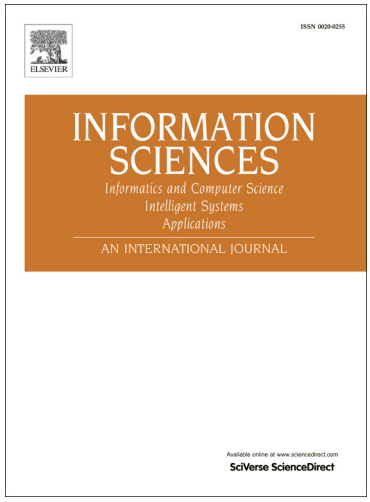
Robust twin boosting for feature selection from high-dimensional omics data with label noise

He, Shan; Chen, Huanhuan; Zhu, Zexuan; Ward, Douglas G.; Cooper, Helen J.; Viant, Mark R.; Heath, John K.; Yao, Xin

DOI:
10.1016/j.ins.2014.08.048



Robust twin boosting for feature selection from high-dimensional omics data with label noise

Shan He, Huanhuan Chen, Zexuan Zhu, Douglas G. Ward, Helen J. Cooper, Mark R. Viant, John K. Heath, Xin Yao

PII: S0020-0255(14)00853-6
DOI: http://dx.doi.org/10.1016/j.ins.2014.08.048
Reference: INS 11078

To appear in: *Information Sciences*

Received Date: 18 June 2013
Revised Date: 2 August 2014
Accepted Date: 22 August 2014

Please cite this article as: S. He, H. Chen, Z. Zhu, D.G. Ward, H.J. Cooper, M.R. Viant, J.K. Heath, X. Yao, Robust twin boosting for feature selection from high-dimensional omics data with label noise, *Information Sciences* (2014), doi: http://dx.doi.org/10.1016/j.ins.2014.08.048

Robust Twin Boosting for Feature Selection from High-dimensional Omics Data with Label Noise

Shan He[a], Huanhuan Chen[a], Zexuan Zhu[b,*], Douglas G. Ward[c], Helen J. Cooper[d], Mark R. Viant[d], John K. Heath[d], Xin Yao[a]

[a] *CERCIA, School of Computer Science, University of Birmingham, UK*
[b] *College of Computer Science and Software Engineering, Shenzhen University, China*
[c] *School of Cancer Sciences, University of Birmingham, UK*
[d] *School of Biosciences, University of Birmingham, UK*

Abstract

Omics data such as microarray transcriptomic and mass spectrometry proteomic data are typically characterized by high dimensionality and relatively small sample sizes. In order to discover biomarkers for diagnosis and prognosis from omics data, feature selection has become an indispensable step to find a parsimonious set of informative features. However, many previous studies report considerable label noise in omics data, which will lead to unreliable inferences to select uninformative features. Yet, to the best of our knowledge, very few feature selection methods are proposed to address this problem. This paper proposes a novel ensemble feature selection algorithm, robust twin boosting feature selection (RTBFS), which is robust to label noise in omics data. The algorithm has been validated on an omics feature selection test bed and seven real-world heterogeneous omics datasets, of which some are known to have label noise. Compared with several state-of-the-art ensemble feature selection methods, RTBFS can select more informative features despite label noise and obtain better classification results. RTBFS is a general feature selection method and can be applied to other data with label noise. MATLAB implementation of RTBFS and sample datasets are available at: http://www.cs.bham.ac.uk/~szh/TReBFSMatlab.zip

Keywords: Feature Selection, Boosting, Ensemble Learning.

1. Background

Omics technologies such as genomics, proteomics and metabolomics have become powerful tools for analyzing biological systems. In medical science, omics technologies have been applied to discover molecular signatures as biomarkers for disease diagnosis, prognosis and staging. Omics data are usually acquired on small number of patient samples, typically tens to a few hundred in each disease group, but each sample data often contains tens of thousands of variables or features.

*Corresponding author. E-mail addresses: s.he@cs.bham.ac.uk (S. He), h.chen@cs.bham.ac.uk (H. Chen), zhuzx@szu.edu.cn (Z. Zhu), d.g.ward@bham.ac.uk (D. G. Ward), h.j.cooper@bham.ac.uk (H. J. Cooper), m.viant@bham.ac.uk (M. R. Viant), j.k.heath@bham.ac.uk (J. K. Heath), x.yao@cs.bham.ac.uk (X. Yao)

Feature selection plays a crucial role in omics data analysis. Feature selection methods can select a subset, usually a parsimonious subset of important features for building fast and robust learning models with better generalization capability, therefore improve the learning accuracy and interpretability of the results [29, 72, 64]. Current approaches to feature selection include: filter methods [68], wrapper methods [15, 56], embedded methods [29], and hybrid methods [74, 73, 72, 12, 25]. Filter methods evaluate the goodness of features based on the intrinsic characteristic of the data and without any consideration of the learning models. They are computationally fast, but they could select features not suitable for the learning models. Contrarily, wrapper methods involve the learning models and evaluate features directly according to the learning performance. They tend to obtain better learning performance at the price of computation cost. Embedded methods, which could be seen as more efficient wrapper methods, perform feature selection while building the learning model, i.e., feature selection and learning model are optimised simultaneously. Hybrid methods try to take advantage of both filter and wrapper methods in some specific hybridization frameworks. The focus of this study is on the development of a generic embedded feature selection method by performing classification and feature selection simultaneously, for various omics data based on boosting algorithms.

When feature selection is applied to omics data, one problem is the fact that there is considerable noise in the omics data. Firstly, there is data noise arising from biological variation and analytical variance. More crucially, many studies reported that label noise [31] is also common in omics data sets [45, 70, 7]. Label noise might be introduced by false diagnosis, which usually occurs near the decision boundary of the feature space. Such noise is called mislabeling. It is also very likely during decision making, experts may introduce mislabeling because they are distracted. Another reason is that in some heterogeneous disease such as cancer, subgroups may behave differently - a subgroup might only be one or a few individuals in these small studies and would appear to be outliers, which would introduce significant label noise. Label noise will lead to unreliable inferences from the omics data and consequently result in the selection of unreliable features. Such a problem presents significant challenges to feature selection algorithms to extract reliable combination of biomarkers from omics data with high discriminant power.

The label noise problem has been recognized by machine learning research community. In the past few years, several studies on label noise problem have been proposed [22, 45, 70, 39, 7, 71, 27, 55, 21, 67, 37]. However, most of the studies focus on designing classifiers that are robust to label noise. The only exception is [21] which proposes a novel feature selection method to select reliable features from data with label noise. The method can robustly evaluate the mutual information [63] between features and labels using a probabilistic label noise model and a nearest neighbours-based entropy estimator, then a backward greedy search procedure is applied to search for relevant sets of features based on the mutual information. The method is essentially a filter based method, therefore it ignores the interaction with the classifier, a general drawback of filter based methods as mentioned above, which may lead to worse classification performance compared with wrapper and embedded feature selection methods. Apart from this drawback, methods based on mutual information such as in [21] "does not always guarantee to decrease the misclassification probability" [20], which may also lead to worse classification performance.

To address the challenge of selecting reliable features from data with label noise, a novel embedded feature selec-

tion algorithm called robust twin boosting feature selection (RTBFS) is proposed in this study. The RTBFS algorithm is based on the twin boosting framework, which is a novel ensemble feature selection algorithmic framework proposed in [8]. When applied to several omics datasets with known label noise, the standard twin boosting algorithm in [8] could not select features of satisfactory performance (discussed in Section 2). Therefore, in this study, the twin boosting framework is improved for better robustness to label noise by incorporating a novel robust loss function, i.e., robust eta-loss [36] and a robust weak learner, i.e., robust componentwise linear least squares.

In the experimental study, the performance of RTBFS is firstly thoroughly investigated using a feature selection test bed based on a real-world microarray dataset of which the optimal features are known. In order to evaluate the robustness of RTBFS against label noise, RTBFS is tested on the contaminated test bed to which different percentage of label noise is deliberately introduced. And then the RTBFS algorithm is applied to seven publicly available real-world omics datasets including three microarray transcriptomic data, four Mass Spectrometry Time-Of-Flight (MS-TOF) proteomic data and one Ion Molecule Reaction MS (IMR-MS) metabolomic data. Among these datasets, the three microarray datasets have been reported to have label noise [45, 70, 39, 7]. It is also worth mentioning that, while the label noise problem has received attention in genomic research community, it was until recently researchers in other omics, e.g., proteomics research communities started to discuss this issue using simulated data [50].

For comparison purpose, several twin boosting variants and several state-of-the-art feature selection algorithms, e.g., a fast correlation based feature selection algorithm [69] and an ensemble embedded feature selection algorithm [1] based on the popular support vector machine recursive feature elimination (SVM-RFE) [30] are implemented. Apart from the algorithms mentioned above, the results of RTBFS are also compared with the results from several novel feature selection algorithms published in literature recently. The results show that RTBFS can robustly extract parsimonious informative features from noisy omics data that generate better classification models, which is highly desirable for clinical omics data analysis where accuracy and interpretability are important.

2. Boosting Based Feature Selection and Label Noise Problem

Boosting is a type of ensemble machine learning method, which constructs a set of classifiers to classify new data points in some way (typically by weighted or unweighted voting of their predictions) [16]. Since the publication of Adaboost [23], boosting algorithms have attracted much attention in the machine learning and statistics communities due to their simplicity and competitive prediction accuracy. Boosting is also a good feature selection scheme especially for high-dimensional data where the number of features is much larger than the sample size. However, until recently the feature selection function of boosting algorithm has not been fully exploited. For example, in [14] and [26], only univariate methods such as t-test and Wilcoxon test, were used for selecting informative features prior to their boosting classifiers.

One of the pioneering exploitations of boosting algorithm framework for feature selection is the twin boosting framework [8]. This framework consists of two stages of boosting to select the best subset of features. Using several

benchmark machine learning problems, Bülmann and Hothorn [8] showed that the twin boosting has better feature selection behavior than standard boosting on high-dimensional data. In particular, it is able to select a parsimonious set of important features with fewer falsely selected features.

However, when applied to data with label noise, such as omics datasets, the standard twin boosting yield less satisfactory results. One of the reasons lies in the logistic loss function used in the two boosting steps. In [43], Long and Servedio analyzed several popular loss functions including the logistic loss function, and from their theoretical point view, all these convex loss functions are sensitive to the label noise. Another important reason is the fact that the weak learner used in the twin boosting is standard decision tree, which is also well-known to be susceptible to label noise [3].

3. Robust Twin Boosting Feature Selection (RTBFS)

The aim of this study is to improve robustness of the twin boosting against label noise by introducing three methods from robust statistics and regularization. The following sections first give an introduction to the generic boosting algorithm framework to make this paper self-contained, and then introduce a detailed introduction to the standard twin boosting and then introduce the three new methods in details.

3.1. Standard Boosting Algorithm and Functional Gradient Descent

Boosting algorithm is a kind of supervised learning algorithm which aims to build prediction models based on training data set. Assuming the data set is given by $(\mathbf{x}_i, y_i)_{i=1}^N$ of N samples, where $\mathbf{x}_i \in \mathcal{X}$ is a p-dimensional predictor variable. Denote $\mathbf{y} = y_i,\ i = 1, \cdots, N$, and $y_i \in \mathcal{Y} = \{1, \cdots, k\}$. For binary classification, without loss of generality, let $y_i \in \{+1, -1\}$. A general supervised learning algorithm aims to estimate the unknown function $f^*(\cdot)$ by minimizing the expected value of some specified convex loss function ρ iteratively over a set of learnt functions $f(\cdot) : \mathbb{R}^p \to \mathcal{Y}$:

$$f^*(\cdot) = \arg\min_{f(\cdot)} \mathbb{E}_{\mathbf{x},\mathbf{y}}[\rho(\mathbf{y}, f(\mathbf{x}))], \tag{1}$$

where $\rho(\mathbf{y}, f) : \mathcal{Y} \times \mathcal{X} \to \mathbb{R}$.

In order to minimise the above empirical risk, the boosting algorithm essentially seeks an approximation of $f(\mathbf{x})$ using an additive form of expansion, e.g., in the form of a weighted sum of functions called weak (or base) learners [24]:

$$f(\mathbf{x}; \{\beta_m, \mathbf{a}_m\}_1^M) = \sum_{m=1}^{M} \beta_m \hat{g}^{[m]}(\mathbf{x}; \mathbf{a}_m), \tag{2}$$

where β_m is called boosting weight which will be discussed in detailed in the following section and $\hat{g}(\mathbf{x}; \mathbf{a})$ is a simple parametrised function called weak learner that takes $\mathbf{x}$ as input and outputs a real value, characterised by parameters $\mathbf{a} = \{a_1, a_2, \cdots\}$. It is worth mentioning that, the approximation of some function $f(\mathbf{x})$ using expansion can be found in other function approximation methods such as artificial neural networks [52] and wavelet [17].

According to the empirical risk minimisation principle, the empirical risk, e.g., the average value of the loss function on the training set should be minimised as follows:

$$f^*(\cdot) = \arg\min_{f(\cdot)} \frac{1}{n} \sum_{i=1}^{N} \rho(y_i, f(\mathbf{x}_i)). \tag{3}$$

Denoted $\mathbf{P} = \{\beta_m, \mathbf{a}_m\}_1^M$, and plug equation (2) into equation (3), the boosting training algorithm essentially aims to minimise the empirical risk in equation (3) by solving a parameter optimisation problem:

$$\mathbf{P}^* = \arg\min_{\mathbf{P}} \sum_{i=1}^{N} \rho(y_i, f(\mathbf{x}_i, ; \mathbf{P})) \tag{4}$$

and then:

$$f^*(\cdot) = f(\mathbf{x}; \mathbf{P}^*). \tag{5}$$

To solve equation (4), Friedman proposed the functional gradient descent (FGD), a more general statistical framework for boosting algorithms, in [24]. FGD essentially seeks the steepest-descent in function space, which expresses the solution for the parameters in the form:

$$\mathbf{P}^* = \sum_{m=0}^{M} \mathbf{p}_m \tag{6}$$

where $\mathbf{p}_0$ is an initial guess and $\{\mathbf{p}_m\}_1^M$ are successive increments based on the sequence of preceding steps:

$$\mathbf{p}_m \equiv (\beta_m, \mathbf{a}_m) = \arg\min_{\beta, \mathbf{a}} \sum_{i=1}^{N} \rho(y_i, f^{[m-1]}(\mathbf{x}_i) + \beta \hat{g}^{[m]}(\mathbf{x}_i; \mathbf{a})) \tag{7}$$

$$f^{[m]}(\mathbf{x}) = f^{[m-1]}(\mathbf{x}) + \beta_m \hat{g}^{[m]}(\mathbf{x}; \mathbf{a}_m). \tag{8}$$

However, for many loss functions $\rho(y, f)$ and weak learners $\hat{g}$ equation (4) is a hard optimisation problem. Since equations (7) and (8) can be seen as the steepest descent step towards the data based estimation of f^* in function space, under the constraint that the step direction $\hat{g}(\mathbf{x}; \mathbf{a}_m)$ is a member of the $\hat{g}(\mathbf{x}; \mathbf{a})$, one can obtain the best steepest descent step direction defined only at the data points $\{\mathbf{x}_i\}_1^N$ in the N-dimensional data space at $f^{[m]}(\mathbf{x})$:

$$-u_i^{[m]} = -\left[\frac{\partial \rho(y_i, f(\mathbf{x}_i))}{\partial f(\mathbf{x}_i)}\right]_{f(\mathbf{x}) = f^{[m-1]}(\mathbf{x})} \quad \text{for } i = 1, \ldots, N. \tag{9}$$

In order to generalise the gradient in equation (9) to other $\mathbf{x}$ values, the weak learner $\hat{g}^{[m]}(\mathbf{x}, \mathbf{a})$ can be fitted with $\{(x_i, -u_i^{[m]})\}_{i=1}^N$, which is most highly correlated with $\{-u_i^{[m]}\}_1^N$ over the data distribution.

The following one-dimensional optimisation problem is then solved to obtain the boosting weight β_m,:

$$\beta_m = \arg\min_{\beta} \sum_{i=1}^{N} \rho\left(y_i, f^{[m-1]}(\mathbf{x}_i) + \beta \hat{g}^{[m]}(\mathbf{x}_i)\right). \tag{10}$$

The generalization ability of the model can be degraded if it is fitted to the training set too closely. Besides controlling the number of gradient boosting iterations, the robust twin boosting can be regularized by the shrinkage technique [10, 24] which consists of modifying the update rule as follows:

$$\hat{f}^{[m]} = \hat{f}^{[m-1]} + \nu\beta_m \hat{g}^{[m]}(\cdot), \tag{11}$$

where parameter $\nu \in (0, 1]$ is called the learning rate. A small ν yields dramatic improvement in the model's generalization ability over twin boosting without shrinkage, with the price of increasing computational time during training. The reason for the improvement is "less clear" as Friedman himself put it in [24]. Recently, [60] shows that margin maximisation may be achieved by using shrinkage to scale back the step size. However, the theoretical guidance for selecting the optimal ν is lacking. Moreover, the optimal number of iterations, m_{stop}, and the learning rate, ν, also interdependent which makes using cross-validation to find the optimal learning rate difficult. Therefore, the learning rate ν is set to 0.3 in this study by empirical experiments.

3.2. Twin Boosting with Feature Selection

The twin boosting [8] extends the generic boosting framework by executing two stages of boosting, of which the first stage is the standard boosting as mentioned above, and the second stage is an improved boosting that enforces to resemble the first boosting round as detailed below. The motivation of adding the second round of boosting is to get sparser models to select most informative variables from the variables selected from the first round. The key idea is to modify the functional gradient descent step in the second round of boosting so that the weak learners that contribute more to the first round of boosting will be more likely selected. The details of the algorithm are explained below.

After running the first stage of standard boosting algorithm of m_1 iterations, one can obtain the fitted function at the data points $\hat{\mathbf{f}}_{init}^{[m_1]} = (\hat{f}_{init}^{[m_1]}(x_1), \cdots, \hat{f}_{init}^{[m_1]}(x_N))$ and the subset of indices corresponding to selected features $\hat{\mathcal{V}}^{[m_1]}$. Then in order to select more relevant features, at the second stage of twin boosting, for every randomly generated feature subset $\mathcal{W} \subseteq \hat{\mathcal{V}}^{[m_1]}$, the boosting algorithm tries to fit $u_1, \cdots, u_N$ to $x_1^{\mathcal{W}}, \cdots, x_N^{\mathcal{W}}$ by WeakLearner, where $x^{\mathcal{W}}$ denotes $\{x^{(j)};\ j \in \mathcal{W}\}$. Denote the fitted function by $\hat{h}_{\mathcal{W}}(\cdot)$, or $\hat{\mathbf{h}}_{\mathcal{W}} = (\hat{h}_{\mathcal{W}}(x_1), \cdots, \hat{h}_{\mathcal{W}}(x_N))$, then the twin boosting chooses the best feature subset $\mathcal{W}$ by solving the following optimisation problem [8]:

$$\begin{aligned}
&\hat{\mathcal{W}} = \arg\max_{\mathcal{W}} C_{\mathcal{W}}^2 (2\langle \mathbf{u}, \hat{\mathbf{h}}_{\mathcal{W}} \rangle - \|\hat{\mathbf{h}}_{\mathcal{W}}\|^2), \\
&C_{\mathcal{W}} = \langle \hat{\mathbf{f}}_{init}^{[m_1]} - \overline{\hat{f}_{init}^{[m_1]}}, \hat{\mathbf{h}}_{\mathcal{W}} \rangle / \|\hat{\mathbf{h}}_{\mathcal{W}}\|, \\
&\overline{\hat{f}_{init}^{[m_1]}} = \frac{1}{N} \sum_{i=1}^{N} \hat{f}_{init}^{[m_1]}(x_i).
\end{aligned} \tag{12}$$

3.3. Robust Twin Boosting

Twin boosting is able to select effective feature subsets and train learning machines, simultaneously. However, most of the data in bioinformatics research are noisy and of small sample size. The original implementation of twin

boosting uses logistic function as its cost function, which is not robust to data with label noise [43].

In order to address these problems, this study extends the twin boosting by proposing robust twin boosting to label noise. The pseudo code of the improved robust twin boosting algorithm is presented in Algorithm 1. In summary, the algorithm as described in Algorithm 1 robustifies the original twin boosting by robust eta-loss function and componentwise linear least squares learner (as detailed in Section 3.3.1 and Section 3.3.2, respectively).

Algorithm 1 Robust Twin Boosting Framework

begin { First stage of boosting }
initialize $\hat{f}^{[0]} \equiv \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$. Set $m = 0$.
for $m = 1, \ldots, m_{\text{stop}}$ **do**
    Compute u_i and evaluate at $\hat{f}_{m-1}(x_i)$ using equation (14).
    Fit $u_1, \ldots, u_n$ by WeakLearner to generate an approximation of the negative gradient vector $\hat{g}^{[m]}(\cdot)$ using equation (15).
    Perform line search to find optimal β_m according to equation (10).
    Update $\hat{f}^{[m]}$ using equation (11).
end for
output the final function estimate $\hat{f}^{[m_{\text{stop}}]}$.
end
begin { Second stage of boosting }
initialize $\hat{f}^{[0]} \equiv \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$. Set $m = 0$.
for $m = 1, \ldots, m_{\text{stop}}$ **do**
    Compute u_i and evaluate at $\hat{f}_{m-1}(x_i)$ using equation (14).
    For every randomly generated feature subset $\mathcal{W} \subseteq \hat{\mathcal{V}}^{[m_1]}$, fit $u_1, \ldots, u_n$ to $x_1^{\mathcal{W}}, \ldots, x_n^{\mathcal{W}}$ by equation (15).
    Choose the best feature subset $\mathcal{W}$ according to equation (12).
    Let $\hat{g}^{[m]}(\cdot) = \hat{h}_{\hat{\mathcal{W}}}(\cdot)$.
    Perform line search to find optimal β_m according to equation (10).
    Update $\hat{f}^{[m]}$ using equation (11).
end for
output the final function estimate $\hat{f}^{[m_{\text{stop}}]}$; the subset of most effective features $\mathbf{x}^{\hat{\mathcal{W}}}$ and their feature importance values, which is the frequencies of the selected features from all m_{stop} iterations.
end

3.3.1. Robust Eta-Loss Function

In [36], Kanamori *et al.* introduced a loss function called robust eta-loss. The loss function is based on contamination models, which are the statistical models describing mislabels [11]. In such models, the distribution of

contaminated data is modeled as a mixture of the distributions of the non-contaminated part and the contaminated part. The eta-boost loss function uses the contamination models to describe the change in sign of class labels near decision boundaries and consequently to prevent overweighting of mislabeling samples. Kanamori *et al.* also introduced robust estimator in robust statistics to limit the influence of outliers. For details of the robust eta-loss function please refer to [36]. Here the robust eta-loss function as defined in equation (13) is adopted.

$$\rho(y, f) = \begin{cases} -yf & \text{if } yf \leq 0, \\ \frac{1}{\eta^2}\left[(1-\eta)(\exp(-2yf)-1)\eta + (2\eta-1)\log(1+(\exp(-2yf)-1)\eta)\right] & \text{otherwise,} \end{cases} \tag{13}$$

The parameter η is the contamination ratio, which is the main tuning parameter of the robust eta-loss function. However, our experimental results suggest that the performance of the robust eta-loss function is not significantly degraded by the inappropriate value of η (see Section 7.1). The negative gradient for the robust eta-loss function is defined in equation (14) and plotted in Fig. 2.

$$u_i = \begin{cases} y_i & \text{if } y_i \hat{f}_{m-1} \leq 0, \\ y_i \frac{(1-\eta)\exp(-2y_i \hat{f}_{m-1})+\eta}{(1-\eta)\exp(2y_i \hat{f}_{m-1})+\eta} & \text{otherwise,} \end{cases} \tag{14}$$

From the figure, it can be seen that compared with other loss functions, the robust eta-loss function negative gradient for the robust eta-loss is less aggressive in terms of penalise mis-classification, e.g., the penalty increases linearly on the right hand of the figure. Such behaviour is helpful for improving the robustness to label noise since it limits the influence mis-classification caused by label noise.

3.3.2. Robust Componentwise Linear Least Squares Learner

In the original twin boosting, the weak learner $\hat{g}(\cdot)$ is not robust to label noise. In this study, the robust componentwise linear least squares (RCLLS) learner is employed. RCLLS uses an iteratively reweighted least squares algorithm to solve weighted linear least squares (WLLS), which is defined as follows:

$$\begin{aligned} \hat{g}(x) &= \hat{\gamma}_{\hat{\mathcal{V}}} \mathbf{x}^{(\hat{\mathcal{V}})}, \\ \hat{\gamma}_j &= \langle \mathbf{u}, \mathbf{x}^{(j)} \rangle / \|\mathbf{x}^{(j)}\|^2, \\ \hat{\mathcal{V}} &= \arg\min_{1 \leq j \leq p} \sum_{i=1}^{N} w_i (u_i - \hat{\gamma}_j x_i^{(j)})^2, \end{aligned} \tag{15}$$

where w_i is the weight of a data point and $\mathbf{x}^{(\hat{\mathcal{V}})}$ denotes $\{x^{(j)};\ j \in \hat{\mathcal{V}}\}$.

The weights at each iteration are calculated by applying a weighting function w to the residuals from the previous iteration, which gives lower weights to points that do not fit well. Several runs of experiments are executed to determine the weight vector $\mathbf{w}$ which generates the best results. In our implementation, the logistic weight vector [34] is chosen:

$$\mathbf{w} = \tanh(\mathbf{r})/\mathbf{r}.$$

The value $\mathbf{r}$ is given by [34] as follows:

$$\mathbf{r} = \mathbf{u}^{[m-1]}/(1.205s \cdot \sqrt{1-\mathbf{h}}),$$

where $\mathbf{h}$ is the vector of leverage values $\mathbf{h} = \mathbf{x}(\mathbf{x}\mathbf{x}^T)^{-1}\mathbf{x}^T$. The constant s is an estimate of the standard deviation of the error term:

$$s = \text{MAD}/0.6745,$$

where MAD is the median absolute deviation of the residuals from their median. The constant 0.6745 is used to make the estimate unbiased for the normal distribution [34]. The weighted LS-SVM [59, 66, 35, 42] could be another good choice of robust weak learner, yet this study focuses on RCLLS for the sake of simplicity.

4. Other Twin Boosting Variations

As discussed in Section 2, the performance of the the twin boosting framework on noisy data depends on the loss function and the weak learner. In order to evaluate the performance of the proposed RTBFS algorithm, several twin boosting variations, with different loss functions and different weak learners are implemented for comparison.

In the original paper [8], Bülmann and Hothorn suggested to use the logistic loss function: $\rho(y, f) = \log_2(1 + \exp(-2yf))$. For classification, which is our case, decision tree is recommended as the weak learner. This algorithm is termed as standard twin logistic boosting (STLB). All other settings, e.g., the number of iterations of STLB are the same as those of RTBFS. STLB serves as the baseline algorithm for comparison throughout the experiments.

Twin boosting with two other robust loss functions proposed in recent years are also tested in comparison with RTBFS. The first robust loss function tested is the Savage loss function [46]:

$$\rho(y, f) = \frac{1}{(1 + e^{2yf})^2}.$$

The negative gradient or the derivation of the Savage loss function function can be easily obtained:

$$\frac{\partial \rho(y, f)}{\partial f} = \frac{4ye^{2yf}}{(e^{2yf} + 1)^3}.$$

The second robust loss function is the Huberised logistic loss function [31] which is defined as follows:

$$H_c(\rho) = \begin{cases} \frac{1}{2}\rho^2 & \text{if } |\rho| \le c, \\ c(|\rho| - \frac{1}{2}c) & \text{otherwise,} \end{cases},$$

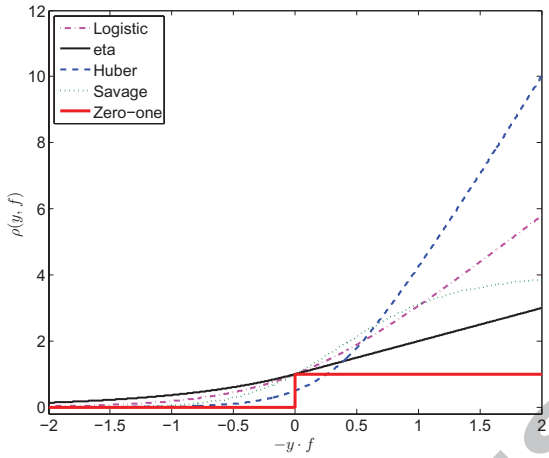


Figure 1: Logistic loss, robust eta-boost loss, Huberized logistic loss and Savage loss functions.

where ρ is the loss function, in this case, it is the logistic loss function. Parameter c is dependent on the scale of ρ which is defined as

$$c = p\text{-quantile}|\rho|.$$

where $p = 0.95$. For simplicity, this function is called Huber loss function.

The derivation of the Huber loss function yields:

$$\frac{\partial H_c(\rho)}{\partial \rho} = \begin{cases} \rho & \text{if } |\rho| \le c, \\ c \cdot \text{sign}(\rho) & \text{otherwise,} \end{cases}.$$

The three robust loss functions in comparison with logistic loss and 0-1 loss function is plotted in Fig. 1. Their derivatives are shown in Fig. 2.

The logistic loss function does not deal with label noise explicitly,while the later two alternative robust loss functions try to decrease the sensitivity to label noise. However, their strategies for dealing with label noise are different. The Huber loss function essentially use robust estimation to reduce the effect or outliers. This Huber loss function might have a slower convergence in case of no noise. The Savage loss function was designed from the perspective of probability elicitation in statistics with the aim to resist label noise but to converge fast when there is no noise.

In addition to RCLLS, decision tree and decision stump as weak learners as used in the original twin boosting paper are also implemented in this study.

5. Datasets

The experimental study of the proposed algorithm is performed on an omics test bed and various real-world omics data.

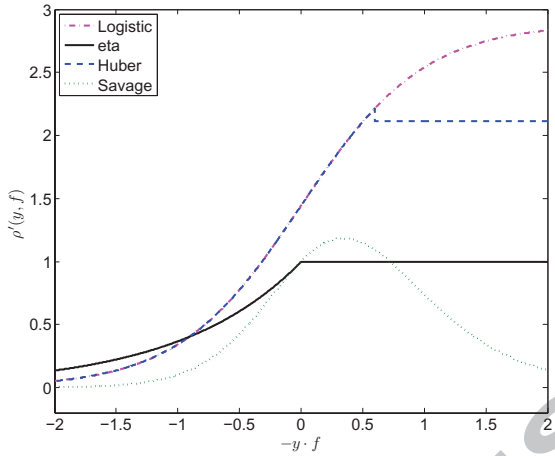


Figure 2: Derivatives of logistic loss, robust eta-boost loss, Huberized logistic loss and Savage loss functions.

5.1. Omics Feature Selection Test Bed

In [9], Choudhary *et al.* proposed an omics test bed[1] for the evaluation of feature selection algorithms. Based on a real-world microarray dataset of breast cancer, Choudhary *et al.* extracted an optimal feature set of the top s features ($s \leq 7$) that achieves the best classification performance with a particular classifier using exhaustive search techniques from a previously established 70-gene prognosis profile. To estimate the error rate of feature selection algorithms on the dataset $\mathcal{S}$, the test bed algorithm employs bootstrap which draws t samples of size b and applies feature selection algorithm $\mathcal{A}$ to each sample. The given classification algorithm $\mathcal{R}$ is used to classify the t feature sets selected by $\mathcal{A}$ to obtain the error. The following measures are used to evaluate the feature selection algorithm:

- The average error increase, δ, which is calculated from:

$$\delta(\mathcal{R}, \mathcal{A}; b, s) = \frac{1}{t} \sum_{j=1}^{t} e_j(\mathcal{R}, \mathcal{A}; b, s) - \epsilon_{opt}(\mathcal{R}, \mathcal{A}; b, s),$$

  where $e_j(\mathcal{R}, \mathcal{A}; b, s)$ and $\epsilon_{opt}(\mathcal{R}, \mathcal{A}; b, s)$ are the errors for the optimal feature set of size s found by the exhaustive search and the feature set of size s selected by algorithm $\mathcal{A}$, respectively.

- The average proportional increase in error, υ, which is calculated from:

$$\upsilon(\mathcal{R}, \mathcal{A}; b, s) = \frac{1}{t} \sum_{j=1}^{t} \frac{e_j(\mathcal{R}, \mathcal{A}; b, s)}{\epsilon_{opt}(\mathcal{R}, \mathcal{A}; b, s)}$$

- The average number, $\tau(\mathcal{R}, \mathcal{A}; b, s)$, of features in the feature set found by the algorithm also in the optimal feature set.

[1]The test bed can be downloaded at http://public.tgen.org/tgen-cb/support/testbed/

Table 1: Details of the seven publicly available datasets used in this study. n is the number of samples and p is the number of variables.

Dataset	Publication	Type	n	p
Leukemia-m	[28]	Microarray transcriptomic	72	7129
Prostate	[57]	Microarray transcriptomic	102	12600
Colorectal-m	[4]	Microarray transcriptomic	62	2000
Pancreatic	[33]	SELDI-TOF MS proteomic	191	6771
Leukemia-p	[54]	SELDI-TOF MS proteomic	417	119
Liver-p	[51]	MALDI-TOF MS proteomic	129	13600
Liver-m	[47]	IMR-MS metabolomic	126	114

This test bed provides an easy-to-use platform to evaluate feature selection algorithms under uniform conditions and compared with the optimal feature set.

5.2. Real-World Omics Datasets

The efficiency of the proposed algorithm on real-world omics data is assessed with three microarray data, four MALDI-TOF MS proteomic data and one IMR-MS metabolomic dataset. The class labels of these datasets are binary, e.g., disease and normal. The details of these seven datasets are listed in Table 1. All the three microarray datasets are previously reported to have label noise [45, 70, 39, 7]. There is no studies to report label noise problem in other omics datasets yet, but they are also very likely suffered from label noise. For example, the premalignant pancreatic cancer proteomic dataset is prone to have considerable label noise because of the complexity of premalignant cancer [33].

Among these datasets, the three real-world microarray datasets, Pancreatic dataset and Liver-p dataset are raw omics data. They are preprocessed using the techniques detailed in Section 5.2.1 and Section 5.2.2. The Liver-m dataset is selected from one case of five classification tests [47]: NAFLD + AFLD + cirrhosis versus healthy.

5.2.1. Data Preprocessing for Raw TOF MS Data

For the raw TOF mass spectrometry data, the mass/charge (*m/z*) vector is firstly resampled using a resampling algorithm (*msresample.m*) of MATLAB Bioinformatics Toolbox, which resamples a signal with peaks to equally or linearly spaced points, in order to compare different spectra under the same reference and at the same resolution. The second step is to adjust the baseline caused by the chemical noise in the matrix or by ion overloading. The background correction procedure is performed with *msbackadj.m* in the MATLAB Bioinformatics Toolbox using default parameters. Particularly, the background is first estimated within multiple shifted windows, and then the varying baseline is regressed to the window points using a spline approximation following by the adjustment of the background of the input signal. Gaussian kernel smoothing is also used to reduce the background noise caused by factors such as instrument measurement error. Finally, each spectrum is normalized by standardizing the area under the curve (AUC) to the median of the whole set of spectrum using *msnorm.m*. For peak detection, our in-house peak extraction method, SNEO (Smooth Nonlinear Energy Operator) peak extraction method [32] is used.

5.2.2. Data Preprocessing for Raw Microarray Data

The data preprocessing step used for raw microarray data in this study is similar to the one used in [13], which log-transforms the gene expression data and then standardizes them to zero mean and unit variance.

6. Experimental Setup

This section presents the motivations of the experimental design, the experimental setup on both the test bed and real-world omics data, the other feature selection methods considered for comparison study, and the classification algorithms used.

6.1. Questions to be Answered

The feature selection test bed and the seven real-world omics datasets are used to perform the following five experiments to answer five questions:

1. Is RTBFS algorithm sensitive to control parameters?
2. How does RTBFS compare with other feature selection algorithms including those twin boosting variants described in Section 4 in terms of robustness against label noise?
3. How does RTBFS algorithm perform on the data without label noise? In other word, whether will our robustification technique deteriorate the feature selection performance on clean omics data?
4. How does RTBFS algorithm perform on the real-world omics data, which is supposed to have different degrees of label noise, in comparison with other feature selection algorithms?
5. Are those features selected by RTBFS algorithm biologically plausible?

The omics feature selection test bed is used as benchmark to answer questions 1 to 3 and the seven real-world omics datasets to answer questions 4 and 5.

6.2. Experimental Setup on Feature Selection Test Bed

To evaluate the features selected by the feature selection algorithms, the feature selection test bed provides four embedded standard classifiers, namely LDA linear discriminant analysis (LDA), 3-nearest-neighbor (3NN), 5-nearest-neighbor (5NN) and classification-and-regression tree (CART). The test bed is used to perform experiments described as follows:

Experiment 1: Sensitive analysis of RTBFS's control parameters.

Experiment 2: Comparison of RTBFS with other feature selection algorithms on robustness against label noise by testing these algorithms on the test bed contaminated with different percentage of label noise.

Experiment 3: Comparison of RTBFS with other feature selection algorithms on the original clean test bed.

6.3. Experimental Setup on Real-World Omics Datasets

RTBFS is applied to the seven real-world omics datasets to select features in comparison with standard twin boosting and other state-of-the-art feature selection algorithms in the following experiments:

Experiment 4: Comparison of RTBFS in terms of classification accuracy based on the selected features with other feature selection algorithms and the results in the literature.

Experiment 5: Inspection of the features selected by RTBFS using existing biological knowledge in the literature.

It has been pointed out in [5] and [53] that it is absolutely necessary to use external validation datasets to evaluate feature selection algorithms to avoid selection bias. Therefore, the external cross-validation (CV) as suggested in [5] is utilized to obtain unbias results. For the seven real-world omics datasets, a procedure similar to [5] is used. Particularly, a k-fold external CV is employed to the feature selection process. At each round of the external k-fold CV, the feature selection step is applied on the $(k-1)/k$ of the whole data (training data) to select the optimal biomarkers and test the selected biomarkers on the remaining $1/k$ of data (test data). For the convenience of comparison with the results in the literature, a 3-fold CV is used for the three real-world microarray datasets and the Leukemia-p dataset, of which the results were reported from 3-fold CV in [13] [65] [38]. For other datasets, a 10-fold CV is used as suggested in [5] which produces results with smaller variance. In order to compare the feature selection algorithms fairly, the CV data are generated in advance and all algorithms are applied to the same CV data.

6.4. Other Feature Selection Methods for Comparison

Apart from those twin boosting variants described in Section 4, another state-of-the-art ensemble feature selection method that combines support vector machine recursive feature elimination (SVM-RFE) [30] and bootstrapping resampling [1] is considered for comparison. The algorithm is implemented in Java-ML [2], a machine learning library developed in Java. This ensemble feature selection algorithm is termed EnseFS. One key control parameter of EnseFS is the number of selected features. In order to compare with RTBFS fairly, for each real-world omics dataset, the top s_e features ranked by EnseFS are selected, where $s_e = \lceil s_t \rceil$, s_t is the mean number of feature selected by RTBFS, and $\lceil s_t \rceil$ means the ceiling of s_t, i.e. the smallest integer greater than or equal to s_t. The other control parameters of EnseFS used in this study are set according to [1]. In addition to EnseFS, a state-of-the-art filter based feature selection algorithm, fast correlation-based filter (FCBF) method [69] is also taken into account for comparison study. All control parameters of FCBF are set following [69].

6.5. Classification Algorithms

In our experiments, a standard boosting algorithm with logistic loss function (LogitBoost) is employed for classification. Although LogitBoost might favor the STLB since they use the same logistic loss function, we try to avoid bias towards our robust eta-loss function for more objective results. The second classification algorithm used is support

vector machine (SVM) [62]. The implementation of a linear kernel SVM in Spider toolbox[2] is employed. The key parameter C of SVM controlling the trade-off between training errors and smoothness is set to 1 for all experiments.

7. Experimental Studies on Feature Selection Test Bed

In this section, the effects of the parameters to the performance of RTBFS are first investigated, and then the experimental results of the comparison study between RTBFS and other state-of-the-art feature selection methods are presented and discussed.

7.1. Experiment 1: Parameter Sensitivity Analysis

The main control parameters are m_{stop} of the twin boosting procedure and the contamination ratio η in equation (13). The omics feature selection test bed is used to perform the sensitivity analysis of the main control parameters in order to choose the optimal setting.

7.1.1. The Contamination Ratio η

The other control parameter is fixed to default value, i.e., $m_{\text{stop}} = 10$; then the performance of RTBFS in terms of the average proportional increase in error (υ) and the average numbers of true features (τ) for 5NN with different contamination ratio $\eta = 0.3$, 0.5, 0.7, and 0.9 are plotted in Fig. 3.

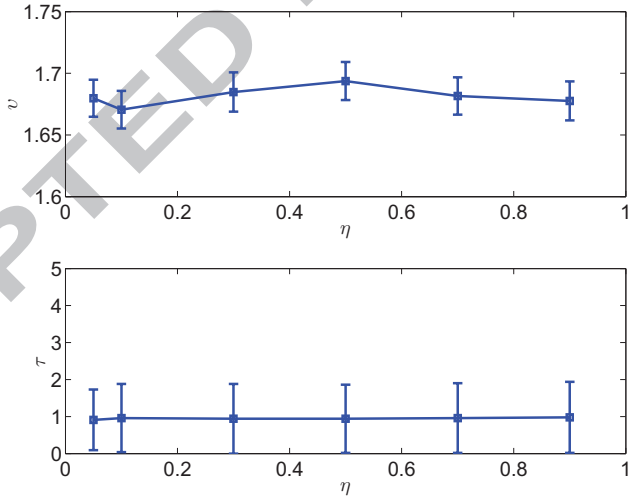


Figure 3: Performance of RTBFS in terms of the average proportional increase in error (υ) and the average number of true features (τ), for 5NN with different contamination ratio η.

[2] Available at http://www.kyb.tuebingen.mpg.de/bs/people/spider/main.html

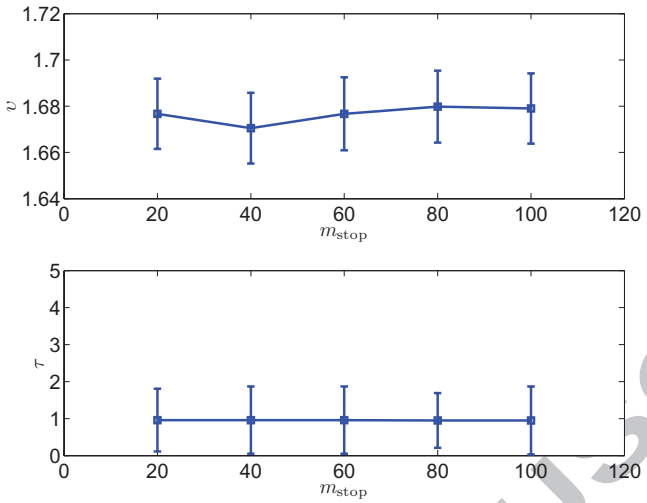


Figure 4: Performance of RTBFS in terms of the average proportional increase in error (υ) and the average number of true features (τ), for for 5NN with different number of stopping iterations m_{stop}.

It can be seen that the performance of RTBFS is not sensitive to η. It is suitable to choose $\eta = 0.9$ as default value since it only marginally decreases the performance compared with the optimal $\eta = 0.1$ but it can handle datasets with maximum label noise.

7.1.2. The Stopping Iterations m_{stop}

To investigate the effect of the stopping iterations m_{stop} on the performance of RTBFS, the other parameter is fixed to a default value and the performance of RTBFS with the following different m_{stop}: 5, 10, 15, 20 and 30 is plotted in Fig. 4.

It can be seen from the figure that, the performance of RTBFS is not sensitive to m_{stop}. However, when m_{stop} is too large, e.g, $m_{\text{stop}} = 30$ the performance in terms of both υ and τ deteriorates.

7.1.3. Parameter Selection for RTBFS

Based on the above sensitivity analysis, the parameters of RTBFS are set as $\eta = 0.9$ for all the experiments on the eight datasets, and $m_{\text{stop}} = 20$ for the omics feature selection test bed. However, for the real-world omics datasets, they generally consists more variables (see Table 1), therefore, $m_{\text{stop}} = 100$ is set for the seven real-world omics datasets.

7.2. Experiment 2: Results on the Test Bed with Label Noise

The omics feature selection test bed is employed to select the best weak learner for the standard Twin boosting algorithm. The test bed is contaminated by 0%, 5%, 10%, 20% and 30% label noise to simulate the real-world situations. The results using the average proportional increase in error, υ, with respect to the percentage of contamination, *perc*, from the representative 5NN classifier are reported.

7.2.1. Comparison Results with Other Twin Boosting Variations

The results of the three weak learners in terms of υ for 5NN with different contamination percentage are reported in Fig. 5. From the figure, it can be observed that RCLLS is the best weak learners in terms of selecting informative features from noisy data.

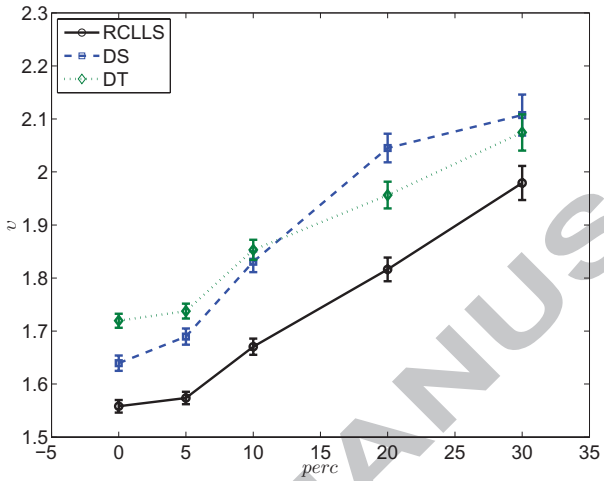


Figure 5: Performance of twin boosting algorithms with Componentwise Linear Least Squares (RCLLS), Decision Stump (DS) and Decision Tree (DT) in terms of the average proportional increase in error (υ) for 5NN. *perc* is the percentage of contamination.

The results of twin boosting algorithms with different loss functions are also plotted in Fig. 6. It can be seen from the figure that RTBFS or twin boosting with the robust eta-loss function performs better than other loss functions with increasing label noise. Surprisingly, the performance differences between the standard logistic loss function and the other two robustified loss functions, e.g., Savage and Huber loss function are not significant.

7.2.2. Comparative Results with Other Feature Selection Algorithms

The robustness of RTBFS is also compared with other feature selection algorithms described in Section 6.4. Since the focus is on the classification performance of the selected features, the results of these algorithms in terms of υ for 5NN, LDA, 3NN, and CART with different contamination percentage are plotted in Fig. 7.

It can be seen from the figure that all the five feature selection algorithms are affected by label noise. However, for RTBFS, in terms of υ, it has better performance than STLB. With the increase of *perc*, the difference between RTBFS and STLB increases. RTBFS also outperforms EnseFS and FCBFS for all the cases.

7.3. Experiment 3: Comparison with Other Feature Selection Algorithms on the Clean Original Test Bed

RTBFS algorithm is evaluated on the original test bed without any label noise in comparison with STLBFS, EnseFS and the best results of the popular sequential floating forward search (SFFS) algorithm [49] adopted from [9]. In this experiment, $t = 100$ samples of size $b = 50$ are drawn out of $n = 295$. The samples are saved and all the

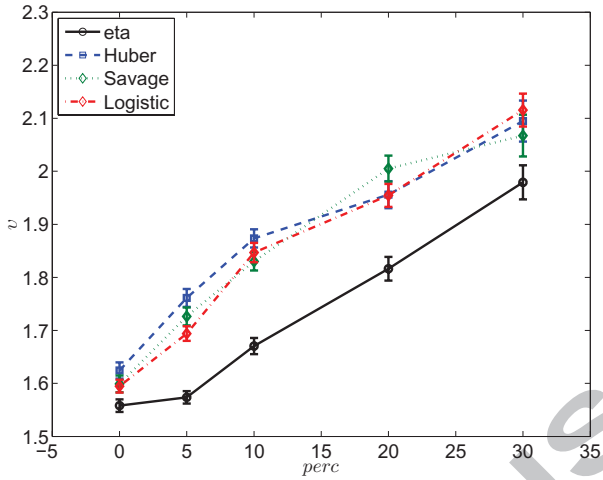


Figure 6: Performance of twin boosting algorithms with robust eta (RTBFS), Huber, Savage and standard Logistic loss functions (STBL) in terms of the average proportional increase in error (υ) for 5NN. *perc* is the percentage of contamination.

Table 2: Performance of RTBFS algorithm in comparison with STLBFS, EnseFS and SFFS with $t = 100$, $b = 50$, and $s = 4$. The results in bold font indicate the best results.

	LDA ($\epsilon = 0.1599$)			3NN ($\epsilon = 0.1360$)			5NN ($\epsilon = 0.1428$)			CART ($\epsilon = 0.1496$)		
	δ	υ	τ	δ	υ	τ	δ	υ	τ	δ	υ	τ
RTBFS	**0.0528**	**1.3305**	0.76	**0.0821**	**1.5950**	1.02	**0.0806**	**1.5581**	**1.44**	**0.0874**	**1.6013**	0.98
STLBFS	0.0599	1.3743	0.85	0.0863	1.6253	0.82	0.0859	1.5948	1.27	0.0912	1.6275	**1.04**
EnseFS	0.0614	1.3842	**0.90**	0.0851	1.6169	**1.09**	0.0850	1.5889	1.29	0.0930	1.6394	1.01
FCBFS	0.0643	1.4020	0.58	0.0922	1.6683	0.86	0.0916	1.6346	1.18	0.0980	1.6742	0.73
SFFS	0.0880	1.5505	0.61	0.1128	1.8290	0.66	0.1129	1.7904	0.66	0.1092	1.7300	0.74

feature selection algorithms are applied to the same samples. Follow the instructions in [9], the selected features are evaluated using classifiers: LDA, 3NN, 5NN, and CART. The results of feature set size $s = 4$ are listed in Table 2.

From the table, it is seen that SFFS and FCBFS are not competitive with the other two twin-boosting based feature selection algorithms. Comparing RTBFS with STLBFS, RTBFS has better results for all classifiers in terms of δ and υ. However, it is also interesting to note that the values of τ from the features selected by RTBFS are smaller than those from the features selected by STLBFS and EnseFS, the other performance indicators, e.g., δ and υ are still better, which indicates that although RTBFS misses some features in the optimal set (hence the smaller τ vaules), it selects the most important features that contribute more significantly to the accuracy of the classification.

8. Experimental Studies on Real-World Omics Datasets

The test bed is not sufficient to evaluate the real performance of RTBFS. Seven real-world omics datasets are further used to challenge RTBFS and a few representative state-of-the-art feature selection algorithms are considered

for comparison study.

8.0.1. Comparison Between RTBFS and Other Implemented Feature Selection Methods

RTBFS and other state-of-the-art feature selection algorithms are applied to the seven public available real-world omics datasets. The results of classification accuracy (Acc) and area under receiver operating characteristic curve (AUC), from RTBFS in comparison with other feature selection algorithms are reported in Table 3.

From our previous experiment on the omics feature selection test bed, it has been observed that different classification algorithms (even very simple ones) using the same feature set selected by the same feature selection algorithm actually generate significantly different results. LogitBoost and SVM as detailed in Section 6.5 have been shown to be relatively robust in many classification problems of small sample size, which is the case of omics data. Therefore, the results of LogitBoost and SVM using the features selected by the feature selection algorithms are tabulated in Table 3. In order to compare the results rigorously, a student's paired two-tailed *t*-test is also performed to test whether the results (Acc or AUC) from other feature selection algorithms are significantly different (p-value < 0.05) from the corresponding result of the RTBFS algorithm using the same classifier.

From Table 3, it is seen that in comparison with STLBFS, RTBFS always selects fewer features but generates significantly better results on most of the datasets. For example, the number of features selected by the RTBFS algorithm for Pancreatic dataset is only 19.99, much smaller than the 30.43 features selected by STLBFS, but achieves similar or even better classification results. By comparing the features of the Pancreatic dataset selected by RTBFS with those by STLBFS, it is found that although most of the top ranking features selected by RTBFS also appear in the feature sets selected by STLBFS, some features selected by RTBFS are not selected by STLBFS. These features might be more informative so that smaller number of features selected by RTBFS outperform those selected by STLBFS.

Comparing with EnseFS, RTBFS performs significantly better with LogitBoost on six datasets and competitively on Leukemia-p dataset. RTBFS also selects significantly better features for SVM on three datasets. It is also interesting to note that, EnseFS performs significantly worse on Prostate microarray dataset than the other algorithms.

In comparison to the filter-based feature selection algorithm, FCBFS, it can be seen from Table 3 that, FCBFS usually selects much more features than RTBFS. However, the classification performance using the features selected by FCBFS is not comparable to the corresponding results of the same classifiers using features selected by RTBFS. For example, FCBFS selects on average 29.6 features while RTBFS selects only 12.9 features on the seven datasets. Moreover, using these features, FCBFS with LogitBoost only obtains two similar results but five significantly worse results; and with SVM it obtains two significantly better one similar but four significantly worse results.

8.0.2. Comparison Results from RTBFS and Those from Literature

Many state-of-the-art feature selection algorithms have been proposed and applied separately to the seven real-world omics datasets. Some of the results from the literature are cited in comparison with our RTBFS algorithm. Note that comparing the results of this study directly with others published in the literature may not be appropriate due to

Table 3: Test results (%) of the seven real-world omics datasets from 30 independent runs of k-fold cross validation (CV). Notation: $+$, $-$, and $=$ indicate the corresponding result is statistically better, worse, and comparable, respectively, when compared with RTBFS.

	Algorithms	#Features	LogitBoost		SVM	
			Acc	AUC	Acc	AUC
Leukemia-m	RTBFS	8.86	93.10	92.73	83.60	82.80
	STLBFS	9.34	91.36^{-}	91.35^{-}	$82.30^{=}$	$81.97^{=}$
	EnseFS	9	90.77^{-}	90.58^{-}	93.27^{+}	93.50^{+}
	FCBFS	40.67	91.70^{-}	91.75^{-}	85.04^{+}	88.05^{+}
Colorectal-m	RTBFS	7.57	84.67	84.17	83.59	82.89
	STLBFS	9.13	82.58^{-}	83.05^{-}	$83.48^{=}$	$83.39^{=}$
	EnseFS	8	82.13^{-}	81.56^{-}	$84.07^{=}$	$83.73^{=}$
	FCBFS	21.49	$84.21^{=}$	$83.80^{=}$	85.93^{+}	85.24^{+}
Prostate	RTBFS	7.02	92.05	92.50	82.63	83.75
	STLBFS	8.72	90.25^{-}	91.07^{-}	79.16^{-}	80.28^{-}
	EnseFS	8	57.01^{-}	57.52^{-}	53.57^{-}	54.63^{-}
	FCBFS	60.60	92.13^{-}	92.55^{-}	76.86^{-}	77.49^{-}
Pancreatic	RTBFS	19.99	72.71	72.94	70.33	70.45
	STLBFS	30.43	69.12^{-}	69.45^{-}	$70.66^{=}$	$70.89^{=}$
	EnseFS	20	66.30^{-}	67.50^{-}	69.46^{-}	69.70^{-}
	FCBFS	9.23	65.27^{-}	65.24^{-}	65.70^{-}	65.75^{-}
Leukemia-p	RTBFS	8.32	91.13	92.12	90.19	91.01
	STLBFS	10.38	89.14^{-}	89.58^{-}	88.04^{-}	89.48^{-}
	EnseFS	11	$90.60^{=}$	$91.86^{=}$	78.37^{-}	82.51^{-}
	FCBFS	8.69	$89.97^{=}$	$91.13^{=}$	87.59^{-}	89.00^{-}
Liver-p	RTBFS	21.42	95.24	95.09	93.28	92.96
	STLBFS	22.45	93.77^{-}	93.38^{-}	$93.02^{=}$	$92.70^{=}$
	EnseFS	22	92.35^{-}	92.87^{-}	$92.23^{=}$	$92.46^{=}$
	FCBFS	59.32	94.38^{-}	93.85^{-}	$91.96^{=}$	$91.85^{=}$
Liver-m	RTBFS	17.20	88.06	85.89	75.67	66.56
	STLBFS	25.25	85.33^{-}	83.47^{-}	$73.65^{=}$	$67.05^{=}$
	EnseFS	18	79.37^{-}	74.69^{-}	77.61^{+}	72.83^{+}
	FCBFS	7.2	72.99^{-}	64.43^{-}	72.78^{-}	49.67^{-}

the different preprocessing, classifier, performance evaluation schema, experiment design, etc. used in different work. However, the reported results in the literature confirm the efficiency of RTBFS to a certain extent.

For the three microarray datasets, In [13], Dettling applied Wilcoxon test to select 200 genes and used BagBoosting for classification. The accuracy rates are 95.92%, 83.90% and 92.47% for Leukemia-m, Colorectal-m and Prostate, respectively. In [65], *t*-test was applied to select the top 20 genes and the samples were classified using a novel classification approach that utilized Kullback-Leibler divergence. The accuracy rates generated by their method on Leukemia-m, Colorectal-m and Prostate datasets are 97.42%, 83.08% and 89.52%, respectively. From the comparison, it is seen that on the three microarray datasets, RTBFS selects fewer features yet generates comparable classification performance.

In [26], Ge and Wong systematically compared different feature selection algorithms and classifiers on the Pancreatic dataset. Compared with their results, RTBFS performs better than all the methods except for the logistic regression algorithm using features selected by student *t*-test (the number of features are not reported). However, the result, e.g., 75% accuracy from their logistic regression algorithm was based on only one run of 10-fold CV, which might not objectively reflect the true discriminant performance of the features. Moreover, it is worth mentioning that the biomarkers selected in [26] are from *m/z* ratios; and many top biomarkers are adjacent *m/z* values, which might not be as informative as real peaks used in our study.

For the Leukemia-p dataset, Bayesian network (BN) based feature selection approach was applied in [38], which achieves 93.4% AUC using 96 features from just one independent run of 3-fold CV. Our RTBFS algorithm selects only on average 8.32 features that achieves average AUC of 92.12% from 30 runs of 3-fold CV.

On the the Liver-p dataset, Ressom *et al.* in [51] extracted eight biomarkers from a training data of 30 HCC and 30 cirrhosis spectra using ACO algorithm. They tested the eight biomarkers using SVM on a blind test data that consisted of 48 HCC and 21 cirrhosis spectra, which achieves 94% sensitivity and 100% specificity. Interestingly, using nine features selected by RTBFS algorithm from the same training data, 100% sensitivity and 97.92% specificity on the same blind test data are achieved. It is noted that the ACO algorithm is a heuristic optimisation algorithm and the algorithm was run 100 times in [51], which is time consuming and requires a lot of computational power.

To select features from the Liver-m dataset, Netzer *et al.* [47] proposed a novel ensemble-based feature selection algorithm termed stacked feature ranking (SFR). The ACU value of SVM using the top 20 features ranked by SFR on the same dataset (NAFLD + AFLD + cirrhosis versus healthy) is 87%, but the test data used by their 10-fold CV was not external to the feature selection procedure, therefore, feature selection bias was introduced and the result could be overoptimistic. The features selected by RTBFS are also compared with those reported in [47] in Section 8.1.2.

8.1. Features Selected by RTBFS

In order to further investigate whether the features selected by RTBFS are biologically plausible, Colorectal-m and Liver-m metabolomic datasets are taken as examples to discuss the details of features selected by RTBFS.

Table 4: Selected features (genes) from the Colorectal-m dataset using RTBFS. Features with asterisks are those also selected among the 35 top-ranked genes by a EPDM [41]. '#' denotes the number of times the gene is selected by RTBFS in the 3-fold CV.

#	Gene	Sequence	Name
3	H08393*	3' UTR	COLLAGEN ALPHA 2(XI) CHAIN (Homo sapiens)
3	R87126*	3' UTR	MYOSIN HEAVY CHAIN, NONMUSCLE (Gallus gallus)
3	M63391*	Gene	Human desmin gene, complete cds
2	H06524	3' UTR	GELSOLIN PRECURSOR, PLASMA (HUMAN)
2	M26383*	Gene	Human monocyte-derived neutrophil-activating protein (MONAP) mRNA, complete cds
2	X63629	Gene	H.sapiens mRNA for p cadherin.
2	R54097	3' UTR	TRANSLATIONAL INITIATION FACTOR 2 BETA SUBUNIT (HUMAN)
2	J02854	Gene	MYOSIN REGULATORY LIGHT CHAIN 2, SMOOTH MUSCLE ISOFORM (HUMAN);contains element TAR1 repetitive element
2	T62947	3' UTR	60S RIBOSOMAL PROTEIN L24 (Arabidopsis thaliana)
2	T47377*	3' UTR	S-100P PROTEIN (HUMAN)
2	R62549	3' UTR	PUTATIVE SERINE/THREONINE-PROTEIN KINASE B0464.5 IN CHROMOSOME III (Caenorhabditis elegans)
2	H43887*	3' UTR	COMPLEMENT FACTOR D PRECURSOR (Homo sapiens)
2	Z50753*	Gene	H.sapiens mRNA for GCAP-II/uroguanylin precursor

8.1.1. Selected Features of the Colorectal-m Dataset

The features selected from the Colorectal-m dataset by RTBFS in a typical run of 3-fold CV are reported in Table 4. The gene numbers with asterisks were also selected among the 35 top-ranked genes by a emerging patterns discovery method (EPDM) [41]. It can be seen from the table that, three genes selected three times by RTBFS in the 3-fold CV were also selected by EPDM in [41]. It is also worth mentioning that, these three genes, e.g., H08393 (COL11A1), R87126 (MYH9) and M63391 (Desmin) were also within the top six genes selected by EPDM.

It has been reported that three out of the top six genes namely H08393 (COL11A1), R87126 (MYH9), M63391 (Desmin) and M26383 (MONAP or Interleukin 8) are closely associated with colorectal cancer [19, 40, 6, 44]. It is not surprising since these genes were also selected by EPDM in [41]. However, it is more interesting to note that, the other two genes among the top six genes selected by RTBFS but not by EPDM, i.e., H06524 (Gelsolin) and X63629 (P-Cadherin), are recently found as colorectal cancer biomarkers in [18, 61].

8.1.2. Selected Features of the Liver-m Metabolomic Dataset

The features selected from the Liver-m dataset [47] by RTBFS are listed in Table 5. By comparing the features selected by the SFR algorithm in [47], it is noticed that the features selected by RTBFS have very little in common with that of SFR. Only Acetaldehyde, feature 38, and feature 106, which are selected at least once by RTBFS in the 10-fold CV, were ranked at the top 2nd, 7th and 9th biomarkers by SFR in [47]. These features are marked with asterisks in Table 5. It can be seen that, although feature sets selected by RTBFS are significantly different from those selected by SFR, some top features identified by RTBFS, e.g., isoprene, ethanol, and acetaldehyde, are known to be

Table 5: Selected features from the Liver-m metabolomic dataset using RTBFS over 10 rounds(R). The features with asterisks are those also selected by SFR in [47].

Rank	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10
1	103	I	I	20	20	20	101	E	101	103
2	I	20	101	E	101	E	20	101	20	117
3	CO2	101	20	101	32	101	I	61	I	20
4	102	E	H2S	CO2	47	I	CO2	64	32	I
5	101	32	CO2	65	I	H2S	123	CO2	CO2	47
6	20	102	E	I	P	32	117	I	P	CO2
7	117	H2S	32	H2S	H2S	117	B	32	106*	H2S
8	H2S	CO2	117	51	CO2	CO2	H2S	H2S	94	64
9	E	117	102	102	51	51	E	20	H2S	38*
10	64	38*	51	67	117	102	104	120	64	106*
11	32	64	104	32	102	64	85	40	117	120
12	27	106*	119	104	38*	90	64	117	102	101
13	106*	46	103	90	120	119	46	A*	46	46
14	46	120	64	47	64	106*	A*		33	40
15	120	93	40	P	104	33	38*		27	
16	111	67	38*	117	106*	103	94			
17	104	A*		106*	103	46				
18	40	33		64						

I: Isoprene; E: Ethanol; A: Acetaldehyde; P: Propene; B: Butadiene.

associated with liver disease [58, 48].

9. Conclusion

Omics data are generally high-dimensional and sometimes contains substantial label noise. It is very challenging to classify disease and normal samples based on such high-dimensional and noisy data. It is more important, however even more challenging to extract a parsimonious set of reliable features with high predictive performance from such data for further study.

In this study, based on the twin boosting framework, a novel feature selection algorithm, namely robust twin boosting (RTBFS) is proposed to extract parsimonious informative feature sets from high-dimensional omics data with label noise. RTBFS is first evaluated on the omics feature selection test bed where the optimal features are known. Several experiments are performed on the test bed with and without label noise. The results show that RTBFS is more robust than other feature selection algorithms when the data contains substantial label noise while maintaining a similar or better performance on the test bed. Then RTBFS is applied to seven real-world omics datasets. In order to avoid selection bias, an external CV is employed to obtain unbias classification results [5]. The results indicate that, compared with other state-of-the-art feature selection algorithms, RTBFS is capable of selecting parsimonious

yet informative feature sets and achieving better classification results. The features selected by RTBFS with literature results are further investigated and the results show that the features selected by RTBFS are biological plausible and medically relevant. It is worth mentioning that RTBFS is not only primarily proposed for noisy omics data, it can also be generally applied to other data with label noise.

10. Acknowledges

This work is supported by the Leverhulme Trust Early Career Fellowship (ECF/2007/0433), the Royal Society International Exchanges 2011 NSFC cost share scheme (IE111069), National Natural Science Foundation of China (61205092), the NSFC-RS joint project (61211130120), the Guangdong Foundation of Outstanding Young Teachers in Higher Education Institutions (Yq2013141), the Shenzhen Scientific Research and Development Funding Program (JCYJ20130329115450637, KQC201108300045A, and ZYC201105170243A), and the Guangdong Natural Science Foundation (S2012010009545).

References

[1] T. Abeel, T. Helleputte, Y. Van de Peer, P. Dupont, Y. Saeys, Robust biomarker identification for cancer diagnosis with ensemble feature selection methods, Bioinformatics 26 (2010) 392–398.

[2] T. Abeel, Y. Van de Peer, Y. Saeys, Java-ml: A machine learning library, The Journal of Machine Learning Research 10 (2009) 931–934.

[3] J. Abellán, A.R. Masegosa, An experimental study about simple decision trees for bagging ensemble on datasets with classification noise, in: Symbolic and Quantitative Approaches to Reasoning with Uncertainty, Springer, 2009, pp. 446–456.

[4] U. Alon, N. Barkai, D.A. Notterman, K. Gish, S. Ybarra, D. Mack, A.J. Levine, Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays, Proceedings of the National Academy of Sciences 96 (1999) 6745–6750.

[5] C. Ambroise, G. McLachlan, Selection bias in gene extraction on the basis of microarray gene-expression data, Proceedings of the National Academy of Sciences 99 (2002) 6562–6566.

[6] A. Avezzù, M. Agostini, S. Pucciarelli, M. Lise, E.D. Urso, I. Mammi, I. Maretto, M.V. Enzo, C. Pastrello, M. Lise, D. Nittib, A. Viela, The role of myh gene in genetic predisposition to colorectal cancer: another piece of the puzzle, Cancer Letters 268 (2008) 308–313.

[7] J. Bootkrajang, A. Kabán, Classification of mislabelled microarrays using robust sparse logistic regression, Bioinformatics 29 (2013) 870–877.

[8] P. Bülmann, T. Hothorn, Twin boosting: improved feature selection and prediction, Statistics and Computing 20 (2010) 119–138.

[9] A. Choudhary, M. Brun, J.P. Hua, J. Lowey, E.R. Dougherty, S. Miyano, Genetic test bed for feature selection, Bioinformatics 22 (2006) 837–842.

[10] J. Copas, Regression, prediction and shrinkage, Journal of the Royal Statistical Society. Series B (Methodological) (1983) 311–354.

[11] J. Copas, Binary regression models for contaminated data, Journal of the Royal Statistical Society. Series B (Methodological) 50 (1988) 225–265.

[12] J. Derrac, I. Triguero, S. Garcia, F. Herrera, Integrating instance selection, instance weighting, and feature weighting for nearest neighbor classifiers by coevolutionary algorithms, IEEE Transactions on Systems Man and Cybernetics Part B-Cybernetics 42 (2012) 1383–1397.

[13] M. Dettling, BagBoosting for tumor classification with gene expression data, Bioinformatics 20 (2004) 3583–3593.

[14] M. Dettling, P. Bülmann, Boosting for tumor classification with gene expression data, Bioinformatics 19 (2003) 1061–1069.

[15] R. Diao, Q. Shen, Feature selection with harmony search, IEEE Transactions on Systems Man and Cybernetics Part B-Cybernetics 42 (2012) 1509–1523.

[16] T.G. Dietterich, Ensemble methods in machine learning, Lecture Notes in Computer Science 1857 (2000) 1–15.

[17] D.L. Donoho, Nonlinear wavelet methods for recovery of signals, densities, and spectra from indirect and noisy data, in: In Proceedings of Symposia in Applied Mathematics, American Mathematical Society, 1993, pp. 173–205.

[18] N.J. Fan, C.F. Gao, C.S. Wang, J.J. Lv, G. Zhao, X.H. Sheng, X.L. Wang, D.H. Li, Q.Y. Liu, J. Yin, Discovery and verification of gelsolin as a potential biomarker of colorectal adenocarcinoma in the chinese population: Examining differential protein expression using an itraq labelling-based proteomics approach, Canadian Journal of Gastroenterology 26 (2012) 41–47.

[19] H. Fischer, R. Stenling, C. Rubio, A. Lindblom, Colorectal carcinogenesis is associated with stromal expression of col11a1 and col5a2., Carcinogenesis 22 (2001) 875–8.

[20] B. Frénay, G. Doquire, M. Verleysen, Theoretical and empirical study on the potential inadequacy of mutual information for feature selection in classification, Neurocomputing 112 (2013) 64 – 78.

[21] B. Frénay, G. Doquire, M. Verleysen, Estimating mutual information for feature selection in the presence of label noise, Computational Statistics & Data Analysis 71 (2014) 832–848.

[22] B. Frénay, M. Verleysen, Classification in the presence of label noise: a survey, IEEE Transactions on Neural Networks and Learning Systems 25 (2014) 845–869.

[23] Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, in: Proceedings of the Second European Conference on Computational Learning Theory (EuroCOLT'95), Springer-Verlag, London, UK, 1995, pp. 23–37.

[24] J. Friedman, Greedy function approximation: A gradient boosting machine, The Annals of Statistics 29 (2001) 1189–1232.

[25] J.Q. Gan, B.A.S. Hasan, C.S.L. Tsui, A filter-dominating hybrid sequential forward floating search method for feature subset selection in high-dimensional space, International Journal of Machine Learning and Cybernetics 5 (2014) 413–423.

[26] G. Ge, G.W. Wong, Classification of premalignant pancreatic cancer mass-spectrometry data using decision tree ensembles, BMC Bioinformatics 9 (2008) 275.

[27] R. Gerlach, J. Stamey, Bayesian model selection for logistic regression with misclassified outcomes, Statistical Modelling 7 (2007) 255–273.

[28] T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, H. Coller, M.L. Loh, J.R. Downing, M.A. Caligiuri, C.D. Bloomfield, E.S. Lander, Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, Science 286 (1999) 531–537.

[29] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, Journal of Machine Learning Research 3 (2003) 1157–1182.

[30] I. Guyon, J. Weston, S. Barnhill, V. Vapnik, Gene selection for cancer classification using support vector machines, Machine Learning 46 (2002) 389–422.

[31] T. Hastie, R. Tibshirani, J.H. Friedman, The Elements of Statistical Learning, Springer, 2003.

[32] S. He, X. Li, M.R. Viant, X. Yao, Profiling mass spectrometry proteomics data using smoothed nonlinear energy operator and bayesian additive regression trees, Proteomics 9 (2009) 4176–4191.

[33] S.R. Hingorani, E.F. Petricoin, A. Maitra, V. Rajapakse, C. King, M.A. Jacobetz, S. Ross, T.P. Conrads, T.D. Veenstra, B.A. Hitt, et al., Preinvasive and invasive ductal pancreatic cancer and its early detection in the mouse, Cancer Cell 5 (2004).

[34] P.J. Huber, Robust Statistics, John Wiley & Sons, Inc., 1981.

[35] H. Jiyan, G. Guan, W. Qun, Robust location algorithm based on weighted least-squares support vector machine (wls-svm) for non-line-of-sight environments, International Journal of the Physical Sciences 6 (2011) 5897–5905.

[36] T. Kanamori, T. Takenouchi, S. Eguchi, N. Murata, Robust loss functions for boosting, Neural Computation 19 (2007) 2183–2244.

[37] A. Karmaker, S. Kwek, A boosting approach to remove class label noise, International Journal of Hybrid Intelligent Systems 3 (2006) 169–177.

[38] K. Kuschner, D. Malyarenko, W. Cooke, L. Cazares, O.J. Semmes, E. Tracy, A bayesian network approach to feature selection in mass spectrometry data, BMC Bioinformatics 11 (2010) 177.

[39] Y.Y. Leung, C.Q. Chang, Y.S. Hung, An integrated approach for identifying wrongly labelled samples when performing classification in microarray data, PLoS One 7 (2012) e46700.

[40] A. Li, M.L. Varney, R.K. Singh, Expression of interleukin 8 and its receptors in human colon carcinoma cells with different metastatic potentials, Clinical Cancer Research 7 (2001) 3298–3304.

[41] J. Li, L. Wong, Corrigendum: Identifying good diagnostic gene groups from gene expression profiles using the concept of emerging patterns., Bioinformatics 18 (2002) 1406–1407.

[42] J. Liu, J. Li, W. Xu, Y. Shi, A weighted *lq* adaptive least squares support vector machine classifiers–robust and sparse approximation, Expert Systems with Applications 38 (2011) 2253–2259.

[43] P.M. Long, R.A. Servedio, Random classification noise defeats all convex potential boosters, Machine Learning 78 (2010) 287–304.

[44] Y. Ma, J. Peng, W. Liu, P. Zhang, L. Huang, B. Gao, T. Shen, Y. Zhou, H. Chen, Z. Chu, et al., Proteomics identification of desmin as a potential oncofetal diagnostic and prognostic biomarker in colorectal cancer, Molecular & Cellular Proteomics 8 (2009) 1878–1890.

[45] A. Malossini, E. Blanzieri, R.T. Ng, Detecting potential labeling errors in microarrays by data perturbation, Bioinformatics 22 (2006) 2114–2121.

[46] H. Masnadi-Shirazi, N. Vasconcelos, On the design of loss functions for classification: theory, robustness to outliers, and savageboost, in: D. Koller, D. Schuurmans, Y. Bengio, L. Bottou (Eds.), Advances in Neural Information Processing Systems 21, 2009, pp. 1049–1056.

[47] M. Netzer, G. Millonig, M. Osl, B. Pfeifer, S. Praun, J. Villinger, W. Vogel, C. Baumgartner, A new ensemble-based algorithm for identifying breath gas marker candidates in liver disease using ion molecule reaction mass spectrometry, Bioinformatics 25 (2009) 941–947.

[48] O. Niemelä, Distribution of ethanol-induced protein adducts in vivo: relationship to tissue injury, Free Radical Biology and Medicine 31 (2001) 1533–1538.

[49] P. Pudil, J. Novovičová, J. Kittler, Floating search methods in feature selection, Pattern Recognition Letters 15 (1994) 1119–1125.

[50] M. Rantalainen, C.C. Holmes, Accounting for control mislabeling in case - control biomarker studies, Journal of Proteome Research 10 (2011) 5562–5567.

[51] H. Ressom, R. Varghese, S. Drake, G. Hortin, M. Abdel-Hamid, C. Loffredo, R. Goldman, Peak selection from maldi-tof mass spectra using ant colony optimization, Bioinformatics 23 (2007) 619–626.

[52] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors, Nature 323 (1986) 533–536.

[53] Y. Saeys, I. Inza, P. Larranaga, A review of feature selection techniques in bioinformatics, Bioinformatics 23 (2007) 2507–2517.

[54] O.J. Semmes, L. Cazares, M. Ward, L. Qi, M. Moody, E. Maloney, J. Morris, M. Trosset, M. Hisada, S. Gygi, S. Jacobson, Discrete serum protein signatures discriminate between human retrovirus-associated hematologic and neurologic disease, Leukemia 19 (2005) 1229–1238.

[55] A.A. Shanab, T.M. Khoshgoftaar, R. Wald, Robustness of threshold-based feature rankers with data sampling on noisy and imbalanced data, in: Proceedings of the Twenty-Fifth International Florida Artificial Intelligence Research Society Conference (2012) 92–97.

[56] A. Sharma, S. Imoto, S. Miyano, V. Sharma, Null space based feature selection method for gene expression data, International Journal of Machine Learning and Cybernetics 3 (2012) 269–276.

[57] D. Singh, P.G. Febbo, K. Ross, D.G. Jackson, J. Manola, C. Ladd, P. Tamayo, A.A. Renshaw, A.V. D'Amico, J.P. Richie, E.S. Lander, M. Loda, P.W. Kantoff, T.R. Golub, W.R. Sellers, Gene expression correlates of clinical prostate cancer behavior, Cancer Cell 1 (2002) 203–209.

[58] B.G. Stone, T.J. Besse, W.C. Duane, C.D. Evans, E.G. DeMaster, Effect of regulating cholesterol biosynthesis on breath isoprene excretion in men, Lipids 28 (1993) 705–708.

[59] J.A. Suykens, J. De Brabanter, L. Lukas, J. Vandewalle, Weighted least squares support vector machines: robustness and sparse approximation, Neurocomputing 48 (2002) 85–105.

[60] M. Telgarsky, Margins, shrinkage, and boosting, Journal of Machine Learning Research 28 (2013).

[61] V. Van Marck, C. Stove, K. Jacobs, G. Van den Eynden, M. Bracke, P-cadherin in adhesion and invasion: Opposite roles in colon and bladder carcinoma, International Journal of Cancer 128 (2011) 1031–1044.

[62] V. Vapnik, The Nature of Statistical Learning Theory, Springer-Verlag, 1995.

[63] J. Vergara, P. Estévez, A review of feature selection methods based on mutual information, Neural Computing and Applications 24 (2014) 175–186.

[64] P. Wei, P. Ma, Q. Hu, X. Su, C. Ma, Comparative analysis on margin based feature selection algorithms, International Journal of Machine Learning and Cybernetics 5 (2014) 339–367.

[65] X. Wei, K.C. Li, Exploring the within- and between-class correlation distributions for tumor classification, Proceedings of the National Academy of Sciences 107 (2010) 6737–6742.

[66] W. Wen, Z. Hao, X. Yang, Robust least squares support vector machine based on recursive outlier elimination, Soft Computing 14 (2010) 1241–1251.

[67] V. Wheway, Using boosting to detect noisy data, in: Advances in Artificial Intelligence. PRICAI 2000 Workshop Reader, 2001, pp. 123–130.

[68] J.B. Yang, C.J. Ong, An effective feature selection method via mutual information estimation, IEEE Transactions on Systems Man and Cybernetics Part B-Cybernetics 42 (2012) 1550–1559.

[69] L. Yu, H. Liu, Efficient feature selection via analysis of relevance and redundancy, The Journal of Machine Learning Research 5 (2004) 1205–1224.

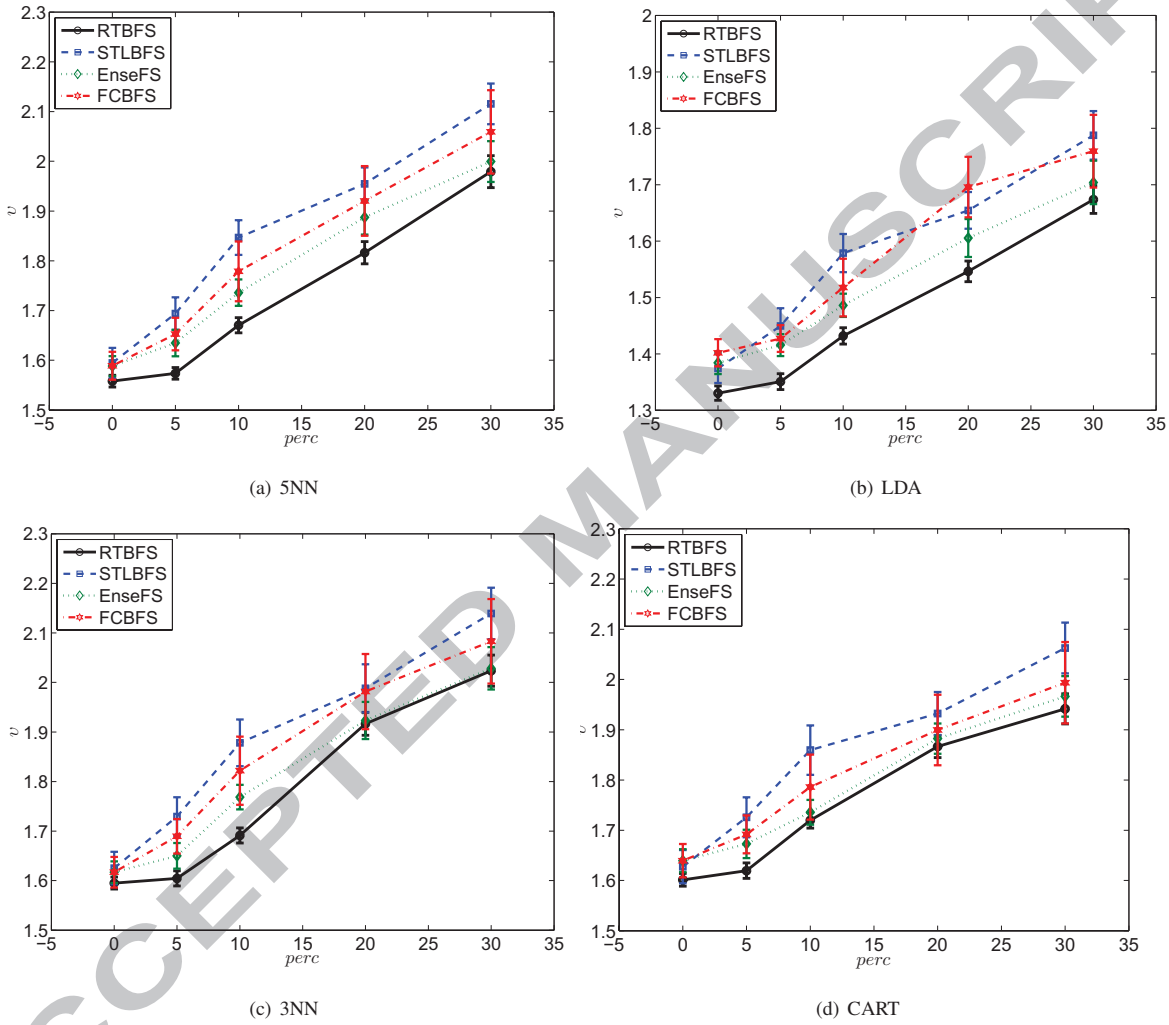
[70] C. Zhang, C. Wu, E. Blanzieri, Y. Zhou, Y. Wang, W. Du, , Y. Liang, Methods for labeling error detection in microarrays based on the effect of data perturbation on the regression model, Bioinformatics 25 (2009) 2708–2714.

[71] W. Zhang, R. Rekaya, K. Bertrand, A method for predicting disease subtypes in presence of misclassification among training samples using gene expression: application to human breast cancer, Bioinformatics 22 (2006) 317–325.

[72] Z. Zhu, S. Jia, Z. Ji, Towards a memetic feature selection paradigm, IEEE Computational Intelligence Magazine 5 (2010) 41–53.

[73] Z. Zhu, Y.S. Ong, M. Dash, Markov blanket embedded genetic algorithm for gene selection, Pattern Recogniton 40 (2007) 3236–3248.

[74] Z. Zhu, Y.S. Ong, M. Dash, Wrapper-filter feature selection algorithm using a memetic framework, IEEE Transactions on Systems Man and Cybernetics Part B-Cybernetics 37 (2007) 70–76.



(a) 5NN (b) LDA

(c) 3NN (d) CART

Figure 7: Performance of the five feature selection algorithms in terms of the average proportional increase in error, υ, for (a) 5NN, (b) LDA, (c) 3NN, and (d) CART with different contamination percentages *perc*. The smaller value of υ indicates the better performance of the feature selection algorithms.