Knowledge-Based Systems 66 (2014) 28–35

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Detecting potential labeling errors for bioinformatics by multiple voting

Donghai Guan [a,b], Weiwei Yuan [a,c,*], Tinghuai Ma [d], Sungyoung Lee [e]

[a] *College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China*
[b] *College of Automation, Harbin Engineering University, China*
[c] *College of Computer Science and Technology, Harbin Engineering University, China*
[d] *School of Computer & Software, Nanjing University of Information Science & Technology, China*
[e] *Dept. of Computer Engineering, Kyung Hee University, Republic of Korea*

ARTICLE INFO

Article history:
Received 7 June 2013
Received in revised form 31 March 2014
Accepted 7 April 2014
Available online 18 April 2014

Keywords:
Bioinformatics analysis
Mislabeled data detection
Single-voting
Multiple-voting
Classification

ABSTRACT

Classification techniques are important in bioinformatics analysis as they can separate various bioinformatical data into distinct groups. To obtain good classifiers, accurate labeling of the training data is required. However labeling in practical bioinformatics applications might be erroneous due to various reasons. To identify those mislabeled data, an ensemble learning based scheme, single-voting has been widely used. It generates multiple classifiers and makes use of their voting to detect mislabeled data. Single-voting scheme mainly consists of two components: data partitioning component to generate multiple classifiers, and mislabeled detection component to identify mislabeled data. Existing works in this field mainly focus on mislabeled detection part and neglect data partitioning. However, our analysis shows that data partitioning plays an important role in single-voting scheme. This analysis helps us proposing a novel multiple-voting scheme. It is superior to traditional single-voting by reducing the unreliable influence from data partitioning. Empirical and theoretical evaluations on a set of bioinformatics datasets illustrate the utility of our proposed scheme.

1. Introduction

Classification techniques are widely used for bioinformatics data analysis [1–5]. It can separate bioinformatics data with similar features into distinct sets, which can support many applications. In classification, a training set is required to train a classifier, which can be used later to classify new data. To obtain a satisfied classifier, the training data is generally required to be with accurate features and labels.

However, in the field of bioinformatics, mislabeling of training data is usually present mainly due to two reasons including subjective nature of the labeling task and the insufficient information to determine the true label. Subjective mislabeling occurs when experts give the labeling according to their personal judgments. The annotations provided by multiple experts might disagree with the general consensus, which leads to mislabeling errors. For example, in [6], 9 mislabeled samples are detected from 49 breast tumor training data. The other source of mislabeling is from insufficient information. For example, a physician may not be able to make the right diagnosis if certain expensive medical procedures are missing.

Existing study [7] has shown that even a small number of mislabeled data could dramatically degrade the performance of the obtained classifier. This has attracted many researchers to develop various techniques to address this issue [8–22]. Existing methods can be classified into two groups: robust classifier designing [8,9] and mislabeled data detecting [10–22]. Robust classifier designing mainly focuses on developing novel classifiers which are robust to mislabeled data during model training. While, mislabeled data detection is to detect and remove mislabeled data prior to training. Our study focuses on mislabeled data detection techniques, which mainly consists of two types: *k*-nearest neighbor based and ensemble learning based.

The core idea of *k*-nearest neighbor (kNN) based algorithms is to compare the label of one sample with the labels of its surrounding neighbors [10]. If there is strong inconsistency among these labels, this training sample is treated as mislabeled. One problem with this approach is from the limitation of kNN algorithm. Not every data distribution is suitable for kNN based method. There are some data distributions wherein the neighbor samples have different labels. Moreover, this group of algorithms does not propagate the mislabeling information to the detection

[*] Corresponding author at: College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China. Tel.: +82 312012950.
E-mail address: yuanweiwei00@khu.ac.kr (W. Yuan).

http://dx.doi.org/10.1016/j.knosys.2014.04.013
0950-7051/© 2014 Elsevier B.V. All rights reserved.

of other training examples, so each training sample is checked independently.

By contrast, ensemble learning based algorithms are used more widely [11–14,16,18,21] for mislabeling detection. The representative algorithms in this group are majority and consensus filtering [13]. In their algorithms, the training data is firstly randomly partitioned into several subsets. Each subset will be checked for mislabeled data separately. The checking is through the voting of multiple classifiers which are trained based on the remaining subsets. These algorithms mainly consists of two steps: data partitioning and multiple classifier voting. As partitioning and voting are executed only once, they are called single-voting scheme in this work.

As an ensemble learning based algorithm, single-voting can achieve accurate mislabeling detection performance based on the voting of multiple classifiers. For single-voting scheme, various elegant voting policies have been proposed, such as majority voting and consensus voting. However, data partitioning, an actual important part of single-voting, is usually neglected. So far random partitioning (randomly partition training data into several subsets) is widely used as it has various advantages. But on the other hand, our analysis has shown that its randomness property makes single-voting unreliable. Some successful detected mislabeled data under one partitioning case are failed to identify when the partitioning changes.

To address this issue, in this paper, we propose a novel multiple-voting scheme. Multiple-voting consists of several single-voting detectors which are different to each other due to various random partitioning. Multiple-voting is superior to single-voting by alleviating the dependency of mislabeled data detection on data partitioning. We also propose various fusion techniques to combine the decisions from different detectors, including one vote veto, majority voting, and consensus voting. Based on the proposed multiple-voting scheme, new variants of majority filtering and consensus filtering algorithms are proposed.

The comparison of multiple-voting and single-voting is analyzed both theoretically and experimentally. Experimental results indicate that our proposed scheme can effectively improve the performance of single-voting. Straightforwardness is a distinguished advantage of our scheme. It can be easily applied on existing single-voting approaches.

In summary, the main technical contribution is pointing out the limitation of existing single-voting scheme and proposing an efficient multiple-voting scheme with sufficient theoretical proofs for solving it.

2. Related works

Mislabeled training data detection and elimination is crucial to improve the accuracy of classifiers when mislabeling is present in the training set. Various techniques have been proposed, among which, ensemble learning based methods including majority filtering (*MF*) and consensus filtering (*CF*) have been widely used. *MF* utilizes the idea of majority voting, while *CF* utilizes the idea of consensus voting.

The general idea of *MF* and *CF* is as follows: They employ ensemble classifier to detect mislabeled instances by constructing a set of base-level classifiers and then using their classifications to identify mislabeled instances. The general approach is to tag an instance as mislabeled if x of the m base-level classifiers cannot classify it correctly. *MF* tags an instance as mislabeled if more than half of the m base level classifiers classify it incorrectly. *CF* requires that all base-level classifiers must fail to classify an instance as the class given by its training label for it to be eliminated from the training data.

The reason to employ ensemble classifiers in *MF* and *CF* is that ensemble classifier has better performance than each base-level classifier on a dataset if two conditions hold: (1) the probability of a correct classification by each individual classifier is greater than 0.5 and (2) the errors in predictions of the base-level classifiers are independent.

Shown in Table 1, majority filtering begins with n equalsized disjoint subsets of the training set E (step 1) and the empty output set A of detected noisy examples (step 2). The main loop (steps 3–6) is repeated for each training subset E_i . In step 4, subset E_t is formed which includes all examples from E except those in E_i, which then is used as the input an arbitrary inductive learning algorithm that induces a hypothesis (a classifier) H_j (step 6). Those examples from E_i for which majority of the hypotheses does not give the correct classification are added to A as potentially noisy examples (step 14).

Consensus filtering algorithm is shown in Table 2. Its only difference with *MF* is at step 14. In *CF*, the example in E_i is regarded as a noisy example only when all the hypotheses incorrectly classify it. Compared with *MF*, *CF* is more conservative due to the severer condition for noise identification, and which results in fewer instances being eliminated from the training set. The drawback of *CF* is the added risk in retaining bad data.

Majority filtering and consensus filtering are regarded as single-voting detectors. Single-voting detector consists of two steps. The first step is data partitioning. The training data E will be randomly divided into n equal size subsets $(E_1, E_2, \ldots, E_n)$. Then each subset E_i is taken out. Other $n-1$ subsets, $E \setminus E_i$ are used to train k different classifiers based on different classification algorithms. These k classifiers will be used as noise filters to detect the potential mislabeled data in E_i. Each classifier will classify the data in E_i individually. Suppose e is one training data in E_i; its given label is Label_e; its predicted label by classifier C is PLabel_e. If PLabel_e equals to Label_e, then classifier C will treat e as a noise-free data. Otherwise, e will be treated as a mislabeled data. Considering different classifiers (totally num. is k) might have different opinions on e, a voting mechanism is needed to combine their opinions.

3. The proposed multiple voting scheme

In single-voting, the voting of different classifiers can guarantee the reliability for mislabeling detection to some extent. However, it

Table 1
Majority filtering algorithm.

```
Algorithm 1: Majority Filtering (MF)

Input: $E$ (training set)
Parameter: $n$ (number of subjects), $y$ (number of learning algorithms),
$A_1, A_2, \ldots, A_y$ ($y$ kinds of learning algorithms)
Output: A (detected noisy subset of $E$)
(1) form $n$ disjoint almost equally sized subset of $E_i$, where $\bigcup_i E_i = E$
(2) $A \leftarrow \emptyset$
(3) for $i = 1, \ldots, n$ do
  (4) form $E_t \leftarrow E \setminus E_i$
  (5) for $j = 1, \ldots y$ do
    (6) induce $H_j$ based on examples in $E_t$ and $A_j$
  (7) end for
  (8) for every $e \in E_i$ do
    (9) $ErrorCounter \leftarrow 0$
    (10) for $j = 1, \ldots, y$ do
      (11) if $H_j$ incorrectly classifies $e$
      (12) then $ErrorCounter \leftarrow ErrorCounter + 1$
    (13) end for
    (14) if $ErrorCounter > \frac{y}{2}$, then $A \leftarrow A \cup \{e\}$
  (15) end for
(16) end for
```

Table 2
Consensus filtering algorithm.

```
Algorithm 2: Consensus Filtering (MF)

Input: $E$ (training set)
Parameter: $n$ (number of subjects), $y$ (number of learning algorithms), $A_1, A_2, \dots, A_y$ ($y$ kinds of learning algorithms)
Output: $A$ (detected noisy subset of $E$)
form $n$ disjoint almost equally sized subset of $E_i$, where $\bigcup_i E_i = E$
$A \leftarrow \emptyset$
for $i = 1, \dots, n$ do
    form $E_t \leftarrow E \setminus E_i$
    for $j = 1, \dots y$ do
        induce $H_j$ based on examples in $E_t$ and $A_j$
    end for
    for every $e \in E_i$ do
        $ErrorCounter \leftarrow 0$
        for $j = 1, \dots, y$ do
            if $H_j$ incorrectly classifies $e$
            then $ErrorCounter \leftarrow ErrorCounter + 1$
        end for
        if $ErrorCounter = y$, then $A \leftarrow A \cup \{e\}$
    end for
end for
```

neglects the influence of data partitioning. For different data partitioning cases, the trained classifiers are also different as training data changes. Consequently, the classifiers' detection results can also vary. Therefore, it is possible that in partitioning case 1, a mislabeled data could be successfully detected; while in partitioning case 2, the same mislabeled data is failed to be detected. Since the partitioning is random, single-voting is risky and unrilable.

To reduce the effects of data partitioning, a novel multiple-voting scheme is proposed (Fig. 1). As shown in Fig. 1, multiple-voting consists of t single-voting detectors. Each single-voting detector M_i will generate its own decision about suspected mislabeled data index A_i in the 1st layer voting. Finally, in the 2nd layer voting, all the different decisions A_i will be combined to output the final decision A about which data is mislabeled.

In Fig. 1, the 1st layer voting can use either majority voting or consensus voting. In the 2nd layer voting, as our new proposed layer, we have proposed three voting policies for it: one vote veto, majority and consensus voting. One vote veto tags a data as mislabeled if at least one single-voting detector agrees with that. Majority and consensus voting in the 2nd layer are identical to them used in 1st layer voting.

By adopting majority or consensus voting, existing work [9] consists of two methods: majority filtering (*MF*) and consensus filtering (*CF*). Based on our multiple-voting scheme, several new variants are developed (Table 3).

For the new proposed *MF*/*CF* variants, their only difference is how to combine the detection results in the second layer voting. Instead of presenting all the algorithms here, we select MF_{MF} as the representative and present its algorithm in Table 4.

4. Analysis of proposed multiple-voting scheme

In identifying mislabeled instances, two types of error can be made. The first type ($E1$) occurs when declaring a correctly labeled example as mislabeled and is subsequently discarded. The second type of error ($E2$) corresponds to declare a mislabeled example as correctly labeled. In this section we analyze the probability of each of these types of errors for our proposed multiple-voting approaches.

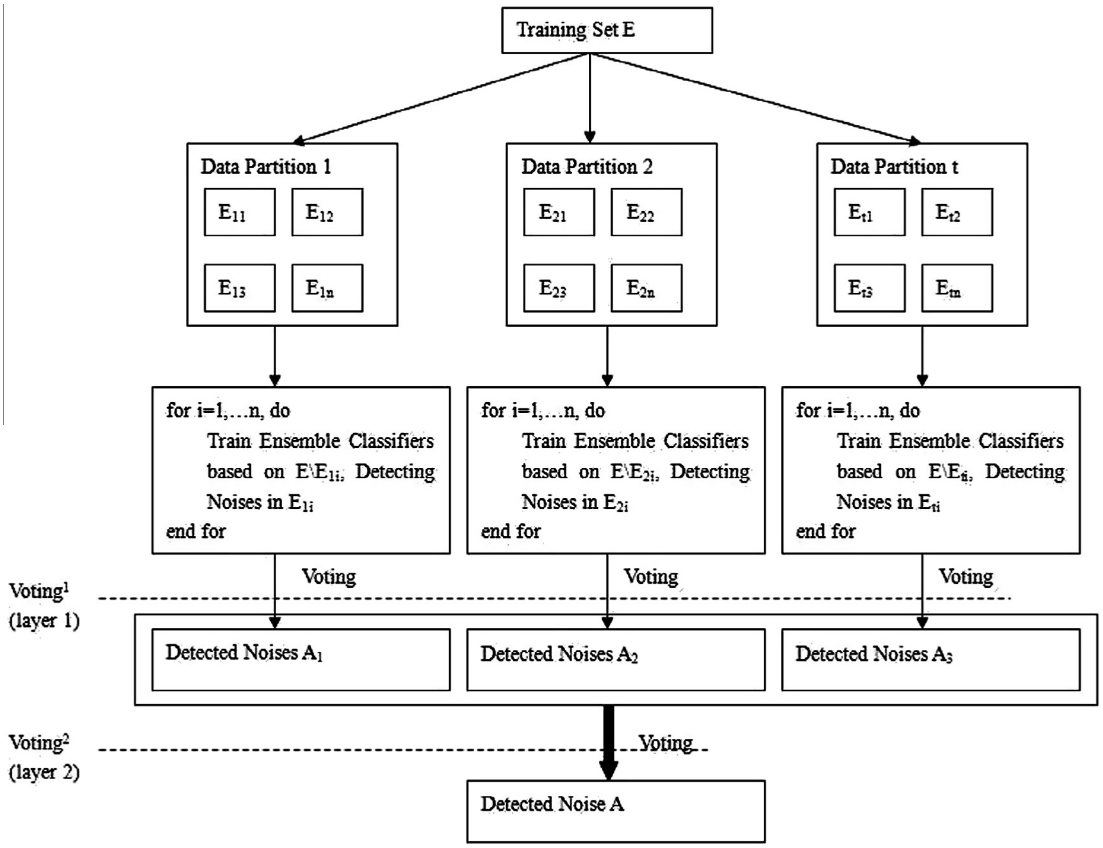

Fig. 1. Multiple-voting based mislabeled data detection scheme.

Table 3
Our proposed multiple-voting based methods.

Proposed methods		1st Layer voting policy	2nd Layer voting policy
MF variants	MF_1	Majority voting	One vote veto
	MF_{MF}	Majority voting	Majority voting
	MF_{CF}	Majority voting	Consensus voting
CF variants	CF_1	Consensus voting	One vote veto
	CF_{MF}	Consensus voting	Majority voting
	CF_{CF}	Consensus voting	Consensus voting

Table 4
Proposed MF_{MF} algorithm.

```
Algorithm 3: MajorityFiltering_MajorityFiltering ($MF_{MF}$)
Input: $E$ (training set)
Parameter: $n$ (number of subsets), $y$ (number of learning algorithms),
$t$ (number of times of subsets partitioning), $A_1, A_2, \ldots, A_y$ ($y$ kinds of learning
   algorithms)
Output: $A$ (detected noisy subset of $E$)
for $p = 1, \ldots, t$ do
    form $n$ disjoint almost equally sized subset of $E_{pi}$, where $\bigcup_i E_{pi} = E$
    $A^p \leftarrow \emptyset$
    for $i = 1, \ldots, n$ do
        form $E_t \leftarrow E \setminus E_{pi}$
        for $j = 1, \ldots y$ do
            induce $H_{pj}$ based on examples in $E_t$ and $A_j$
        end for
        for every $e \in E_{pi}$ do
            $ErrorCounter \leftarrow 0$
            for $j = 1, \ldots, y$ do
                if $H_{pj}$ incorrectly classifies $e$
                then $ErrorCounter \leftarrow ErrorCounter + 1$
            end for
            if $ErrorCounter > \frac{y}{2}$, then $A^p \leftarrow A^p \cup \{e\}$
        end for
    end for
end for
$A \leftarrow \emptyset$
for every $e \in E$ do
    $ErrorCounter \leftarrow 0$
    for $j = 1, \ldots, p$ do
        if $e \in A^p$
        then $ErrorCounter \leftarrow ErrorCounter + 1$
    end for
    if $ErrorCounter > \frac{p}{2}$, then $A \leftarrow A \cup \{e\}$
end for
```

Let $P(E1_i)$ and $P(E2_i)$ be the probability that classifier i makes an $E1$ and $E2$ error respectively. To clarity the analysis, it is assumed that all m various classifiers have the same probability of making an $E1$ error that is equal to $P\ (E1)$. The same assumption is for the $P(E2_i)$ that is equal to $P\ (E2)$.

4.1. Analysis of MF and our proposed MF variants

For majority filtering (*MF*), it makes an $E1$ (or $E2$) error when more than half of these m classifiers fail to classify the instance correctly. Therefore,

$$P(E1_{MF}) = \sum_{j>m/2}^{j=m} P(E1)^j (1 - P(E1))^{m-j} \binom{m}{j}$$

$$P(E2_{MF}) = \sum_{j>m/2}^{j=m} P(E2)^j (1 - P(E2))^{m-j} \binom{m}{j}$$

We have proposed three *MF* variants MF_1,MF_{MF}, and MF_{CF} which run *MF* for several times (suppose this value is t) and combine the results based on one vote veto, majority voting, and consensus voting respectively. Suppose $P(E1_{MF_i})$ and $P(E2_{MF_i})$ be the probability that single-voting detector MF_i makes an $E1$ and $E2$ error respectively. To simply the analysis, we assume that each $P(E1_{MF_i})$ is identical and equals to $P(E1_{MF})$. Each $P(E2_{MF_i})$ is also identical and equals to $P(E2_{MF})$.

MF_1 will make an $E1$ error if there is at least one *MF* mistakenly declares the instance as mislabeled. Or we can say $E1$ error will be made except all the *MF* detectors do not make this mistake. On the other hand, it will make an $E2$ error only if all the MF_i mistakenly declare the mislabeled instance as correctly labeled one. Therefore,

$$P(E1_{MF_1}) = 1 - (1 - P(E1_{MF}))^t$$

$$P(E2_{MF_1}) = P(E2_{MF})^t$$

MF_{MF} will make an $E1$ (or $E2$) error when more than half of these j *MF* detectors make an error. Therefore,

$$P(E1_{MF_{MF}}) = \sum_{j>t/2}^{j=t} P(E1_{MF})^j (1 - P(E1_{MF}))^{t-j} \binom{t}{j}$$

$$P(E2_{MF_{MF}}) = \sum_{j>t/2}^{j=t} P(E2_{MF})^j (1 - P(E2_{MF}))^{t-j} \binom{t}{j}$$

MF_{CF} will make an $E1$ error only when all the j *MF* detectors make this $E1$ error. It will make an $E2$ error if there is at least one *MF* detector makes this $E2$ error. Therefore,

$$P(E1_{MF_{CF}}) = P(E1_{MF})^t$$

$$P(E2_{MF_{CF}}) = 1 - (1 - P(E2_{MF}))^t$$

Because each MF_i tags a training sample independently based on the random data partitioning, the mistakes they make can be regarded as independent of each other. Therefore, if $P(E1_{MF})$ and $P(E2_{MF})$ are less than 0.5, we have the following relationship with above probabilities:

(1) $P(E1_{MF_{CF}}) < P(E1_{MF_{MF}}) < P(E1_{MF}) < P(E1_{MF_1})$.

(2) $P(E2_{MF_1}) < P(E2_{MF_{MF}}) < P(E2_{MF}) < P(E2_{MF_{CF}})$.

Since $P(E) = P(E1) + P(E2)$, thus, we have $P(E_{MF_{MF}}) < P(E_{MF})$. It means mathematically, MF_{MF} can make few errors than *MF*. For $P(E_{MF_1})$ and $P(E_{MF_{CF}})$, Compared to *MF*, they make less mistakes for one type of error, but simultaneously make more mistakes for the other type of error. Therefore, it is hard to judge whether they are better than *MF*. It depends on whether their improvements on one type of error can complement their loss on the other type of error. This will be tested through experiments in the following section.

4.2. Analysis of CF and our proposed CF variants

The notations in this part are same to those in Section 4.1. Meanwhile, the assumptions for problem analysis are also identical.

We have the following probabilities of errors for each *CF* related methods.

$$P(E1_{CF}) = P(E1)^m$$

$$P(E2_{CF}) = 1 - (1 - P(E2))^m$$

$$P(E1_{CF_1}) = 1 - (1 - P(E1_{CF}))^t$$

$$P(E2_{CF_1}) = P(E2_{CF})^t$$

$$P(E1_{CF_{MF}}) = \sum_{j>t/2}^{j=t} P(E1_{CF})^{j}(1 - P(E1_{CF}))^{t-j}\binom{t}{j}$$

$$P(E2_{CF_{MF}}) = \sum_{j>t/2}^{j=t} P(E2_{CF})^{j}(1 - P(E2_{CF}))^{t-j}\binom{t}{j}$$

$$P(E1_{CF_{CF}}) = P(E1_{CF})^{t}$$

$$P(E2_{CF_{CF}}) = 1 - (1 - P(E2_{CF}))^{t}$$

The following relationships can be summarized from above probabilities,

(1) $P(E1_{CF_{CF}}) < P(E1_{CF_{MF}}) < P(E1_{CF}) < P(E1_{CF_1})$.

(2) $P(E2_{CF_1}) < P(E2_{CF_{MF}}) < P(E2_{CF}) < P(E2_{CF_{CF}})$.

Thus, we have$P(E_{CF_{MF}}) < P(E_{CF})$. For $P(E_{CF_1})$ and $P(E_{CF_{CF}})$, Compared to *CF*, they make less mistakes for one type of error, but simultaneously make more mistakes for the other type of error. Therefore, it is hard to judge whether they are better than *CF*. It depends on whether their improvements on one type of error can complement their loss on the other type of error. This will be also be tested through experiments in the following section.

5. Experimental work

5.1. Datasets

Nine bioinformatics datasets are used in this work (Table 5). All of these datasets are obtained from the well-known UCI Repository (http://archive.ics.uci.edu/ml/). The purpose of each dataset is as follows: Parkinson (Discriminate healthy people from those with Parkinson), Iris (Classify iris plants to iris setosa, iris virginica, and iris versicolor), WDBC (Classify breast mass to malignant or benign), Heart disease (Presence of heart disease or not), Diabetes (Diabetes test is positive or negative), Breast cancer (Classify Wisconsin breast cancer data into malignant or benign), Cardiotocography (Classify fetal cardiotocograms to different fetal states: normal, suspect, and pathologic), Acute Inflammations1 (perform the presumptive diagnosis of diseases of urinary system (Inflammation of urinary bladder)), Acute Inflammations2 (perform the presumptive diagnosis of diseases of urinary system (Nephritis of renal pelvis origin)).

5.2. Experimental configurations and results

To evaluate the effectiveness of the proposed multiple-voting based mislabeling detection scheme, we compare our proposed *MF* variants and *CF* variants with conventional *MF* and *CF*.

Refer to Fig. 1, the experimental comparisons are configured as follows: data is partitioned into three subsets ($n = 3$); single-voting is executed for ten times ($t = 10$); to train multiple classifiers, three algorithms are used including naïve Bayes, decision tree, and k-NN ($k = 3$).

In experiments, each dataset was divided into a training set and a test set. Training set includes mislabeled data. Each mislabeling detection algorithm filtered mislabeled data from the training set, and the performances of each algorithm were evaluated using the test set. Classification accuracy has been widely used in previous studies evaluating mislabeled data detection performance. In this study, the k-nearest neighbor ($k = 3$) was used. When two noise detection methods are applied to the same dataset with the same kNN algorithm, higher classification accuracy indicates better noise detection performance.

Table 5
Datasets used in this work.

Data name	# Of samples	# Of features
Parkinson	197	23
Iris	150	4
WDBC	569	31
Heart disease	303	13
Diabetes	768	8
Breast cancer	699	9
Cardiotocography	2126	23
Acute Inflammations1	120	6
Acute Inflammations2	120	6

To determine classification accuracy, each dataset *D* was processed as follows:

- Three trials derived from threefold cross-validation of *D* were used to evaluate the performance of each feature selection algorithm. During each trial, 66.6% of *D*, or Tr, was used as a training set. The remaining 33.3% of *D*, or Ts, was used as a test set to evaluate the classification accuracy of each class noise detection method. We artificially changed some labels that were originally correct in Tr, according to predefined mislabeled ratios to generate mislabeled data. We considered four different mislabeled ratios: 10%, 20%, 30%, and 40%. For example, if we wanted to evaluate the classification on Tr under a 10% mislabeled ratio, we randomly selected 10% of the samples from Tr and changed correct labels to incorrect labels.
- The average classification accuracy was obtained by averaging the accuracies of three trials.
- Considering that the partitioning of *D* and that the mislabeled data generated could influence average classification accuracy, we executed each experiment 10 times for 10 classification accuracies (executed the previous two steps 10 times).
- Finally, the reported accuracy was calculated as the average of these 10 values.

The performances of each mislabeled data detection method on Parkinson, Iris, and Wdbc are shown in Table 6. The noise ratio for Parkinson is up to 30% because the classification accuracy is too low to consider when the noise ratio is 40%.

Table 6
Performance of each mislabeled data detection method on Parkinson, Iris, Wdbc.

Noise	Class noise detection algorithms							
	MF related				*CF* related			
	MF	MF_1	MF_{MF}	MF_{CF}	CF	CF_1	CF_{MF}	CF_{CF}
Dataset1: Parkinson								
10%	0.755	0.745	0.790	0.763	0.741	0.743	0.749	0.726
20%	0.690	0.691	0.698	0.639	0.614	0.662	0.623	0.562
30%	0.510	0.550	0.579	0.491	0.454	0.564	0.476	0.404
Ave.	0.652	0.662	0.689	0.631	0.603	0.656	0.616	0.564
Dataset2: Iris								
10%	0.936	0.929	0.937	0.940	0.933	0.939	0.940	0.928
20%	0.933	0.929	0.936	0.921	0.915	0.927	0.913	0.876
30%	0.892	0.883	0.916	0.873	0.844	0.897	0.863	0.797
40%	0.860	0.832	0.871	0.847	0.825	0.872	0.843	0.741
Ave.	0.905	0.893	0.915	0.895	0.879	0.909	0.890	0.836
Dataset3: Wdbc								
10%	0.968	0.966	0.968	0.965	0.953	0.971	0.955	0.928
20%	0.962	0.974	0.972	0.931	0.887	0.953	0.903	0.824
30%	0.938	0.969	0.952	0.908	0.801	0.926	0.833	0.698
40%	0.798	0.953	0.895	0.672	0.614	0.786	0.632	0.501
Ave.	0.917	0.966	0.947	0.869	0.814	0.909	0.831	0.738

Parkinson: as shown in Table 6, in terms of average accuracy, the ranking of *MF* related methods is MF_{MF}, MF_1, *MF*, and MF_{CF}. The ranking of *CF* related methods is CF_1, CF_{MF}, *CF*, and CF_{CF}. For all the different noise ratios, MF_{MF} is always better than *MF*. But MF_1 is only better than *MF* when noise level is 30%. Therefore, in *MF* related methods, only MF_{MF} is better than *MF* when considering both accuracy and robustness. In *CF* related methods, CF_1 and CF_{MF} are better than *CF* in all the different noise ratios. Compared to conventional *MF* and *CF* methods, the improvements of MF_{MF}, CF_1, and CF_{MF} are correlated to the noise ratio. Basically the improvements become more significant when the number of mislabeled samples increases.

Iris: In Table 6, among *MF* related methods, the accuracy ranking is MF_{MF}, *MF*, MF_{CF}, and MF_1. The rank of *CF* related methods is CF_1, CF_{MF}, *CF*, and CF_{CF}. In four various noise levels, MF_{MF}, CF_1, and CF_{MF} are consistently better than original methods. When the noise ratios are small (10% and 20%) the improvements are not very obvious. However the improvements become more significant when the noises are above 20%.

Wdbc: the accuracy ranking of *MF* related methods is MF_1, MF_{MF}, *MF*, and MF_{CF}; the accuracy ranking of *CF* related methods is CF_1, CF_{MF}, *CF*, and CF_{CF}. Compared to *MF*, MF_1 and MF_{MF} give similar or better accuracies in all the four different noise ratios. Similarly, CF_1 and CF_{MF} are better than *CF* under all various noise ratios. For this dataset, when noise ratio is 10%, the improvements of these variants are little. But when noise ratio is above 10%, the improvements become much more significant. For example, when noise ratio is 40%, the accuracy of MF_1 is 0.953 and it is only 0.798 for *MF*.

The performances of each mislabeled data detection method on Heart disease, Diabetes, Breast cancer are shown in Table 7.

Heart disease: Table 7 shows that among *MF* related methods, the ranking of accuracies is MF_1, MF_{MF}, *MF*, and MF_{CF}; among *CF* related methods, the ranking of accuracies is CF_1, CF_{MF}, *CF*, and CF_{CF}. Compared to *MF*, MF_1 and MF_{MF} show better accuracies on all the different ratios. In addition, the improvement is more significant when the noise ratio is higher. For *CF* related methods, the performance of CF_{MF} is similar to *CF*. CF_1 defeats *CF* on all the different ratios. Moreover, the improvement is more obvious when the number of mislabeled samples increases.

Diabetes: as Table 7 shows, in *MF* variants, MF_{MF} is the best one which is better than *MF* in all the different noise ratios, while MF_1's performance is similar to *MF*. In *CF* related methods, both CF_1 and CF_{MF} are better than *CF* in all the noise ratios. CF_1 is the best one in *CF* variants. For all the improved variants, the improvement tends to be significant when the noise ratio is higher.

Table 7
Performance of each mislabeled data detection method on heart disease, diabetes, breast cancer.

Noise	Class noise detection algorithms							
	MF related				*CF* related			
	MF	MF_1	MF_{MF}	MF_{CF}	CF	CF_1	CF_{MF}	CF_{CF}
Dataset4: heart disease								
10%	0.819	0.822	0.823	0.802	0.8	0.814	0.799	0.771
20%	0.786	0.808	0.797	0.759	0.754	0.776	0.754	0.703
30%	0.751	0.768	0.766	0.705	0.697	0.748	0.700	0.648
40%	0.693	0.728	0.714	0.669	0.662	0.708	0.663	0.620
Ave.	0.762	0.782	0.775	0.734	0.728	0.762	0.729	0.686
Dataset5: diabetes								
10%	0.782	0.769	0.785	0.769	0.768	0.778	0.773	0.762
20%	0.753	0.757	0.769	0.757	0.752	0.768	0.761	0.737
30%	0.758	0.758	0.769	0.751	0.738	0.759	0.752	0.713
40%	0.728	0.735	0.746	0.71	0.701	0.723	0.709	0.687
Ave.	0.755	0.755	0.767	0.747	0.74	0.757	0.749	0.725
Dataset6: breast cancer								
10%	0.968	0.972	0.97	0.965	0.964	0.969	0.964	0.948
20%	0.967	0.974	0.971	0.957	0.948	0.967	0.949	0.918
30%	0.957	0.968	0.964	0.917	0.903	0.949	0.902	0.845
40%	0.899	0.947	0.928	0.868	0.843	0.921	0.849	0.771
Ave.	0.948	0.965	0.958	0.927	0.915	0.952	0.916	0.871

Table 8
Performance of each mislabeled data detection method on Cardiotocography, Acute Inflammations1, Acute Inflammations2.

Noise	Class noise detection algorithms							
	MF related				*CF* related			
	MF	MF_1	MF_{MF}	MF_{CF}	CF	CF_1	CF_{MF}	CF_{CF}
Dataset7: Cardiotocography								
10%	0.985	0.984	0.984	0.984	0.977	0.979	0.977	0.973
20%	0.979	0.983	0.983	0.979	0.958	0.970	0.963	0.942
30%	0.971	0.981	0.977	0.958	0.918	0.951	0.933	0.868
40%	0.925	0.971	0.951	0.887	0.839	0.921	0.865	0.736
Ave.	0.965	0.979	0.974	0.952	0.923	0.955	0.935	0.880
Dataset8: Acute Inflammations1								
10%	0.990	0.991	0.992	0.986	0.991	0.999	0.989	0.981
20%	0.948	0.969	0.985	0.942	0.943	0.977	0.957	0.926
30%	0.862	0.855	0.898	0.829	0.829	0.876	0.851	0.786
40%	0.778	0.796	0.782	0.735	0.733	0.815	0.742	0.673
Ave.	0.895	0.903	0.914	0.873	0.874	0.917	0.885	0.842
Dataset9: Acute Inflammations2								
10%	0.991	0.999	0.996	0.986	0.984	0.991	0.986	0.977
20%	0.977	0.988	0.985	0.971	0.967	0.994	0.971	0.945
30%	0.945	0.950	0.948	0.920	0.913	0.964	0.928	0.847
40%	0.812	0.836	0.830	0.768	0.803	0.862	0.795	0.737
Ave.	0.931	0.943	0.940	0.911	0.917	0.953	0.920	0.877

Breast cancer: the ranking of *MF* related methods is MF_1, MF_{MF}, *MF*, and MF_{CF}; the ranking of *CF* related methods is CF_1, CF_{MF}, *CF*, and CF_{CF}. In addition, although MF_1, MF_{MF}, and CF_1 are better than original methods on all the different ratios, the improvements are more when the noise ratio is higher.

Table 8 shows the experimental results on Cardiotocography, Acute Inflammations1, and Acute Inflammations2.

Cardiotocography: the ranking of *MF* related methods is MF_1, MF_{MF}, *MF*, and MF_{CF}; the ranking of *CF* related methods is CF_1, CF_{MF}, *CF*, and CF_{CF}.

Acute Inflammations1: the ranking of *MF* related methods is MF_{MF}, MF_1, *MF*, and MF_{CF}; the ranking of *CF* related methods is CF_1, CF_{MF}, *CF*, and CF_{CF}.

Acute Inflammations2: the ranking of *MF* related methods is MF_1, MF_{MF}, *MF*, and MF_{CF}; the ranking of *CF* related methods is CF_1, CF_{MF}, *CF*, and CF_{CF}.

In Tables 6–8, we have compared various noise detection methods on each individual dataset. We summarize the performances of *MF* variants and *CF* variants in Tables 9 and 10. In these tables, in addition to the average accuracies, there is the other metric, named Scores. For each individual dataset, the best algorithm is assigned

Table 9
The accuracies of *MF* variants in all the different datasets.

Data	MF	MF_1	MF_{MF}	MF_{CF}
MF related class noise detection algorithms				
Parkinson	0.652	0.662	0.689	0.631
Iris	0.905	0.893	0.915	0.895
Wdbc	0.917	0.966	0.947	0.869
Heart	0.762	0.782	0.775	0.734
Diabetes	0.755	0.755	0.767	0.747
Breast	0.948	0.965	0.958	0.927
Cardio	0.965	0.979	0.974	0.952
Acute1	0.895	0.903	0.914	0.873
Acute2	0.931	0.943	0.940	0.911
Ave.	0.859	0.872	0.875	0.848
Scores	0	5	4	−9

Table 10
The accuracies of *CF* variants in all the different datasets.

Data	CF	CF_1	CF_{MF}	CF_{CF}
CF related class noise detection algorithms				
Parkinson	0.603	0.656	0.616	0.564
Iris	0.879	0.909	0.89	0.836
Wdbc	0.814	0.909	0.831	0.738
Heart	0.728	0.762	0.729	0.686
Diabetes	0.74	0.757	0.749	0.725
Breast	0.915	0.952	0.916	0.871
Cardio	0.923	0.955	0.935	0.880
Acute1	0.874	0.917	0.885	0.842
Acute2	0.917	0.953	0.920	0.877
Ave.	0.821	0.863	0.830	0.780
Scores	0	9	0	−9

“+1”, the worst one is assigned “−1”. This metric can reflect the robustness of each method on various datasets.

As shown in Table 9, in *MF* variants, MF_{MF} is the best one in terms of accuracy and the second best one in terms of robustness. MF_1 is also good. But MF_{CF} could not improve the performance. It is the worst one among *MF* variants.

Table 10 shows that CF_1 make significant improvement on *CF* in terms of both accuracy and robustness. For CF_{MF}, for the first three datasets, its improvement on *CF* is obvious; for the other six datasets, the improvement is little. The worst method among *CF* variants is CF_{CF}.

From above experimental analysis, we can make the following conclusions:

(1) Among our proposed *MF* variants, MF_{MF} and MF_1 can make consistently improvement on *MF*. The best one is MF_{MF}.
(2) Among our proposed *CF* variants, CF_{MF} and CF_1 can make consistently improvement on *CF*. The best one is CF_1.
(3) The improvements of these proposed variants become more significant when the number of mislabeled samples increase. As the examples, we show the improvements of MF_{MF} and CF_1 on different noise ratios in Tables 11 and 12. It clearly shows that the improvement is highly correlated to the noise ratio.

The above experiments have verified the good performances of MF_{MF} and CF_1. With the same datasets and experimental configurations, we further compare them with edited nearest neighbors (ENN) [10], a well-known *k*-nearest neighbor based mislabeled detection method. Two ENN methods, ENN1 ($k = 1$) and ENN3 ($k = 3$) are used. The results in Table 13 indicate that the performances of MF_{MF} and CF_1 are significantly better than ENN methods.

Table 11
The improvements of MF_{MF} on different noise ratios.

Data	Noise ratios			
	10%	20%	30%	40%
Improvement of MF_{MF} on MF				
Parkinson	0.045	0.008	0.069	*
Iris	0.001	0.003	0.024	0.011
Wdbc	0	0.010	0.014	0.097
Heart	0.004	0.011	0.014	0.021
Diabetes	0.003	0.016	0.011	0.018
Breast	0.002	0.004	0.007	0.029
Cardio	−0.001	0.004	0.006	0.026
Acute1	0.002	0.007	0.036	0.004
Acute2	0.005	0.008	0.003	0.018
Ave.	0.007	0.008	0.020	0.028

Table 12
The improvements of CF_1 on different noise ratios.

Data	Noise ratios			
	10%	20%	30%	40%
Improvement of CF_1 on CF				
Parkinson	0.002	0.048	0.110	*
Iris	0.006	0.012	0.053	0.047
Wdbc	0.018	0.066	0.125	0.172
Heart	0.014	0.022	0.051	0.046
Diabetes	0.010	0.016	0.021	0.022
Breast	0.005	0.019	0.046	0.078
Cardio	0.002	0.012	0.033	0.082
Acute1	0.008	0.034	0.047	0.082
Acute2	0.007	0.027	0.051	0.059
Ave.	0.008	0.028	0.060	0.070

Table 13
Comparison between our proposed methods and edited nearest neighbors.

Data	MF_{MF}	CF_1	ENN1	ENN3
Multiple-voting and edited nearest neighbor comparison				
Parkinson	0.689	0.656	0.634	0.677
Iris	0.915	0.909	0.848	0.882
Wdbc	0.947	0.909	0.853	0.899
Heart	0.775	0.762	0.745	0.745
Diabetes	0.767	0.757	0.710	0.715
Breast	0.958	0.952	0.878	0.899
Cardio	0.974	0.955	0.883	0.918
Acute1	0.914	0.917	0.839	0.870
Acute2	0.940	0.943	0.873	0.898
Ave.	0.875	0.863	0.807	0.832

5.3. Discussions

We have conducted the mathematical analysis for the proposed methods in previous section. The experimental results are consistent with the mathematical analysis. Both of them show that MF_{MF} and CF_{MF} could improve the performance of conventional *MF* and *CF*. Three new observations from experimental results include (1) CF_1 could provide significant improvement on *CF*; (2) the performances of CF_{CF} is quite poor; (3) the performances of MF_{CF} is quite poor.

To analysis above observations, we choose one dataset to analyze the reasons. The random selected dataset is Breast cancer. This dataset consists of 683 samples. Referring to the experimental setup in Section 5.2, we use 3-cross validation method. In each time of validation, the training data consists of 455 samples ($683*2/3$). We will set the noise ratio to 40%, therefore, around 182 mislabeled samples. This experiment is conducted for five times. The average error made by each type of noise detection method is shown in Table 14.

(1) Analysis why CF_1 provide significant improvement on *CF*.
As we analyzed in last section, compared to *CF*, CF_1 will reduce the number of one type of error and increase the number of the other type of error. Mathematically,

Table 14
Number of errors made by each noise detection method.

Methods	Error number	
	*E*1	*E*2
MF	38	24
MF_{CF}	5	65
CF	4	89
CF_1	12	31
CF_{CF}	1	149

$P(E1_{CF}) < P(E1_{CF_1}),\ P(E2_{CF_1}) < P(E2_{CF})$. Shown in Table 14, *CF* makes 4 type I errors and 89 type II errors. CF_1 makes 12 type I errors and 31 type II errors. Obviously the gain on type II error is much more than the lose on type I error. Therefore, the performance of CF_1 is better than *CF*.

(2) Analysis why CF_{CF} provide the poor performance. Mathematically, $P(E2_{CF}) < P(E2_{CF_{CF}}),\ P(E1_{CF_{CF}}) < P(E1_{CF})$. Shown in Table 14, *CF* makes 4 errors in type I and 89 errors in type II. CF_{CF} only makes 1 error in type I. But meanwhile, it makes 149 type II errors. Obviously the lose on type II error is much more than the gain on type I error. Therefore, the performance of CF_{CF} is worse than *CF*.

(3) Analysis why MF_{CF} provide the poor performance. Mathematically, $P(E1_{MF_{CF}}) < P(E1_{MF}),\ P(E2_{MF} < P(E2_{MF_{CF}})$. Shown in Table 14, *MF* makes 38 errors in type I and 24 errors in type II. MF_{CF} only makes 5 errors in type I. But meanwhile, it makes 65 type II errors. Obviously the lose on type II error is much more than the gain on type I error. Therefore, the performance of MF_{CF} is worse than *MF*.

6. Conclusions and future works

In bioinformatic applications, the mislabeling of training examples is a serious problem which can degrade of the performance of data analysis. The main technical contribution of this work is pointing out the limitation of traditional single-voting scheme and proposing a multiple-voting scheme to solve the problem. Single-voting consists of two steps: data partitioning and mislabel detecting. The main limitation of single-voting scheme is its unreliability due to the influence of data partitioning. To address this issue, our proposed multiple-voting scheme runs single-voting for multiple times and then combine their detection results by proposed fusion strategies, which include one vote veto, majority voting, and consensus voting.

According to the proposed multiple-voting scheme, conventional single-voting based methods including *MF* and *CF* have been extended into several new variants. Through mathematical and experimental analysis, some variants have shown promising performances which are superior to original *MF* and *CF*. These variants include MF_{MF} (combining the decisions of multiple *MF* detectors by majority voting) and CF_1 (combining the decisions of multiple *CF* detectors by one vote veto).

Other variants, including MF_1, MF_{CF}, and CF_{CF}, cannot improve the mislabeling detection performance in terms of classification accuracy. However, they are useful for the mislabeling detection applications wherein the costs of making an $E1$ and $E2$ error are different. For example, if the cost of tagging a noise-free instance as mislabeled instance ($E1$ error) is significantly higher than tagging a mislabeled instance as noise-free instance ($E2$ error), then MF_{CF} and CF_{CF} are the good candidates. Conversely if the cost of $E2$ error is significantly higher than $E1$ error, then MF_1 is the good candidate.

There are several advantages with multiple-voting scheme. It is straightforward to understand and implement. All the single-voting based methods can be easily extended to multiple-voting scheme. There are few parameters involved in these new variants. In addition, the efficiency of the proposed multiple-voting can be easily improved by parallelizing each individual detectors. Finally, it is easy to see that the proposed multiple-voting scheme is a general method. In this work, it is proposed to handle the bioinformatic mislabeling problem. But in essence, it can handle mislabeling from any domains.

Acknowledgements

This research was supported by National Natural Science Foundation of China (Grant Nos. 61100007, 61100081, 61173143) and the collaborative research project under NSFC-NRF cooperative Program (Grant No. 613111015). It was also supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No. 2011-0030823).

References

[1] Y. Saeys, I. Inza, P. Larranaga, A review of feature selection techniques in bioinformatics, Bioinformatics 23 (19) (2007) 2507–2517.
[2] R. Stevens, C. Goble, P. Baker, A. Brass, A classification of tasks in bioinformatics, Bioinformatics 17 (2) (2001) 180–188.
[3] C. Xue et al., Classification of real and pseudo microRNA precursors using local structure-sequence features and support vector machine, BMC Bioinform. 6 (2005) 310–320.
[4] B. Duval, J. Hao, Advances in metaheuristics for gene selection and classification of microarray data, Brief. Bioinform. 11 (1) (2009) 127–141.
[5] B. Wu et al., Comparison of statistical methods for classification of ovarian cancer using mass spectrometry data, Bioinformatics 19 (13) (2003) 1636–1643.
[6] M. West et al., Predicting the clinical status of human breast cancer by using gene expression profiles, in: Proc. Natl Acad. Sci. USA, 1998, pp. 11462–11467.
[7] A. Malossini, E. Blanzieri, R.T. Ng, Assessment of SVM reliability for microarray data analysis, in: 14th Conference on Machine Learning, Benelearn, Enschede, WP05-03, 2005.
[8] J. Bootkrajang, A. Kaban, Classification of mislabelled microarrays using robust sparse logistic regression, Bioinformatics 29 (7) (2013) 870–877.
[9] J. Saez, M. Galar, J. Luengo, F. Herrera, A first study on decomposition strategies with data with class noise using decision trees, in: Hybrid Artificial Intelligent Systems, Lecture Notes in Computer Science, vol. 7209, 2012, pp. 25–35.
[10] D.L. Wilson, Asymptotic properties of nearest neighbor rules using edited data, IEEE Trans. Syst. Man Cybernet. 2 (3) (1992) 431–433.
[11] J. Young, J. Ashburner, S. Ourselin, Wrapper methods to correct mislabeled training data, in: 3rd International Workshop on Pattern Recognition in Neuroimaging, 2013, pp. 170–173.
[12] D. Guan, W. Yuan, et al., Identifying mislabeled training data with the aid of unlabeled data, Appl. Intell. 35 (3) (2011) 345–358.
[13] C.E. Brodley, M.A. Friedl, Improving automated land cover mapping by identifying and eliminating mislabeled observations from training data, in: Geoscience and Remote Sensing Symposium, 1996, pp. 1379–1381.
[14] C.E. Brodley, M.A. Friedl, Identifying mislabeled training data, J. Artif. Intell. Res. 11 (1999) 131–167.
[15] A.I. Marques et al., Analysis of new techniques to obtain quality training sets, Pattern Recogn. Lett. 24 (2003) 1015–1022.
[16] Z.H. Zhou, Y. Jiang, Editing training data for knn classifiers with neural network ensemble, in: Lecture Notes in Computer Science, vol. 3173, 2004, pp. 356–361.
[17] G.H. John, Robust decision trees: removing outliers from databases, in: Proceeding of International Conference on Knowledge Discovery and Data Mining, 1995, pp. 174–179.
[18] X. Wu, X. Zhu, Q. Chen, Eliminating class noise in large datasets, in: Proceeding of International Conference on Machine Learning, 2003, pp. 920–927.
[19] A.I. Marques et al., Decontamination of training data for supervised pattern recognition, in: Advances in Pattern Recognition Lecture Notes in Computer Science, vol. 1876, 2000, 621–630.
[20] B.B. Chaudhuri, A new definition of neighborhood of a point in multi-dimensional space, Pattern Recogn. Lett. 17 (1996) 11–17.
[21] S. Verbaeten, A.V. Assche, Ensemble methods for noise elimination in classification problems, in: Proceeding of 4th International Workshop on Multiple Classifier Systems, 2003, pp. 317–325.
[22] D. Metxas et al., Distinguishing mislabeled data from correctly labeled data in classifier design, in: 16th IEEE International Conference on Tools with Artificial Intelligence, 2004, pp. 668–672.