Statistics and Its Interface Volume 10 (2017) 175–185

REC: fast sparse regression-based multicategory classification

Chong Zhang, Xiaoling Lu, Zhengyuan Zhu, Yin Hu,
Darshan Singh, Corbin Jones, Jinze Liu,
Jan F. Prins, and Yufeng Liu*

Recent advance in technology enables researchers to gather and store enormous data sets with ultra high dimensionality. In bioinformatics, microarray and next generation sequencing technologies can produce data with tens of thousands of predictors of biomarkers. On the other hand, the corresponding sample sizes are often limited. For classification problems, to predict new observations with high accuracy, and to better understand the effect of predictors on classification, it is desirable, and often necessary, to train the classifier with variable selection. In the literature, sparse regularized classification techniques have been popular due to the ability of simultaneous classification and variable selection. Despite its success, such a sparse penalized method may have low computational speed, when the dimension of the problem is ultra high. To overcome this challenge, we propose a new sparse REgression based multicategory Classifier (REC). Our method uses a simplex to represent different categories of the classification problem. A major advantage of REC is that the optimization can be decoupled into smaller independent sparse penalized regression problems, and hence solved by using parallel computing. Consequently, REC enjoys an extraordinarily fast computational speed. Moreover, REC is able to provide class conditional probability estimation. Simulated examples and applications on microarray and next generation sequencing data suggest that REC is very competitive when compared to several existing methods.

Keywords and phrases: LASSO, Parallel computing, Probability estimation, Simplex, Variable selection.

1. INTRODUCTION

Classification is an important supervised learning problem. With a training data set containing both predictors and labels, the main goal of classification is to build a classifier and predict labels for new instances with only predictors observed. Classification problems are prevalent in many scientific disciplines. For example, in cancer research, the patients can benefit most from the treatments if the corresponding prediction of their cancer subtype is accurate. In artificial intelligence, a precise classifier can help the machines to identify the characters of different people's handwriting with a very high accuracy. Various classifiers have been proposed in the literature. See, for example, Hastie et al. (2009) and Zhang and Singer (2010) for a comprehensive review.

In the last decade, fast growing technology has enabled us to gather massive data sets. For instance, in bioinformatics, microarray data sets often contain the expression levels for tens of thousands of genes (see, for example, Zhang et al., 2001, 2003, and the references therein). The next generation sequencing data can provide us with information on billions of read alignments. In neuroimaging (Yue et al., 2010; Zhang and Zhang, 2010), each image can be converted into matrices with millions of elements. In contrast to the ultra high dimensionality of such data sets, the number of observations is often small, partly due to the high cost of data collection. Such huge and complex data sets with only handful observations pose unprecedented challenges for existing analytical tools. In the literature, margin based classifiers from the machine learning community can handle high dimensional problems, and consequently are becoming increasingly popular. Typically, one can write the optimization problems of margin based classifiers in the *loss* + *penalty* form. The loss term measures the goodness of fit of the classifier, and the penalty controls the complexity of the classifier to prevent it from overfitting. Besides margin based classifiers, tree-based classification methods are also very effective and commonly used for practical problems (Zhang, 1998; Srivastava et al., 2002).

In this paper, we focus on margin based classifiers. Binary margin based classifiers have been extensively studied in the literature. For example, the penalized logistic regression (Lin et al., 2000), the Support Vector Machine (SVMs, Boser et al., 1992), and the Adaboost (Friedman et al., 2000) are well known binary margin based classifiers. Despite the success of binary margin based classifiers, their extension for multicategory classification can be challenging. To handle a multicategory classification problem with k classes using margin based methods, one possible approach is to use a sequence of binary classifiers, such as the one-versus-one and one-versus-rest procedures. In some situations, these methods may be suboptimal. For instance, in the

*Corresponding author.

one-versus-one scheme, when some classes have relatively small sample sizes, it was shown that weighted learning can be beneficial and crucial (Qiao et al., 2010). However, how to choose the weights among the sequence of binary classifiers remains unclear. Moreover, when using SVMs, the one-versus-one approach may have ties for different classes. For the one-versus-rest approach, it can be Fisher inconsistent if there is no dominating class (Liu, 2007). Therefore, it is desirable to study classifiers that consider all k classes simultaneously in one optimization problem.

In the simultaneous margin based classification literature, a common approach is to use a classification function vector of length k. In particular, each class is associated with one element of the classification function vector, and the prediction rule is based on which element is the largest. To reduce the parameter space and to obtain some theoretical results such as Fisher consistency, a sum-to-zero constraint on the classification function vector is commonly used. Many existing approaches were proposed in this framework, for example, Lee et al. (2004), Zhu and Hastie (2005), Liu and Shen (2006), Tang and Zhang (2006), Zhu et al. (2009), Park et al. (2010), Liu and Yuan (2011) and Zhang and Liu (2013).

In high dimensional classification problems, it is well known that when many noise predictors are included, proper variable selection can help to build a more accurate classifier (Fan and Li, 2006). For example, in microarray and next generation sequencing data sets, many potential housekeeping genes are included in the set of predictors. Without proper variable selection, these noise variables would be used in the resulting classifier, which can lead to bias in the estimation of the classification function. Hence, appropriate screening of the predictors can be crucial for classification accuracy and interpretability. In ultra high dimensional problems, classical model selection methods such as stepwise selection can be very unstable. To overcome this challenge, it is desirable to train classifiers with build-in variable selection (Zhang et al., 2006; Wang and Shen, 2007). However, for many existing simultaneous classifiers, solving the optimization problem with sparse penalties can be computationally intensive, and sometimes the algorithm may fail to converge. For example, for L_1 regularization, multicategory Support Vector Machines (SVMs) proposed by Vapnik (1998), Crammer et al. (2001), Lee et al. (2004), Wang and Shen (2007) and Liu and Yuan (2011) use linear programming. The ψ-learning proposed by Liu et al. (2005) and Liu and Shen (2006) uses difference convex algorithm, which essentially requires inner steps of linear programming. When the data set is high dimensional, linear programming can be very slow. Therefore, it is desirable to have classifiers with efficient algorithms to solve the corresponding large scale optimization problem.

To improve the computational speed, one can first consider to remove the sum-to-zero constraint in the optimization of regular multicategory classifiers. In particular, for binary margin based classifiers, it is sufficient to use one single function for classification. Analogously, for a k-class problem, it should suffice to use $k-1$ classification functions. However, the regular simultaneous classifiers use k classification functions and reduce to $k-1$ by the sum-to-zero constraint. This can be inefficient, and the optimization problem becomes more involved (Zhang and Liu, 2014). Recently, Lange and Wu (2008), Wu and Lange (2010), Wu and Wu (2012) and Zhang and Liu (2014) considered a new framework of simplex based classification, in which a classification function vector with length $k-1$ is used, instead of the regular k functions. In the $k-1$ dimensional Euclidean space where this new classification function vector lies, a centered simplex is constructed with k vertices, and each class is associated to one vertex of the simplex. This new classification framework is free of the explicit sum-to-zero constraint used by the regular simultaneous classifiers, and hence can enjoy a faster computational speed. Details of the simplex based classification can be found in Section 2.

In this paper, we propose a sparse REgression based multicategory Classifier (REC) that employs the simplex classification structure with the least distance prediction rule and the squared error loss. We show that with the L_1 penalty, the corresponding optimization can be decoupled into $k-1$ smaller LASSO problems (Tibshirani, 1996), and one can employ parallel computing to further boost the computational speed. Consequently, REC is very efficient for high dimensional classification problems with a large number of classes. We demonstrate in Sections 4 and 5 that REC enjoys an extraordinary fast computational speed. In particular, REC is able to handle ultra high dimensional problems with over 50,000 predictors on one processor, while the optimization problems of many other simultaneous classifiers with variable selection can be computationally intensive. Moreover, because the squared error loss is differentiable, we show that REC can naturally be used to estimate class conditional probabilities, which can be very helpful in many practical problems.

The rest of the paper is organized as follows. In Section 2, we review some existing multicategory classifiers, and propose our REC method. Section 3 studies the statistical properties of the REC approach. We compare the performance of REC and some other methods with simulated examples in Section 4. We conduct analyses on three real cancer research data sets in Section 5. Some discussions are provided in Section 6. All proofs are collected in the appendix.

2. METHODOLOGY

For a classification problem, we denote by Y the label, and by $\boldsymbol{X} = (X_1, \ldots, X_p)^T$ the predictor vector. The pair $(\boldsymbol{X}, Y)$ is assumed to follow a fixed but unknown distribution $\mathcal{P}(\boldsymbol{X}, Y)$. We observe the training data set $\{(\boldsymbol{x}_i, y_i);\ i = 1, \ldots, n\}$ *i.i.d.* from $\mathcal{P}(\boldsymbol{X}, Y)$, and build a classifier with prediction $\hat{y}(\cdot)$. For any new instance with only $\boldsymbol{x}$ observed, we

use $\hat{y}(\boldsymbol{x})$ as its predicted label. The goal is to minimize the classification error rate, namely, $\text{pr}(Y \neq \hat{y}(\boldsymbol{x}))$, where the probability is taken with respect to the joint distribution $\mathcal{P}(\boldsymbol{X}, Y)$.

In a multicategory classification problem with k classes, we use the label $Y \in \{1, \ldots, k\}$. To consider the multiple classes together, regular simultaneous margin based classifiers map $\boldsymbol{x}$ to a k-dimensional classification function vector $\boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x}))^T$. The corresponding prediction rule is $\hat{y}(\boldsymbol{x}) = \text{argmax}_{j \in \{1,\ldots,k\}} f_j(\boldsymbol{x})$. Note that f_j is not an estimate of the conditional probability for class j. We will discuss their connection for the analysis of consistency. To construct $\boldsymbol{f}(\boldsymbol{x})$, it is common to apply a sum-to-zero constraint on $\boldsymbol{f}$ to reduce the parameter space and to ensure some desirable theoretical properties of the classifier. Namely, we constrain $\boldsymbol{f}$ such that $\sum_{j=1}^k f_j(\boldsymbol{x}) = 0$ for all $\boldsymbol{x}$. Many existing simultaneous margin based classifiers employ this constraint. See, for example, Lee et al. (2004), Zhu and Hastie (2005), Liu and Shen (2006), Tang and Zhang (2006), Zhu et al. (2009), Liu and Yuan (2011) and Zhang and Liu (2013). Note that the fitted classification functions can be used to estimate class conditional probabilities after some transformation (such as the inverse logit link in binary logistic regression). See, for example, Zhang and Liu (2013) for a discussion.

As discussed in Section 1, for regular classifiers with k classification functions, the sum-to-zero constraint on $\boldsymbol{f}$ makes the optimization problem more involved. To remove the sum-to-zero constraint, one possible approach is to consider classification with a k-vertex simplex in a $(k-1)$-dimensional space (Lange and Wu, 2008; Wu and Lange, 2010; Wu and Wu, 2012; Zhang and Liu, 2014). In particular, let $\mathcal{Y} = \{\mathcal{Y}_j;\ j = 1, \ldots, k\}$, and define

$$\mathcal{Y}_j = \begin{cases} (k-1)^{-1/2}\mathbf{1}, & \text{if } j = 1, \\ -\frac{1+\sqrt{k}}{(k-1)^{3/2}}\mathbf{1} + \sqrt{\frac{k}{k-1}}e_{j-1}, & \text{if } 2 \leq j \leq k, \end{cases}$$

where $e_j \in \mathcal{R}^{k-1}$ is a vector of 0's except its j^{th} element is 1, and $\mathbf{1} = (1, \ldots, 1)^T$ (see Figure 1 for $k = 3$). It can be verified that each vector $\mathcal{Y}_j$ has Euclidean norm 1, and the distances between any pair $(\mathcal{Y}_i, \mathcal{Y}_j)$ are equal. Thus, $\{\mathcal{Y}_j;\ j = 1, \ldots, k\}$ form a symmetric simplex in the $(k-1)$-dimensional space. We then assign each class to a vector in $\{\mathcal{Y}_j;\ j = 1, \ldots, k\}$. Without loss of generality, suppose class j is assigned to $\mathcal{Y}_j;\ \ j = 1, \ldots, k$. The classification function vector maps $\boldsymbol{x}$ from the p dimensional space into $\boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_{k-1}(\boldsymbol{x}))^T \in \mathcal{R}^{k-1}$. In terms of prediction, Zhang and Liu (2014) proposed to use the least angle rule. In particular, observe that each $\boldsymbol{f}$ defines k angles with respect to $\mathcal{Y}_j$, namely, $\angle(\mathcal{Y}_j, \boldsymbol{f});\ j = 1, \ldots, k$. The least angle prediction rule is $\hat{y} = \text{argmin}_j \angle(\mathcal{Y}_j, \hat{\boldsymbol{f}}(\boldsymbol{x}))$, which is equivalent to $\hat{y} = \text{argmax}_j \langle \hat{\boldsymbol{f}}(\boldsymbol{x}), \mathcal{Y}_j \rangle$, where $\langle \boldsymbol{x}_1, \boldsymbol{x}_2 \rangle = \boldsymbol{x}_1^T \boldsymbol{x}_2$ is the inner product of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. See Figure 1 in Zhang and Liu (2014) for an illustration of the least angle prediction rule. Here, the terms $\langle \hat{\boldsymbol{f}}(\boldsymbol{x}), \mathcal{Y}_j \rangle$ can be regarded as functional margins in this simplex classification structure. Based on the least angle prediction rule, Zhang and Liu (2014) proposed the angle-based classifiers, which can be regarded as margin based classifiers in this new simplex-based classification framework. Compared to the original classification framework using k functions, the angle-based classifiers proposed by Zhang and Liu (2014) transfer the explicit sum-to-zero constraint onto the new functional margins in this simplex representation, namely, $\sum_{j=1}^k \langle \boldsymbol{f}(\boldsymbol{x}), \mathcal{Y}_j \rangle = 0$, which can be regarded as an implicit sum-to-zero property. Zhang and Liu (2014) showed that this can help to boost computational speed and improve classification performance.

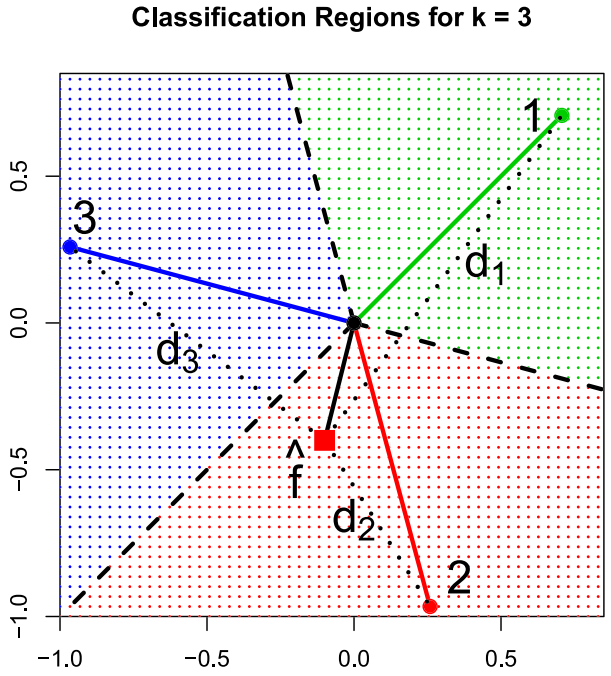

Figure 1. The prediction rule when $k = 3$. The solid green/red/blue lines correspond to classes 1/2/3. The mapped $\boldsymbol{f}$ is closest to $\mathcal{Y}_2$ (2 in the figure) as $d_2 < d_3 < d_1$, hence the predicted label is 2. Moreover, for any $\boldsymbol{x}$ whose mapped $\boldsymbol{f}$ is in the red region, its corresponding prediction would be class 2.

In the literature, Lange and Wu (2008), Wu and Lange (2010) and Wu and Wu (2012) proposed to use the least distance prediction rule $\hat{y}(\boldsymbol{x}) = \text{argmin}_j \|\hat{\boldsymbol{f}}(\boldsymbol{x}) - \mathcal{Y}_j\|_2$. The idea is that for any observation $(\boldsymbol{x}, y)$, we encourage the mapped classification function $\boldsymbol{f}(\boldsymbol{x})$ to be close to $\mathcal{Y}_y$. See Figure 1 for a simple example of the least distance prediction rule with $k = 3$. Lange and Wu (2008), Wu and Lange (2010) and Wu and Wu (2012) proposed to use the ϵ-insensitive loss function to measure the closeness of $\hat{\boldsymbol{f}}$ and $\mathcal{Y}_y$. Note that both prediction rules split the mapped $\mathcal{R}^{k-1}$ space into k non-overlapping sets $C_j;\ j = 1, \ldots, k$, where C_j corre-

sponds to class j. In other words, for a new observation $\boldsymbol{x}$, its predicted label would be class j if $\hat{\boldsymbol{f}}(\boldsymbol{x}) \in C_j$. Moreover, one can verify that these two prediction rules are essentially equivalent, in the sense that the non-overlapping sets $C_j;\ j = 1, \ldots, k$ of the least angle prediction rule are identical to that of the least distance rule. However, the classifiers are based on different motivations, and hence they use different classification loss functions.

In this paper, we propose to use the squared error loss in the distance based classification framework. In particular, we propose the REgression based multicategory Classifier (REC), where the corresponding optimization problem can be written as

$$\min_{\boldsymbol{\beta}_j, \beta_{0j};\ j=1,\ldots,k-1} \frac{1}{n} \sum_{i=1}^{n} \|\boldsymbol{f}(\boldsymbol{x}_i) - \mathcal{Y}_{y_i}\|_2^2 + \lambda J(\boldsymbol{f}). \tag{1}$$

For a given observation in the training data set, the squared loss term measures the closeness of the mapped $\boldsymbol{f}$ and its corresponding vertex of the simplex by the Euclidean distance. Because the data sets we consider are high dimensional, we focus on linear learning in this paper. Specifically, we let $f_j(\boldsymbol{x}) = \boldsymbol{\beta}_j^T \boldsymbol{x} + \beta_{0j}$ for $j = 1, \ldots, k-1$, and choose the L_1 regularization on $\boldsymbol{f}$ to prevent it from overfitting and to perform variable selection simultaneously. In other words, $J(\boldsymbol{f}) = \sum_{j=1}^{k-1} \|\boldsymbol{\beta}_j\|_1 = \sum_{j=1}^{k-1} \sum_{l=1}^{p} |\beta_j^{(l)}|$, where $\beta_j^{(l)}$ is the lth element of $\boldsymbol{\beta}_j$. Note that the lth variable does not contribute to the final classifier if and only if $\hat{\beta}_j^{(l)} = 0$ for all $j = 1, \ldots, k-1$. In (1), λ is a tuning parameter that controls the balance between the loss term and the penalty term. In practice, a proper choice of λ is crucial. More discussions on how to choose λ are provided in Sections 4 and 5.

The optimization (1) is a $(k-1)(p+1)$-dimensional problem, which is typically of the same dimension as the regular simultaneous margin based classifiers with the sum-to-zero constraint. However, REC is free of the explicit sum-to-zero constraint, and hence (1) enjoys a faster computational speed. More importantly, we show that (1) can be decoupled into $k-1$ separate standard LASSO regression problems, each with dimension $p+1$. Hence the computational speed of (1) can be greatly boosted further using parallel computing.

To begin with, observe that the loss term in (1) is a quadratic function, and hence can be decomposed into the sum of distances on each coordinate. The L_1 penalty in (1) is separable. Based on these facts, we can rewrite (1) as

$$\min_{\boldsymbol{\beta}_j, \beta_{0j}} \sum_{j=1}^{k-1} \Big[\frac{1}{n} \sum_{i=1}^{n} (f_j(\boldsymbol{x}_i) - \mathcal{Y}_{y_i}^{(j)})^2 + \lambda \sum_{l=1}^{p} |\beta_j^{(l)}|\Big], \tag{2}$$

where $\mathcal{Y}_{y_i}^{(j)}$ is the j^{th} element of $\mathcal{Y}_{y_i}$. Note that for a given j, each component in (2) is a LASSO regression problem as follows

$$\min_{\boldsymbol{\beta}_j, \beta_{0j}} \frac{1}{n} \sum_{i=1}^{n} \big((\boldsymbol{x}_i^T \beta_j + \beta_{0j}) - \mathcal{Y}_{y_i}^{(j)}\big)^2 + \lambda \sum_{l=1}^{p} |\beta_j^{(l)}|,$$

which can be solved very efficiently. In particular, there are several existing R packages that can be used to solve LASSO problems, including LARS (Efron et al., 2004) and GLMNET (Friedman et al., 2010). Note that the parameter sets involved in different LASSO problems are disjoint. By decoupling (1) into $k-1$ smaller optimization problems and using parallel computing, REC can enjoy an extraordinarily fast computational speed. We confirm this advantage through numerical studies in Sections 4 and 5.

For real applications, it is common to have problems where the numbers of observations in various classes are significantly different. Moreover, it is possible that the cost of misclassifying class j_1 to j_2 is different from that of class j_3 to j_4, where the ordered pairs $(j_1, j_2) \neq (j_3, j_4)$. In these cases, it is known that standard learning such as (1) can be suboptimal (Qiao and Liu, 2009). To alleviate this difficulty, one can use weighted learning with appropriately chosen weights for the loss terms of different classes. See Qiao and Liu (2009) and the references therein for more details on how to choose the weights. Our REC method can be generalized to the weighted REC as follows.

Suppose w_i is the weight for the ith observation. We can rewrite (1) as

$$\min_{\boldsymbol{\beta}_j, \beta_{0j};\ j=1,\ldots,k-1} \frac{1}{n} \sum_{i=1}^{n} w_i \|\boldsymbol{f}(\boldsymbol{x}_i) - \mathcal{Y}_{y_i}\|_2^2 + \lambda J(\boldsymbol{f}), \tag{3}$$

and one can verify that with linear learning and the L_1 penalty, (3) is equivalent to solving $k-1$ subproblems

$$\min_{\boldsymbol{\beta}_j, \boldsymbol{\beta}_{0j}} \frac{1}{n} \sum_{i=1}^{n} w_i \big((\boldsymbol{x}_i^T \beta_j + \beta_{0j}) - \mathcal{Y}_{y_i}^{(j)}\big)^2 + \lambda \sum_{l=1}^{p} |\beta_j^{(l)}|, \tag{4}$$

for $j = 1, \ldots, k-1$. Since many existing packages involve weighted learning, we can obtain the solution to (4) accordingly.

We would like to point out that in the literature, besides the L_1 penalty we used in (2) and (4), there are many other penalties that can perform automatic variable selection. For instance, when several predictors are correlated, LASSO tends to select only one among them (Efron et al., 2004). If it is desirable to select these predictors together, one can use a combination of the L_1 and L_2 penalties, such as the elastic net (Zou and Hastie, 2005). Moreover, one can use non-convex penalties, for example the smoothly clipped absolute deviation (Fan and Li, 2001) and the minimax concave penalty (Zhang, 2010), to estimate the parameters with the oracle property. Among these regularization methods, many are separable penalties. In this case, one can verify that the REC method can still enjoy the decomposition property, which leads to a great boost in the computational speed.

In the next section, we show that besides the advantage in computation, REC also enjoys some desirable statistical properties. In particular, we prove that REC is Fisher con-

sistent, and is able to provide class conditional probability estimation.

3. STATISTICAL PROPERTIES

3.1 Fisher consistency

Before introducing the Fisher consistency, we define some further notation. For a given $\boldsymbol{X} = \boldsymbol{x}$, let $P_j(\boldsymbol{x}) = \text{pr}(Y = j|\boldsymbol{X} = \boldsymbol{x})$ be the class conditional probability for class j. Furthermore, let $E(\|\boldsymbol{f} - \mathcal{Y}_Y\|_2^2|\boldsymbol{X} = \boldsymbol{x})$ be the conditional expected loss. A measurable function $\boldsymbol{f}^*(\boldsymbol{x})$ is called the theoretical minimizer of the conditional expected loss, if

$$E(\|\boldsymbol{f}^* - \mathcal{Y}_Y\|_2^2|\boldsymbol{X} = \boldsymbol{x}) = \inf_{\boldsymbol{f}} E(\|\boldsymbol{f} - \mathcal{Y}_Y\|_2^2|\boldsymbol{X} = \boldsymbol{x}).$$

For a classification problem, one can verify that the classifier $\hat{y}^*(\boldsymbol{x}) = \text{argmax}_j\, P_j(\boldsymbol{x})$ attains the minimal classification error rate for any future observation $\boldsymbol{x}$, and is often referred to as the Bayes classifier. Notice that minimizing the expected classification error rate $E\big(I(Y \neq \hat{y}(\boldsymbol{X}))\big)$ can be regarded as minimizing the $0-1$ loss function, which is discontinuous and non-convex. Consequently, directly finding a classifier that minimizes the classification error rate on the training data set can be difficult. To overcome this challenge, one can employ surrogate loss functions, such as the squared loss of our REC method in (1). Fisher consistency requires that, if one uses $\boldsymbol{f}^*$ as the classifier, the prediction would be identical to that of the Bayes classifier. In particular, for an observed $\boldsymbol{x}$, suppose $P_{j_0}(\boldsymbol{x})$ is the unique maximum among $\{P_1(\boldsymbol{x}), \ldots, P_k(\boldsymbol{x})\}$. Fisher consistency implies that $\text{argmin}_{j=1,\ldots,k} E(\|\boldsymbol{f}^*(\boldsymbol{x}) - \mathcal{Y}_j\|_2^2|\boldsymbol{X} = \boldsymbol{x}) = j_0$ and is unique. It ensures that with infinitely many training samples and $\boldsymbol{f}$ in a rich enough functional class, the resulting classifier achieves the optimal classification error rate. Fisher consistency is a fundamental requirement for classifiers. For the squared error loss in (1) of our REC method, we have the following theorem.

Theorem 1. *The squared error loss in (1) is Fisher consistent.*

Theorem 1 guarantees that if the underlying functional space is rich enough, REC is asymptotically consistent.

3.2 Class conditional probability estimation

In practice, estimation of the class conditional probabilities $P_j(\boldsymbol{x})$ can provide valuable information on the strength of the prediction. It can be an important by-product besides label prediction. How to estimate class conditional probability accurately has drawn much attention in the literature. See, for example, Wang et al. (2008), Wu et al. (2010), Appel et al. (2011), Zhang et al. (2013) and the reference therein. In this section, we show that our REC method can naturally provide class conditional probability estimation, and derive the corresponding formula.

To estimate $\{P_j(\boldsymbol{x});\ j = 1, \ldots, k\}$, a common approach is to explore the relationship between $P_j(\boldsymbol{x})$'s and $\boldsymbol{f}^*(\boldsymbol{x})$. In particular, assume that $P_j(\boldsymbol{x}) = g_j(\boldsymbol{f}^*)$ is a function of $\boldsymbol{f}^*$. Once the classifier $\hat{\boldsymbol{f}}$ is obtained, one can substitute $\boldsymbol{f}^*(\boldsymbol{x})$ by $\hat{\boldsymbol{f}}(\boldsymbol{x})$ in $g_j(\cdot)$, and this leads to an estimation of the class conditional probabilities. The following theorem gives the formulas $g_j(\cdot)$ explicitly for our REC method.

Theorem 2. *Denote by $\boldsymbol{P} = (P_1, \ldots, P_k)^T$ the vector of class conditional probabilities. Let $\mathbf{1}$ be the vector of length k with each element being 1, and let $\bar{\mathcal{Y}} = (\mathcal{Y}^T, \mathbf{1})^T$ be a k-dimensional square matrix. For the REC method, we have $\boldsymbol{P} = \bar{\mathcal{Y}}^{-1}(\boldsymbol{f}^{*T}, \mathbf{1})^T$.*

Consequently, for a given $\hat{\boldsymbol{f}}$, the estimated class conditional probabilities are $\hat{\boldsymbol{P}} = \bar{\mathcal{Y}}^{-1}(\hat{\boldsymbol{f}}^T, \mathbf{1})^T$. Note that we use transformation of $\boldsymbol{f}$ as in Theorem 2 to estimate the class conditional probabilities, which already involves the properties that $\sum_{j=1}^k P_j = 1$. See the proof of Theorem 2 in the appendix for more details.

We would like to point out that the formula given in Theorem 2 does not guarantee $\hat{P}_j \in [0, 1]$ for practical problems. To ensure that all $\hat{P}_j$'s are in the interval $[0, 1]$ and they sum up to 1, we consider the following modification on $\hat{P}_j$ with

$$\hat{P}_j^{scaled} = \frac{\hat{P}_j - \min_{i=1,\ldots,k} \hat{P}_i}{\sum_{m=1}^k (\hat{P}_m - \min_{i=1,\ldots,k} \hat{P}_i)}. \tag{5}$$

Note that a similar modification was previously used in Park et al. (2010) and Zhang and Liu (2013). The proposed scaling function is not unique, and there can be other scaling methods to make the resulting probabilities proper. Furthermore, the scaling function (5) does not affect the consistency of our method.

As a remark, we note that Lange and Wu (2008), Wu and Lange (2010) and Wu and Wu (2012) proposed to use the ϵ-insensitive loss $\|\mathcal{Y}_y - \boldsymbol{f}(\boldsymbol{x})\|_\epsilon = \max\{\|\mathcal{Y}_y - \boldsymbol{f}(\boldsymbol{x})\| - \epsilon, 0\}$ with some small ϵ to measure the closeness between $\boldsymbol{f}$ and $\mathcal{Y}_y$. Because the ϵ-insensitive loss is not differentiable, the corresponding methods do not provide class conditional probability estimation.

4. SIMULATED EXAMPLES

In this section, we demonstrate the performance of REC via three simulated examples. For each example, we generate data such that the label depends only on a few predictors (< 20), and we add noise covariates into the problem. We choose the dimension of covariates to be 100, 1,000, 10,000 and 50,000, and report the corresponding classification performance. To select the best tuning parameter, we build classifiers on a grid of 30 different tuning parameters using the training data set, and the best classifier that minimizes the prediction error rate on a separate tuning data set is selected. We then test the prediction accuracy of the selected classifier on a testing data set with 10^5 observations. We

Table 1. The average classification error rates, probability mean absolute error (MAE) and computational time (in seconds) for the simulated data sets

dim		Example 1			Example 2			Example 3		
		Error	MAE	Time	Error	MAE	Time	Error	MAE	Time
100	SVM1	**0.128**	-	32.61	0.131	-	11.37	0.183	-	10.33
	SVM2	0.131	-	41.92	0.126	-	13.26	0.181	-	11.25
	SVM3	0.141	-	35.28	0.133	-	12.27	**0.177**	-	11.19
	PSVM	0.135	0.082	16.68	0.140	0.147	6.672	0.179	0.155	6.532
	VDA	0.129	-	9.823	**0.122**	-	3.319	0.183	-	3.190
	REC	0.129	**0.078**	**5.382**	0.125	**0.121**	**1.106**	0.179	**0.129**	**0.971**
1,000	SVM1	0.139	-	100.4	0.148	-	17.82	0.213	-	15.48
	SVM2	0.153	-	133.2	**0.139**	-	23.61	0.207	-	21.92
	SVM3	0.161	-	127.4	0.151	-	21.97	**0.201**	-	19.79
	PSVM	0.147	0.091	88.38	0.144	0.150	12.34	0.229	0.161	10.53
	VDA	**0.136**	-	59.42	0.142	-	5.140	0.208	-	4.786
	REC	0.137	**0.080**	**26.81**	**0.139**	**0.127**	**1.859**	0.203	**0.135**	**1.655**
10,000	SVM1	-	-	-	-	-	-	-	-	-
	SVM2	-	-	-	-	-	-	-	-	-
	SVM3	-	-	-	-	-	-	-	-	-
	PSVM	0.159	0.093	919.3	0.166	0.153	189.4	0.239	0.166	214.4
	VDA	0.144	-	412.9	0.143	-	59.77	0.247	-	69.48
	REC	**0.139**	**0.081**	**154.6**	**0.141**	**0.131**	**5.793**	**0.238**	**0.138**	**5.900**
50,000	SVM1	-	-	-	-	-	-	-	-	-
	SVM2	-	-	-	-	-	-	-	-	-
	SVM3	-	-	-	-	-	-	-	-	-
	PSVM	-	-	-	-	-	-	-	-	-
	VDA	0.149	-	3557	0.167	-	341.8	0.216	-	339.9
	REC	**0.143**	**0.081**	**412.2**	**0.139**	**0.132**	**24.92**	**0.203**	**0.140**	**26.77**

perform 1,000 replicates for each example, and report the average misclassification error rates. We also report the time spent on each replicate as a measurement of computational speed. We compare the performance of REC with some existing simultaneous classifiers: Vapnik (1998) (SVM1), Crammer et al. (2001) (SVM2), Lee et al. (2004) (SVM3), Tang and Zhang (2006) (PSVM) and Wu and Lange (2010) (VDA). Note that we modify the PSVM method proposed in Tang and Zhang (2006) to incorporate linear learning.

For variable selection, each existing classifier is trained with the L_1 penalty. For class conditional probability estimation, we report the mean absolute error (MAE), $E|p-\hat{p}|$ only for REC and the existing PSVM method, as the simultaneous SVMs and VDA do not provide class conditional probability estimation directly. All simulation examples are performed using R, on a 2.30 GHz AMD processor.

Example 1. We conduct a twenty-class example in $\mathcal{R}^{19}$. The marginal distribution of $\boldsymbol{X}|Y$ is normal with equal variance. The mean vectors of different classes form a simplex in $\mathcal{R}^{19}$, and the variance parameters are chosen such that the Bayes error is at 0.05. To increase the dimension, we add noise covariates following *i.i.d.* $N(0, 0.01)$ distribution into the data set. We generate the training and tuning data sets, each of size 400.

Example 2. In this example, we generate a four class data set, where $\text{pr}(Y = j) = 1/4$; $j = 1, \ldots, 4$, and $P(\boldsymbol{X}|Y = j) \sim N(\mu_j, \sigma^2 I_2)$; $j = 1, \ldots, 4$, where μ_j are equally distributed on the unit circle, and σ is chosen such that the Bayes error is 0.1. The training and tuning data sets both have sample size 100. We then add noise covariates, which are *i.i.d.* from $N(0, 0.01)$.

Example 3. We generate a four class example on $\mathcal{R}^2$. For each class, the joint distribution of X_1 and X_2 is normal. The centers for the classes $1-3$ are uniformly distributed on the unit circle, while the fourth class has the center $(0, 0)$. The covariance matrices are chosen such that the Bayes error is 0.1. Then we add noise covariates from *i.i.d.* $N(0, 0.01)$ to the data. There are 100 observations used for the training and another 100 for the tuning procedure.

The simulation results are reported in Table 1. One can see that for problems with dimension $\geq$ 10,000, the SVM methods do not work because the linear programming takes too long to converge. For dimension = 50,000, the PSVM method does not work due to the large dimension. The VDA method is free of the commonly used sum-to-zero constraint and enjoys a faster computational speed, compared to the other existing classifiers. However, because the optimization problems for VDA are solved by the majorization-minimization (MM) algorithm, which requires inner loops for each MM step, the corresponding optimization can be computationally intensive for ultra high dimensional problems. Compared to existing simultaneous classifiers, REC

enjoys an extraordinary fast computational speed. One can see that when the dimensionality is ultra high, REC is at least 10 times faster than VDA and the other methods. In terms of classification error rates and probability estimation, REC is highly competitive as well.

5. REAL DATA ANALYSIS

In this section, we explore the classification performance of REC through the analysis of three high dimensional cancer research data sets. We compare REC with several existing methods used in Section 4, and demonstrate that REC can often work competitively in terms of classification accuracy and computational speed. Because some existing methods cannot analyze the full data sets directly due to the ultra high dimensionality of the predictors, we select certain subsets of the predictors from each data set to explore the classification performance. In particular, for each predictor, we compute its ratio of the within-group variance and the between-group variance. Then we select a subset of predictors that have the smallest ratios among all predictors in the data. For demonstration, we choose 1,000, 5,000 variables, and the entire data set for analysis in all three examples. To select the best tuning parameter, because we do not have separate tuning and testing data sets, within each replicate, we choose around 5/6 of the observations as the training data set, and the rest as the testing set. The best tuning parameter is selected via a five-fold cross validation. For each data set, we perform this data splitting process 1,000 times.

The first data set contains the measurement of microarray gene expression levels for patients with four different Glioblastoma Multiforme (GBM) Cancer subtypes, namely, Proneural, Neural, Classical and Mesenchymal (Verhaak et al., 2010). The expression levels of 12,042 genes for 202 patients are available. For the REC analysis of 1,000 genes, there are 122 genes that are selected more than 500 times out of 1,000 replicates. We show a heatmap of these 122 genes in Figure 2. One can see that each GBM subtype has its signature group of genes. This is consistent with Verhaak et al. (2010).

The second data set is obtained from The Cancer Genome Atlas (TCGA) research network (http://cancergenome.nih.gov/). This is a study of four types of squamous lung cancers (LUNG): Basal, Classical, Secretory and Primitive. The numbers of samples within each group are 43, 64, 27 and 42, respectively. We identify the alternative splicing locations using the aligned cumulative transcript graph (Singh et al., 2011). The total number of alternative splicing locations, including annotated and novel, are 422,000. At each of these locations, we extract the percentage spliced-in of the most expressed splices. The locations with total read coverage of less than 5 are considered unreliable and are marked as missing. After we pre-screen the missing values, there are 70,730 predictors available for the classification problem of the four subtypes of squamous lung cancer.

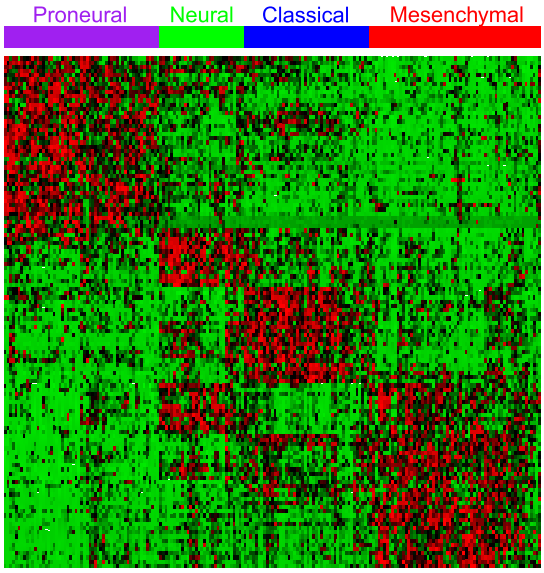

Figure 2. Heatmap for the top 122 genes selected in the GBM data set.

For the third data set, we analyze 819 RNA-seq samples from the TCGA breast cancer study (BREAST), including 91 normal samples and 728 tumor samples. The normal samples are from two molecular subtypes: normal ($n_1 = 79$) and Luminal A ($n_2 = 12$), and the tumor samples are from five molecular subtypes of breast cancer including Luminal A ($n_3 = 359$), Luminal B ($n_4 = 170$), Basal-like ($n_5 = 123$), HER2-enriched ($n_6 = 60$) and Normal-like ($n_7 = 16$). The different patients and subtypes may reflect different mixtures of cells and possibly different cancer mechanisms. We use the transcriptomic features extracted from the RNA-seq data as the predictors in this study. A total of 344,615 splice junctions are extracted from the RNA-seq read alignments, where the corresponding raw data can be found in the TCGA CGHUB repository (https://cghub.ucsc.edu/datasets/data_sets.html). The expression of each splice junction in every sample, as measured by the number of spliced reads, is calculated and used as the expression features of the transcriptomes. After pre-screening, we have 55,188 splice junctions without missing values. Because the sizes of different classes are quite unbalanced, we use weighted learning, where the weights are proportional to the reciprocal of class sizes (Qiao and Liu, 2009).

We report the mean classification error rates and mean computational time in Table 2, based on 1,000 replicates. From the results in Table 2, we can conclude that REC enjoys a very fast computational speed, and the corresponding classification accuracy is highly competitive. This is consistent with the findings in Section 4.

As a further comparison, we report the behavior of a tree based method and two random forest classifiers on the LUNG cancer data set. In particular, we employ the *RTREE* program (Zhang and Bracken, 1995; Zhang et al.,

Table 2. The classification error rates and computational time (clock time in seconds) for the cancer data sets for various margin-based classifiers

dim		GBM		LUNG		BREAST	
		Error	Time	Error	Time	Error	Time
1,000	SVM1	0.113	55.13	0.359	45.09	0.240	492.3
	SVM2	**0.105**	62.04	0.371	46.37	0.229	559.7
	SVM3	0.137	61.16	0.371	44.16	0.235	441.9
	PSVM	0.112	33.17	0.366	24.45	**0.226**	233.8
	VDA	0.113	13.44	**0.345**	9.179	0.228	119.2
	REC	0.107	**3.168**	0.351	**3.280**	0.231	**14.22**
5,000	SVM1	0.173	510.6	0.481	347.1	-	-
	SVM2	0.169	478.0	0.382	365.5	-	-
	SVM3	0.168	622.3	0.419	337.9	-	-
	PSVM	0.187	359.4	0.433	208.8	-	-
	VDA	**0.122**	79.92	0.335	64.21	0.221	911.3
	REC	0.128	**20.36**	**0.333**	**16.45**	**0.206**	**77.24**
Full data	SVM1	-	-	-	-	-	-
	SVM2	-	-	-	-	-	-
	SVM3	-	-	-	-	-	-
	PSVM	0.217	858.8	-	-	-	-
	VDA	0.146	160.3	0.337	727.1	-	-
	REC	**0.132**	**41.72**	**0.319**	**66.80**	**0.217**	**525.0**

Table 3. The classification error rates and computational time (clock time in seconds) for the LUNG cancer data set, for comparison among REC, tree based methods and the one-versus-rest support vector machine (SVM)

dim		LUNG	
		Error	Time
1,000	RTREE	0.355	5.000
	randomForest	0.362	7.232
	varSelRF	0.360	10.79
	SVM	0.366	15.52
	REC	**0.351**	**3.280**
5,000	RTREE	0.367	25.80
	randomForest	0.449	28.93
	varSelRF	0.355	122.7
	SVM	0.379	259.2
	REC	**0.333**	**16.45**
Full data	RTREE	0.431	247.0
	randomForest	0.531	299.6
	varSelRF	-	-
	SVM	-	-
	REC	**0.319**	**66.80**

1996), which can be found at http://c2s2.yale.edu/software/rtree/. For random forests, we use the R packages *randomForest* and *varSelRF*. The former package is an implementation of Breiman's random forest algorithm (Breiman, 2001), and the latter includes variable selection (Diaz-Uriarte and de Andrés, 2005). For the *RTREE* program, we use the one-versus-rest method, and perform the analysis 50 times. For *varSelRF*, we select the best tuning parameters via a five-fold cross validation (see Diaz-Uriarte and de Andrés, 2005, and the manual of the package for details about the tuning parameters). We repeat the procedure 1000 times for random forest methods. Furthermore, we apply a one-versus-rest approach using standard SVMs with L_1 penalization on the LUNG data to examine the corresponding performance in 1,000 replicates.

The results of *RTREE*, random forest methods and one-versus-rest SVM for the LUNG data are reported in Table 3. One can see that the computational time of tree based methods without variable selection is competitive, yet the corresponding classification performance deteriorates when the dimension increases. For random forest with variable selection, its classification accuracy improves, however the computational burden is much heavier. The one-versus-rest SVM performs poorly on this example. Overall, REC is highly competitive among existing methods.

6. DISCUSSION

In this paper, we propose the REC method for multicategory classification problems using the simplex representation for the class labels. Because REC is free of the sum-to-zero constraint, the corresponding optimization problem can enjoy a fast computational speed. More importantly, we show that with the L_1 penalty for variable selection, the optimization of REC can be decoupled into several smaller LASSO problems, which can be solved with an extraordinary fast speed. Statistical properties, such as Fisher consistency and class conditional probability estimation are obtained. We demonstrate that REC is highly competitive among existing methods via simulated and real data examples. In particular, for a glioblastoma multiforme cancer data set, we are able to identify signature gene groups for each cancer subtypes, which are consistent with existing literature. For a TCGA squamous lung cancer data set and a TCGA breast cancer data set, we show that our REC can achieve better classification performance compared with several existing classifiers.

For many bioinformatics problems, the predictors are often highly correlated. In this case, it is known that the L_1 penalty tends to choose only a few predictors among a group of highly correlated ones. For instance, in our glioblastoma multiforme cancer data, the signature gene groups identified by Verhaak et al. (2010) is larger than those reported in Figure 2. To overcome this difficulty, one can consider other penalties to encourage the highly correlated predictors to be selected simultaneously. For example, one can use the elastic net penalty (Zou and Hastie, 2005) or the group LASSO penalty (Yuan and Lin, 2006). Because these penalties are also separable, one can employ a similar decomposition approach as in Section 2, and solve the corresponding optimization problem efficiently.

APPENDIX

Proof of Theorem 1. Denote by S the conditional loss, and rewrite it as

$$S = \sum_{j=1}^{k} P_j \|\boldsymbol{f} - \mathcal{Y}_j\|_2^2.$$

Take partial derivative of S with respect to the mth element of $\boldsymbol{f}$, f_m, and we have

$$\frac{\partial S}{\partial f_m}|_{\boldsymbol{f}^*} = \sum_{j=1}^{k} 2P_j(f_m^* - \mathcal{Y}_j^{(m)}) = 0,$$

which can be further rewritten as

$$\mathcal{Y}\boldsymbol{P} = \boldsymbol{f}^*. \tag{6}$$

Now without loss of generality assume that $P_1 > P_2$. It suffices to show that $\|\boldsymbol{f}^* - \mathcal{Y}_1\|_2^2 < \|\boldsymbol{f}^* - \mathcal{Y}_2\|_2^2$. This is equivalent to showing that

$$\begin{aligned}\Delta :=& \|\boldsymbol{f}^* - \mathcal{Y}_1\|_2^2 - \|\boldsymbol{f}^* - \mathcal{Y}_2\|_2^2 \\ =& \|\mathcal{Y}\boldsymbol{P} - \mathcal{Y}e_1\|_2^2 - \|\mathcal{Y}\boldsymbol{P} - \mathcal{Y}e_2\|_2^2 < 0.\end{aligned}$$

After some calculation, we have $\Delta = 2(e_1 - e_2)^T\mathcal{Y}^T\mathcal{Y}\boldsymbol{P} - (e_1 - e_2)^T\mathcal{Y}^T\mathcal{Y}(e_1 + e_2)$. One can verify that $\mathcal{Y}^T\mathcal{Y} = (1 - t)I_k + tJ_k$, where I_k is the identity matrix, J_k is the matrix whose elements are all 1, and $t = \mathcal{Y}_i^T\mathcal{Y}_j$ is a constant regardless of the choice of i and j. Hence, one can verify that $(e_1 - e_2)^T\mathcal{Y}^T\mathcal{Y}(e_1 + e_2) = 0$, and $\mathcal{Y}^T\mathcal{Y}\boldsymbol{P} = (-(t + 1), t + 1, 0, \ldots, 0)^T$. Because $P_1 > P_2$ and $|t| < 1$, $\Delta = 2(e_1 - e_2)^T\mathcal{Y}^T\mathcal{Y}\boldsymbol{P} = 2(P_2 - P_1)(1 + t) < 0$. This completes the proof. □

Proof of Theorem 2. Note that (6) is a linear system of $\boldsymbol{P}$ with k unknowns and $k - 1$ equations. The result follows from combining (6) with the fact that $\sum_{j=1}^{k} P_j = 1$, solving for $\boldsymbol{P}$, and substituting $\boldsymbol{f}^*$ with $\hat{\boldsymbol{f}}$. □

ACKNOWLEDGEMENT

The authors would like to thank the Editor, Prof. Heping Zhang, for helpful suggestions. The authors were supported in part by US National Science Foundation and Engineering Research Council of Canada (NSERC), NSF grant DMS1407241, IIS1054631, NIH grants CA149569, HG06272, CA142538, P30CA177558, and National Natural Science Foundation of China (NSFC 61472475).

Received 22 December 2015

REFERENCES

Appel, I. J., Gronwald, W. and Spang, R. (2011). Estimating Classification Probabilities in High-dimensional Diagnostic Studies. *Bioinformatics* **27** 2563–2570.

Boser, B. E., Guyon, I. M. and Vapnik, V. N. (1992). A Training Algorithm for Optimal Margin Classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory. COLT'92* 144–152. ACM, New York, NY, USA.

Breiman, L. (2001). Random Forests. *Machine learning* **45** 5–32.

Crammer, K., Singer, Y., Cristianini, N., Shawe-taylor, J. and Williamson, B. (2001). On the Algorithmic Implementation of Multiclass Kernel-based Vector Machines. *Journal of Machine Learning Research* **2** 265–292.

Diaz-Uriarte, R. and de Andrés, S. A. (2005). Variable Selection from Random Forests: Application to Gene Expression Data.

Efron, B., Hastie, T. J., Johnstone, I. and Tibshirani, R. J. (2004). Least Angle Regression. *Annals of Statistics* **32** 407–499. MR2060166

Fan, J. and Li, R. (2001). Variable Selection via Nonconcave Penalized Likelihood and Its Oracle Properties. *Journal of the American Statistical Association* **96** 1348–1360. MR1946581

Fan, J. and Li, R. (2006). Statistical Challenges with High Dimensionality: Feature Selection in Knowledge Discovery. *Proceedings of the International Congress of Mathematicians* **3** 595–622. MR2275698

Friedman, J. H., Hastie, T. J. and Tibshirani, R. J. (2000). Additive Logistic Regression: A Statistical View of Boosting. *Annals of Statistics* **28** 337–407. MR1790002

Friedman, J., Hastie, T. J. and Tibshirani, R. J. (2010). Regularized Paths for Generalized Linear Models via Coordinate Descent. *Journal of Statistical Software* **33**.

Hastie, T. J., Tibshirani, R. J. and Friedman, J. H. (2009). *The Elements of Statistical Learning*, 2nd ed. New York: Springer. MR2722294

Lange, K. and Wu, T. T. (2008). An MM Algorithm for Multicategory Vertex Discriminant Analysis. *Journal of Computational and Graphical Statistics* **17** 527–544. MR2528236

Lee, Y., Lin, Y. and Wahba, G. (2004). Multicategory Support Vector Machines, Theory, and Application to the Classification of Microarray Data and Satellite Radiance Data. *Journal of the American Statistical Association* **99** 67–81. MR2054287

Lin, X., Wahba, G., Xiang, D., Gao, F., Klein, R. and Klein, B. (2000). Smoothing Spline Anova Models for Large Data Sets with Bernoulli Observations and the Randomized GACV. *Annals of Statitics* **28** 1570–1600. MR1835032

Liu, Y. (2007). Fisher Consistency of Multicategory Support Vector Machines. In *Eleventh International Conference on Artificial Intelligence and Statistics* 289–296.

Liu, Y., Shen, X. and Doss, H. (2005). Multicategory ψ-learning and Support Vector Machine: Computational Tools. *Journal of Computational and Graphical Statistics* **14** 219–236. MR2137899

Liu, Y. and Shen, X. (2006). Multicategory Ψ-learning. *Journal of the American Statistical Association* **101** 500–509. MR2256170

Liu, Y. and Yuan, M. (2011). Reinforced Multicategory Support Vector Machines. *Journal of Computational and Graphical Statistics* **20** 901–919. MR2878954

Park, S. Y., Liu, Y., Liu, D. and Scholl, P. (2010). Multicategory Composite Least Squares Classifiers. *Statistical Analysis and Data Mining* **3** 272–286. MR2726657

Qiao, X. and Liu, Y. (2009). Adaptive Weighted Learning for Unbalanced Multicategory Classification. *Biometrics* **65** 159–168. MR2665857

Qiao, X., Zhang, H. H., Liu, Y., Todd, M. J. and Marron, J. S. (2010). Weighted Distance Weighted Discrimination and Its Asymptotic Properties. *Journal of the American Statistical Association* **105** 401–414. MR2656058

Singh, D., Orellana, C. F., Hu, Y., Jones, C. D., Liu, Y., Chiang, D. Y., Liu, J. and Prins, J. F. (2011). FDM: A Graph-based Statistical Method to Detect Differential Transcription Using RNA-seq Data. *Bioinformatics* **27** 2633–2640.

Srivastava, A., Han, E. H., Kumar, V. and Singh, V. (2002). *Parallel Formulations of Decision-tree Classification Algorithms*. Springer.

Tang, Y. and Zhang, H. H. (2006). Multiclass Proximal Support Vector Machines. *Journal of Computational and Graphical Statistics* **15** 339–355. MR2256148

Tibshirani, R. (1996). Regression Shrinkage and Selection via the LASSO. *Journal of the Royal Statistical Society, Series B* **58** 267–288. MR1379242
Vapnik, V. (1998). *Statistical Learning Theory*. Wiley. MR1641250
Verhaak, R. G., Hoadley, K. A., Purdom, E., Wang, V., Qi, Y., Wilkerson, M. D., Miller, C. R., Ding, L., Golub, T., Mesirov, J. P., Alexe, G., Lawrence, M., O'Kelly, M., Tamayo, P., Weir, B. A., Gabriel, S., Winckler, W., Gupta, S., Jakkula, L., Feiler, H. S., Hodgson, J. G., James, C. D., Sarkaria, J. N., Brennan, C., Kahn, A., Spellman, P. T., Wilson, R. K., Speed, T. P., Gray, J. W., Meyerson, M., Getz, G., Perou, C. M., Hayes, D. N. and Cancer Genome Atlas Research Network (2010). Integrated Genomic Analysis Identifies Clinically Relevant Subtypes of Glioblastoma Characterized by Abnormalities in PDGFRA, IDH1, EGFR, and NF1. *Cancer Cell* **17** 98–110.
Wang, L. and Shen, X. (2007). On L_1-norm Multi-class Support Vector Machines: Methodology and Theory. *Journal of the American Statistical Association* **102** 595–602. MR2370855
Wang, J., Shen, X. and Liu, Y. (2008). Probability Estimation for Large Margin Classifiers. *Biometrika* **95** 149–167. MR2409720
Wu, T. T. and Lange, K. (2010). Multicategory Vertex Discriminant Analysis for High-Dimensional Data. *Annals of Applied Statistics* **4** 1698–1721. MR2829933
Wu, T. T. and Wu, Y. (2012). Nonlinear Vertex Discriminant Analysis with Reproducing Kernels. *Statistical Analysis and Data Mining* **5** 167–176. MR2910025
Wu, Y., Zhang, H. H. and Liu, Y. (2010). Robust Model-Free Multiclass Probability Estimation. *Journal of the American Statistical Association* **105** 424–436. MR2656060
Yuan, M. and Lin, Y. (2006). Model Selection and Estimation in Regression with Grouped Variables. *Journal of the Royal Statistical Society, Series B* **68** 49–67. MR2212574
Yue, Y., Loh, J. M. and Lindquist, M. A. (2010). Adaptive Spatial Smoothing of FMRI Images. *Statistics and its Interface* **3** 3–13. MR2609707
Zhang, H. P. (1998). Classification Trees for Multiple Binary Responses. *Journal of the American Statistical Association* **93** 180–193.
Zhang, C. H. (2010). Nearly Unbiased Variable Selection under Minimax Concave Penalty. *Annals of Statistics* 894–942. MR2604701
Zhang, H. P. and Bracken, M. B. (1995). Tree-based Risk Factor Analysis of Preterm Delivery and Small-for-gestational-age Birth. *American Journal of Epidemiology* **141** 70–78.
Zhang, H. P., Holford, T. and Bracken, M. B. (1996). A Tree-based Method in Prospective Studies. *Statistics in Medicine* **15** 37–49.
Zhang, C. and Liu, Y. (2013). Multicategory Large-margin Unified Machines. *Journal of Machine Learning Research* **14** 1349–1386. MR3081927
Zhang, C., Liu, Y. and Wu, Z. (2013). On the Effect and Remedies of Shrinkage on Classification Probability Estimation. *The American Statistician* **67** 134–142. MR3303796
Zhang, C. and Liu, Y. (2014). Multicategory Angle-based Large-margin Classification. *Biometrika* **101** 625–640. MR3254905
Zhang, H. P. and Singer, B. (2010). *Recursive Partitioning and Its Applications*, 2nd ed. Springer Verlag.
Zhang, H. P., Yu, C. Y. and Singer, B. (2003). Cell and Tumor Classification using Gene Expression Data: Construction of Forests. *Proceedings of the National Academy of Sciences* **100** 4168–4172.
Zhang, C. M. and Zhang, Z. J. (2010). Regularized Estimation of Hemodynamic Response Function for FMRI Data. *Statistics and its Interface* **3** 15–32. MR2609708
Zhang, H. P., Yu, C. Y., Singer, B. and Xiong, M. (2001). Recursive Partitioning for Tumor Classification with Gene Expression Microarray Data. *Proceedings of the National Academy of Sciences* **98** 6730–6735.
Zhang, H. H., Ahn, J., Lin, X. and Park, C. (2006). Gene Selection using Support Vector Machines with Non-convex Penalty. *Bioinformatics* **22** 88–95.
Zhu, J. and Hastie, T. J. (2005). Kernel Logistic Regression and the Import Vector Machine. *Journal of Computational and Graphical Statistics* **14** 185–205. MR2137897
Zhu, J., Zou, H., Rosset, S. and Hastie, T. J. (2009). Multi-class Adaboost. *Statistics and Its Interface* **2** 349–360. MR2540092
Zou, H. and Hastie, T. J. (2005). Regularization and Variable Selection via the Elastic Net. *Journal of the Royal Statistical Society, Series B* **67** 301–320. MR2137327

Chong Zhang
Department of Statistics and Actuarial Science
University of Waterloo
Canada
E-mail address: chong.zhang@uwaterloo.ca

Xiaoling Lu
Center for Applied Statistics
School of Statistics
Renmin University of China
China
E-mail address: xiaolinglu@ruc.edu.cn

Zhengyuan Zhu
Department of Statistics
Iowa State University
USA
E-mail address: zhuz@iastate.edu

Yin Hu
Sage Bionetworks
USA
E-mail address: snowy8677@gmail.com

Darshan Singh
Department of Computer Science
University of North Carolina at Chapel Hill
USA
E-mail address: darshan@cs.unc.edu

Corbin Jones
Department of Biology
University of North Carolina at Chapel Hill
USA
E-mail address: cdjones@email.unc.edu

Jinze Liu
Department of Computer Science
University of Kentucky
USA
E-mail address: liuj@cs.uky.edu

Jan F. Prins
Department of Computer Science
University of North Carolina at Chapel Hill
USA
E-mail address: prins@email.unc.edu

Yufeng Liu
Department of Statistics and Operations Research
Department of Genetics
Department of Biostatistics
Carolina Center for Genome Sciences
UNC Lineberger Comprehensive Cancer Center
University of North Carolina at Chapel Hill
USA
E-mail address: yfliu@email.unc.edu