trained to effectively detect (un)representative data samples for the instance selection purpose. In addition, only half of the original training set is needed when using the ReDD approach for large scale instance selection, which greatly reduces the time complexity during instance selection. Moreover, the ReDD approach maintains the final classification accuracy of the baseline instance selection methods over all of the training data samples.

4.3. Sensitivity analysis

To further understand the effect of different sizes of D1 on the performance of ReDD, seven different dataset sizes of D1 are considered for the comparison. They are 50%, 30%, 10%, 5% of D1 and 1000, 500, and 100 data samples of D1. In addition, IB3 and DROP3 are used for the instance selection task for the detector, which is based on CART. The reasons of not using GA is because the classification performance of using GA is similar to the ones of using IB3 and DROP3, and performing GA for instance selection requires very large computational cost. Fig. 4 shows the comparative results over the four datasets.

As we can see, when the dataset sizes of D1 become smaller, the final classification accuracy degrades gradually. This indicates that although using certainly fewer data samples of D1 can largely reduce the computational cost of performing instance selection, it cannot make the classifiers perform similar to the baseline instance selection approach.

However, these results imply that using 30% of D1 still can produce reasonably well classification accuracy. In other words, when the dataset size is extremely large, the classifier based on using 30% of D1 in the ReDD process is likely to perform similar to the one based on the baseline instance selection approach. Moreover, the computational time can be largely reduced, which is shorter than using 50% of D1.

Table 6
Average classification accuracy of CART, *k*-NN, and SVM by ReDD and the baselines.

	Breast cancer		Covertype		Person activity		Protein prediction	
	Baseline	ReDD	Baseline	ReDD	Baseline	ReDD	Baseline	ReDD
Instance selection by IB3 (42.02%/43.8%/18.54%/40.69%)								
CART	79.32%	89.69%	91.96%	91.70%	62.46%	62.65%	91.17%	96.64%
k-NN	77.84%	86.31%	96.06%	95.88%	62.26%	62.59%	81.16%	87.16%
SVM	87.83%	98.46%	48.77%	48.71%	61.15%	61.22%	99.11%	99.07%
Avg.	81.66%	91.49%	78.93%	78.76%	61.96%	62.15%	90.48%	94.29%
Instance selection by DROP3 (10.85%/36.46%/44.89%/38.04%)								
CART	99.38%	99.20%	92.51%	92.83%	67.85%	64.33%	99.51%	99.53%
k-NN	99.44%	99.45%	95.70%	95.87%	69.14%	64.67%	99.38%	99.35%
SVM	99.44%	99.43%	48.66%	48.66%	60.72%	60.04%	99.04%	99.09%
Avg.	99.42%	99.36%	78.96%	79.12%	65.90%	63.01%	99.31%	99.32%
Instance selection by GA (56.96%/69.89%/59.46%/62.44%)								
CART	99.31%	99.05%	91.13%	90.31%	62.53%	61.58%	99.48%	99.54%
k-NN	99.35%	99.35%	95.87%	95.54%	62.97%	62.35%	99.20%	99.16%
SVM	99.36%	99.46%	48.63%	48.64%	60.74%	60.68%	99.08%	99.08%
Avg.	99.34%	99.29%	78.54%	78.16%	62.08%	61.54%	99.25%	99.26%

Table 7
Average processing times for baselines and ReDD (hours).

	Breast cancer		Covertype		Person activity		Protein prediction	
	Baseline	ReDD	Baseline	ReDD	Baseline	ReDD	Baseline	ReDD
IB3	12.13	3.17	1997.81	996.39	2018.51	507.48	35.47	9.25
DROP3	455.29	227.86	2206.84	1045.47	457.2	228.32	973.53	485.4
GA	1839.43	1408.03	2812.55	1408.03	1332.89	323.44	2316.63	594.93
Avg.	768.95	546.35	2339.07	1149.96	1269.53	353.08	1108.54	363.19

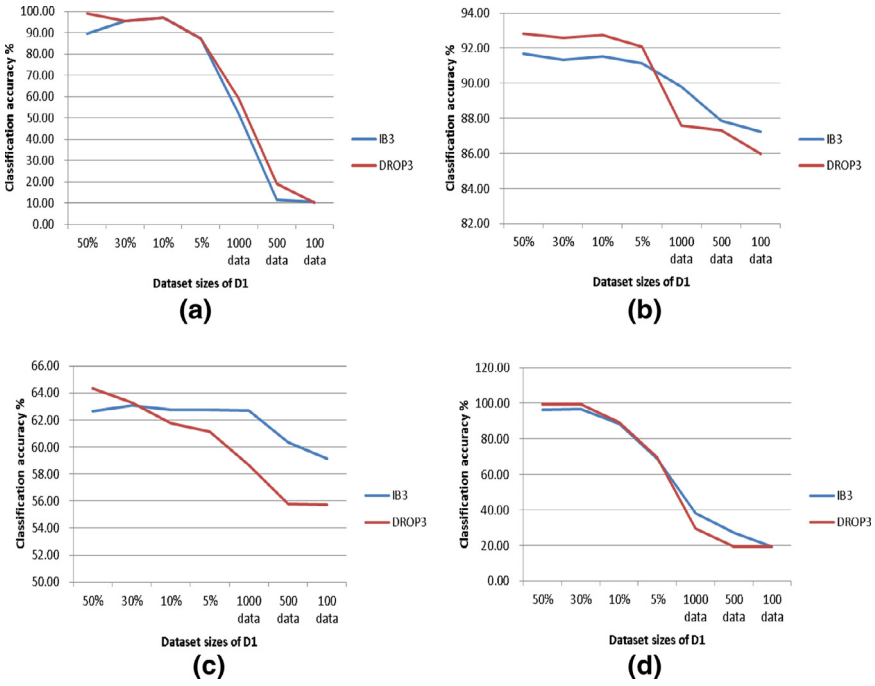

Fig. 4. The classification results of using different dataset sizes of D1. (a) Breast cancer, (b) Covertype, (c) Person activity, (d) Protein prediction.

5. Conclusion

It is difficult for the current instance selection algorithms to efficiently handle very large scale datasets or non-stationary datasets composed of old data and very large amounts of new data samples. In most past studies, more attention has been paid to effective instance selection (or outlier detection) with a reduced or 'clean' dataset being produced by discarding detected outliers prior to the later data mining process, such as classification or clustering. However, even though the discarded instances (or outliers) are often bad data points, they can contain valuable information, which if removed, is never analyzed in instance selection.

To solve the scalability problem for large scale instance selection, our proposed ReDD approach focuses on learning the patterns of the outliers. Specifically, analysis of unrepresentative data (or outliers) is based on identifying patterns in the outliers, such as rule extraction from outliers or training a machine learning model, which can then be used to detect whether the new data samples are (un)representative data.

The results of our first experiment show that current machine learning techniques trained based on a training dataset composed of representative and unrepresentative data samples identified by current instance selection algorithms can effectively detect (un)representative data in new unknown datasets. The second set of experimental results (carried out over four very large scale datasets) demonstrates the effectiveness and efficiency of the ReDD approach. In particular, the final classification accuracy of the three baseline instance selection algorithms and ReDD are very similar, but the time complexity of ReDD is nearly two to three times less than for the baselines.

Several issues can be considered in future work. First, since analysis of unrepresentative data is based on the detection of outliers, the effectiveness of the instance selection algorithm is a very important issue. That is, more sophisticated algorithms can be applied in ReDD. Second, how to effectively extract useful rules from the outliers and what are important outlier patterns are the critical factors for successful unrepresentative data analysis. Last but not least, classifying new data samples into representative and unrepresentative data can facilitate the instance selection task. There are many types of classification techniques that can be applied to this problem.

References

Aggarwal, C.C., Yu, P.S., 2001. Outlier detection for high dimensional data. In: Proceedings of the ACM SIGMOD International Conference on Management of Data. Santa Barbara, California, pp. 37–46.

Aha, D.W., Kibler, D., Albert, M.K., 1991. Instance-based learning algorithms. Mach. Learn. 6 (1), 37–66.

Barneett, V., Lewis, T., 1994. Outliers in Statistical Data, third ed. John Wiley & Sons.

Cano, J.R., Herrera, F., Lozano, M., 2003. Using evolutionary algorithms as instance selection for data reduction: an experimental study. IEEE Trans. Evol. Comput. 7 (6), 561–575.

Chandola, V., Banerjee, A., Kumar, V., 2009. Anomaly detection: a survey. ACM Comput. Surv. 41 (3) article 15.

Chang, Y.-W., Hsieh, C.-J., Chang, K.-W., Ringgaard, M., Lin, C.-J., 2010. Training and testing low-degree polynomial data mappings via linear SVM. J. Mach. Learn. Res. 11, 1471–1490.

Edakunni, N.U. and Vijayakumar, S., 2009. Efficient online classification using an ensemble of Bayesian linear logistic regressors. In: International Workshop on Multiple Classifier Systems, pp. 102–111.

García, S., Derrac, J., Cano, J.R., Herrera, F., 2012. Prototype selection for nearest neighbor classification: taxonomy and empirical study. IEEE Trans. Pattern Anal. Mach. Intell. 34 (3), 417–435.

García-Pedrajas, N., del Castillo, J.A.R., Ortiz-Boyer, D., 2010. A cooperative coevolutionary algorithm for instance selection for instance-based learning. Mach. Learn. 78, 381–420.

Hodge, V.J., Austin, J., 2004. A survey of outlier detection methodologies. Artif. Intell. Rev. 22, 85–126.

Jain, A.K., Duin, R.P.W., Mao, J., 2000. Statistical pattern recognition: a review. IEEE Trans. Pattern Anal. Mach. Intell. 22 (1), 4–37.

Jankowski, N. and Grochowski, M., 2004. Comparison of instances selection algorithms I: algorithms survey. In: International Conference on Artificial Intelligence and Soft Computing, pp. 598-603.

Knorr, E.M., Ng, R., Tucakov, V., 2000. Distance-based outliers: algorithms and applications. VLDB J. 8, 237–253.

Li, X.-B., Jacob, V.S., 2008. Adaptive data reduction for large-scale transaction data. Eur. J. Oper. Res. 188 (3), 910–924.

Liu, H., Motoda, H., 2001. Instance Selection and Construction for Data Mining. Kluwer.

Pradhan, S. and Wu, X., 1999. Instance selection in data mining. Technical report, Department of Computer Science, University of Colorado at Boulder.

Pyle, D., 1999. Data Preparation for Data Mining. Morgan Kaufmann.

Reinartz, T., 2002. A unifying view on instance selection. Data Min. Knowl. Discovery 6, 191–210.

Wilson, D.R., Martinez, T.R., 2000. Reduction techniques for instance-based learning algorithms. Mach. Learn. 38, 257–286.

Witten, I.H., Frank, E., 2005. Data Mining: Practical Machine Learning Tools and Techniques, second ed Morgan Kaufmann.

Wu, X., Kumar, V., Quinlan, J.R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G.J., Ng, A., Liu, B., Yu, P.S., Zhou, Z.-H., Steinbach, M., Hand, D.J., Steinberg, D., 2008. Top 10 algorithms in data mining. Knowl. Inf. Syst. 14 (1), 1–37.

Wei-Chao Lin is an assistant professor at the Department of Computer Science and Information Engineering, Hwa Hsia University of Technology, Taiwan. His research interests are machine learning and artificial intelligence applications.

Chih-Fong Tsai received a PhD at School of Computing and Technology from the University of Sunderland, UK in 2005. He is now a professor at the Department of Information Management, National Central University, Taiwan. His current research focuses on data mining and machine learning. He has published more than 50 professional publications where some were published in prestigious journals including: *ACM Transactions on Information Systems, ACM Transactions on Management Information Systems, Decision Support Systems, IEEE Transactions on Systems, Man and Cybernetics – Part C: Applications and Reviews, Information Processing & Management, Journal of Systems and Software, Journal of the Association for Information Science and Technology*.

Shih-Wen Ke is now an assistant professor at the Department of Information and Computer Engineering, Chung Yuan Christian University, Taiwan. His current research covers information retrieval and machine learning.

Chia-Wen Hung received her Master's degree from the Department of Information Management, National Central University. Her research interests cover machine learning and data mining applications.

William Eberle is currently an associate professor in the Department of Computer Science at Tennessee Technological University. His research areas of interest include data mining, artificial intelligence, and machine learning.