Revista Colombiana de Estadística
January 2020, Volume 43, Issue 1, pp. 103 to 125
DOI: http://dx.doi.org/10.15446/rce.v43n1.80000

Nested and Repeated Cross Validation for Classification Model With High-dimensional Data

Validación cruzada anidada y repetida para el modelo de clasificación con datos de alta dimensión

Yi Zhong[a], Jianghua He[b], Prabhakar Chalise[c]

Department of Biostatistics and Data Science, University of Kansas Medical Center, Kansas City, Kansas, United States

Abstract

With the advent of high throughput technologies, the high-dimensional datasets are increasingly available. This has not only opened up new insight into biological systems but also posed analytical challenges. One important problem is the selection of informative feature-subset and prediction of the future outcome. It is crucial that models are not overfitted and give accurate results with new data. In addition, reliable identification of informative features with high predictive power (feature selection) is of interests in clinical settings. We propose a two-step framework for feature selection and classification model construction, which utilizes a nested and repeated cross-validation method. We evaluated our approach using both simulated data and two publicly available gene expression datasets. The proposed method showed comparatively better predictive accuracy for new cases than the standard cross-validation method.

***Key words*:** Area under ROC curve; Cross-validation; Elastic net; Random forest; Support vector machine.

Resumen

Con la llegada de las tecnologías de alto rendimiento, los conjuntos de datos de alta dimensión están cada vez más disponibles. Esto no sólo ha abierto una nueva visión acerca de los sistemas biológicos, sino que también plantea desafíos analíticos. Un problema importante es la selección de subconjuntos de variables y la predicción de resultados futuros. Es crucial que los modelos no sean sobreajustados y que den resultados precisos con

[a]PhD. E-mail: yi.zhong1006@gmail.com
[b]PhD. E-mail: jhe@kumc.edu
[c]PhD. E-mail: pchalise@kumc.edu

nuevos datos. Además, la identificaci ón confiable de variables informativas con alto poder predictivo (selección de características) es de interés en entornos clínicos. Proponemos un procedimiento de dos etapas para la selección de variables y la construcción de modelos de clasificación, el cual utiliza un método de validación cruzada anidada y repetida. Evaluamos nuestro enfoque utilizando tanto datos simulados como dos conjuntos de datos de expresión génica disponibles públicamente. El método propuesto mostró una precisión predictiva comparativamente mejor para casos nuevos en comparación con el método estándar de validación cruzada.

***Palabras clave*:** Área bajo la curva ROC; Validación cruzada; Red elástica; Bosque aleatorio; Máquina de vectores de soporte.

1. Introduction

Genetic basis of research for complex diseases such as cancer has been increasingly popular in recent years due to the invent of high throughput technologies such as microarray and sequencing technologies. Such technologies query the expression of thousands of genes simultaneously (Trevino, Falciani & Barrera-Saldana 2007). Many cancer researches over the past several years have been devoted to determine differentially expressed genes between tumor cells and normal cells (Zhang, Zhou, Velculescu, Kern, Hruban, Hamilton, Vogelstein & Kinzler 1997). The information obtained from gene expression analysis often helps in predicting patients' clinical outcomes.

Also there have been researches aiming to explore the possibilities of cancer diagnostics and classification using gene expression data (Van't Veer, Dai, Van De Vijver, He, Hart, Mao, Peterse, Van Der Kooy, Marton, Witteveen et al. 2002, Pomeroy, Tamayo, Gaasenbeek, Sturla, Angelo, McLaughlin, Kim, Goumnerova, Black, Lau et al. 2002). However, due to the unique structure of gene expression data, researchers are facing some major challenges. First, gene expression datasets have very high dimensionality; they usually contain thousands of genes assayed on only a few subjects, usually a couple of hundreds. Second, most genes are irrelevant to disease classification. Therefore, selecting a few genes that are associated with disease is important. Selecting subset of genes not only helps reducing the dimensionality of data but also helps improving the classification accuracy (Golub, Slonim, Tamayo, Huard, Gaasenbeek, Mesirov, Coller, Loa, Downing, Caligiuri, Bloomfield & Lander 1999, Lu & Han 2003).

There are three general methods of feature selection including filter methods, wrapper methods, and embedded methods (Guyon 2006). Filter methods use variable ranking techniques for variable selection. For example, the Chi-square statistic is computed for each feature, and these features are ranked based on the Chi-square statistics, then a threshold is determined to remove irrelevant features. Wrapper methods use search strategies (exhaustive search, forward selection, etc.) to generate various combinations of feature subsets. Then, the best combination of features is evaluated by a learning algorithm. Wrapper methods keep adding and/or removing features to find the best feature subsets that maximizes the model

performance (Dash & Liu 1997). Embedded methods build a predictive model and select features simultaneously. For embedded methods, the feature subset is determined by the predictive model when the final model is chosen (Guyon 2006). For example, least absolute shrinkage and selection operator (Lasso) is an embedded feature selection method, in which the feature subset is chosen by the final model. There are many articles published discussing about the feature selection methods. For example, Hira & Gillies (2015) reviewed the details of three methods, and listed several practical algorithms of feature selection methods. Saeys, Inza & Larranaga (2007) summarized the three feature selection methods, and introduced the application of feature selection methods in biostatistics. Kumar & Minz (2014) illustrated the processes of feature selection methods, and also detailed the algorithms for each feature selection method with their computational details. Each method has its own advantages and disadvantages. In this manuscript, we utilize embedded methods because of the following strengths: (1) embedded methods consider the correlation among predictor variables as well, rather than the relationship between outcome and predictors only like filter methods; (2) embedded methods are computationally less intensive than wrapper methods; (3) embedded methods can select features and build classification model simultaneously so that we can study the selected features, as well as predict the future outcome when new data are introduced.

For embedded methods, building the predictive model is the most critical part. After the predictive model is built, the subset of features is also selected. To build the predictive model, the original gene expression dataset is partitioned into training and test datasets. The training dataset is used to build the model while the test dataset is used to assess the test error (generalization error) of the chosen final model. Cross-validation is generally used to find the optimal model by controlling the overfitting of data (Hastie, Tibshirani & H. 2009, Braga-Neto & Dougherty 2004). However, the implementation of a single cross-validation may not perform well, mainly due to the randomness of generation the cross-validation folds (Krstajic, Buturovic, Leahy & Thomas 2014). Krstajic et al. (2014) indicated some pitfalls of using a single cross-validation and have proposed a repeated cross-validation to replace single cross-validation in model selection. Also they have demonstrated that repeated cross-validation method can result in a more robust and stable model. On the other hand, nested cross-validation creates multiple layers of cross-validation which can be used in both model selection and model assessment (Stone 1974). For example, in a two-layer cross-validation, a set of tuning parameters is tuned in the inner loop, and the other tuning parameters are estimated to determine the final predictive model in the outer loop. Another way to use nested cross-validation for model assessment is that the tuning parameters are estimated and the final model is selected in the inner loop, and the model performance is evaluated in the outer loop. Whelan, Watts, Orr, Althoff, Artiges, Banaschewski, Barker, Bokde, Büchel, Carvalho et al. (2014) applied a three-layer nested cross-validation technique to optimize the imaging threshold in the inner loop, to select the tuning parameters of logistic regression via elastic net penalty in the middle loop, and to assess the model performance using the area under the ROC (receiver operating characteristics) curve from the outer loop.

As mentioned before, both nested cross-validation and repeated cross-validation are designed for model selection. Nested cross-validation utilizes multi-layer cross-validation to tune more parameters, and repeated cross-validation repeats the procedure of generating K-folds to alleviate the randomness of fold generation. In this manuscript, we propose a new two-step framework for feature selection and model selection, and apply the proposed algorithm in microarray gene expression data analysis. The training data is first partitioned in K folds, then, within each kth fold, V folds are nested. Our proposed method has two steps: in step 1, we utilize above mentioned classifiers (linear regression via elastic net, Support vector machine, and random forest) to select the features in the inner layer of cross-validation loop; in the step 2, we utilize the classifiers to build classification model using the selected feature subset in the step 1. In addition, we implement the proposed approach both in the simulated data and real life data assessing its performance and present the comparison with different embedded variable selection methods (elastic net, SVM, random forest) with respect to predictive performance and selection accuracy. To the best of our knowledge, although the idea of using nested/repeated cross-validation has been mentioned elsewhere, (i.e. Stone, 1974 firstly briefed the idea of double cross-validation in the research) no existing literature has proposed or assessed a systematic framework to utilize nested/repeated cross validation at computational level.

This manuscript has been organized as follows: in Section 2, we briefly introduce relevant statistical concepts and models; in Section 3, we propose the framework of nested/repeated cross-validation for model selection and feature selection; in section 4, we present a simulation study to investigate and compare the difference between using single cross-validation and nested/repeated cross-validation to build the predictive model; in Section 5, two publicly available gene expression datasets on leukemia by Golub et al. (1999) and The Cancer Genome Atlas Studies (TCGA Network 2017) on cervical cancer data are used to demonstrate the applicability of repeated/nested cross-validation method in analyzing real high dimensional data.

2. Background

A typical gene expression dataset can be presented as $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_n, y_n)\}$, where $i = 1, 2, \ldots, n$, indicating n subjects or samples. $y_i \in \{-1, 1\}$ denotes the outcome of ith subject, and the p-dimensional vector $\boldsymbol{x}_i$ defines the observed independent variables of subject i. The dataset is usually high-dimensional with many variables or features, but a relatively small sample size of n. Then a predictive model can be defined as a statistical model $\hat{f}$, an estimate of the true function f, where f is a function that maps from the gene expression data to the class of the subjects:

$$f : X \to Y \tag{1}$$

In embedded feature selection, the model optimization and variables selection are carried out simultaneously using the coefficient shrinkage or variable ranking criteria. For example, Lasso shrinks some coefficients of variables to zero, and

these variables are eliminated from the model. Usually, the statistical model $\hat{f}$ is estimated by optimization of the objective function, which is similar to empirical risk function minimization. In our work, three different embedded methods are implemented in building the predictive model and feature selection, including regularization regression via elastic net, support vector machine, and random forest.

2.1. Regression via Elastic Net Penalty

The elastic net combines the L-1 norm penalty of Lasso and L-2 norm penalty of ridge regression (Zou & Hastie 2005). Elastic net does an automatic variable selection and allows for more than n (number of observations) variables to be selected. This is because Lasso can automate the variable selection by shrinking some coefficients to zero, while ridge regression helps in regularizing the process, and the elastic net can achieve both advantages of these two methods. In classification applications, the negative binomial likelihood function is used with elastic net penalty. The model is estimated by minimizing the following objective function.

$$\arg\min_{\beta_0,\beta}\left\{\left[\frac{1}{N}\sum_{i=1}^{N} y_i\left(\beta_0+x_i^T\beta\right)-\log\left(1+e^{\beta_0+x_i^T\beta}\right)\right]+\lambda\left[\frac{(1-\alpha)\left\|\beta\right\|^2}{2}+\alpha\left\|\beta\right\|\right]\right\} \quad (2)$$

In the above expression, the first component is the loss function which penalizes the misclassification rate, and the second component is the regularization term. In (2), α and λ are called tuning parameters. The elastic net penalty is controlled by α, which bridges between lasso ($\alpha = 1$) and ridge regression ($\alpha = 0$), whereas the overall strength of the penalty is controlled by λ. The optimal value of α and λ are estimated by minimizing the above objective function. Some of the small coefficients are shrunk towards zero, and the corresponding predictors will be excluded from final model, denoted as "irrelevant" features. The remaining features are considered as "informative" features. The final model $\hat{f}$ can be used to predict the future outcome when new data is available.

2.2. Support Vector Machine

Support vector machine (SVM) creates a classifier function by constructing hyperplanes that separate different categories of the training data, and choosing the hyperplane with the maximal margin between two classes (Cortes & Vapnik 1995). Given labelled pairs $(\boldsymbol{x}_i, y_i), \boldsymbol{x}_i \in R^p, y_i \in \{1,-1\},\ i = 1,2,3,\ldots,n$, all the hyperplanes can be written as $w^T\boldsymbol{x} + b = 0$. Two parallel hyperplanes can separate two classes of data, the region between these two hyperplanes is called "margin", and the distance between these two hyperplanes is $\frac{2}{\|w\|}$. SVM aims to find the hyperplane with the maximal margin by solving the following unconstraint optimization problem:

$$\arg\min_{w,\xi_i,b}\|w\|^2 + C\sum_{i=1}^{N}\max(0, 1-y_i\left(w^T\phi(x_i)+b\right) \quad (3)$$

In the expression (3), w is the weight function that we want to minimize in order to maximize the distance $\frac{2}{\|w\|}$. C is a tuning parameter which is a trade-off between misclassification and size of margin. For example, a large C results in a relatively smaller-margin while most of samples are correctly classified, whereas a small value of C results in a relatively larger-margin but it allows more samples to be misclassified. SVM usually utilizes the kernel function, as $\phi(x_i)$ in (3), to transform the original data from input space to the feature space, which enables linearly inseparable data in low-dimension to be linearly separable in high-dimension to find the best hyperplane. One commonly used kernel function is Gaussian kernel (also called Radial Base Function), which is given by $K(x, x') = \phi(x_i')\,\phi(x_i) = \exp\left(-\gamma\|x - x'\|^2\right)$. The Gaussian kernel is used in our work.

In the optimization problem presented in expression (3), the tuning parameters for SVM with Gaussian kernel are C and γ. C is penalty parameter for misclassified samples and γ is kernel parameter. During the iterative process, the variables are ranked according to some criteria such as area under curve (AUC). The importance of each feature can be explained by the change in AUC when the feature is removed (Nguyen & de la Torre F. 2010). We determine the importance of each feature by assessing how the performance is influenced with or without having the feature. If removing a feature worsen the classification performance, the feature is considered important. The top-ranked features thus selected are the final feature subset.

2.3. Random Forest

Random forest for classification is an ensemble method that constructs multiple bootstrapped decision trees using training samples and combines all the bootstrapped trees to build the predictive model. In random forest, multiple bootstrapped dataset are generated from raw training set. Each bootstrapped dataset will be used to grow a separate decision tree. Then, all the decision trees are combined using the voting strategies (e.g. majority vote, which is the mode of all single decision trees (Breiman 2001). The detailed steps of random forest can be described as follows (1) Bootstrap samples of size $n$are drawn from data D denoted as $D_b = \{(x_{1b}, y_{1b}), \ldots, (x_{nb}, y_{nb})\}$, to create a decision tree; (2) the second step is to train the decision tree f_b based on the bootstrap samples D_b to get $\hat{f}_b$. In growing the single decision tree, m variables are randomly selected at each node of the tree. The m selected variables split the tree to achieve the minimum error; (3) the third step is to grow the tree to largest extent possible (no pruning tree); (4) repeat the previous three steps to build B bootstrapped decision trees. Then, the final ensemble model is obtained by combining the different decision trees using majority vote, denoted as $\hat{f} = mode(\hat{f}_1, \ldots, \hat{f}_B)$.

Variable importance (also known as predictor ranking) is a critical measurement in both decision trees and random forests which depends on the contribution to the tree by each predictor. The Random forest utilizes variable importance to rank the variables. Permutation techniques can be used with random forests to measure the variable importance, the details of computing the variable impor-

tance for each variable are not given here, but can be found elsewhere (Strobl, Boulesteix, Kneib, Augustin & Zeileis 2008). Features which produce large values for this score are ranked as more important than features which produce small values. The important variables are then selected by ranked variable importance.

3. Methods

In this section, we introduce the proposed method of feature selection and model selection using nested and repeated cross-validation. When building the predictive model, the most critical part for the model is to identify the optimal values of the tuning parameters to achieve the minimum test set error.

One of the widely-used techniques for model selection is K-fold cross-validation, for which the final model is chosen when the minimum cross-validation error is achieved (Hastie et al. 2009). In the K-fold cross-validation, the original training dataset is randomly divided into K subsets of equal size then the following step repeats K times: $K-1$ of the subsets are combined to build the model, and the remaining one subset is used to compute the prediction errors. The K sets of predication errors are averaged to produce the cross-validation error. To estimate the optimal value of tuning parameters, a grid of m candidate values of tuning parameters are created, and m models are built, indexed by different value of tuning parameters. The cross-validation error of each of m models is computed, and the final model is then determined by the model with minimum cross-validation error. Furthermore, the feature subset also can be determined by the model using some criteria, such as coefficients shrinkage.

As mentioned in the introduction section, the commonly used single cross-validation is not efficient in dealing with overfitting of the data in general (Varma & Simon 2006). Repeated cross-validation is an improved method by generating multiple sets of K folds. Also, the cross-validation error is calculated as the average across the repeated partitions. On the other hand, we sometimes want to select features, and use the selected features to build a predictive model. In this case, nested cross-validation can be very useful. To achieve the above goals, we propose a systematic framework of combining nested and repeated cross-validation to build the final model. In the proposed method, the cross-validation is carried out in two different layers: inner loop and outer loop. In the inner loop, the subset of features is selected as candidate features. In the outer loop, only the candidate features selected in the inner loop are carried forward to build the final model. The performance of nested and repeated cross-validation has not been extensively explored and discussed in the past mainly because of the computational costs. In this article, we show that the nested and repeated cross-validation can improve the predictive performance and selection accuracy over the traditional single cross-validation method.

3.1. Repeated Cross-Validation

In the repeated cross-validation method, instead of generating only single set of K-folds, multiple sets of K folds are generated. Also, the standard cross-validation error

$$CV\left(\theta\right)=\frac{1}{N}\sum_{k=1}^{K}\sum_{i\in F_{-k}}L\left(y_i,\hat{f}_{\theta}^{-k(i)}\left(\boldsymbol{x}_i,\theta\right)\right) \tag{4}$$

is replaced with the repeated cross-validation error

$$CV_r\left(\theta\right)=\frac{1}{RN}\sum_{r=1}^{R}\sum_{k=1}^{K}\sum_{i\in F-_k}L\left(y_i,\hat{f}_{\theta}^{-k(i)}\left(\boldsymbol{x}_i,\theta\right)\right) \tag{5}$$

Then, the value of tuning parameters is chosen as:

$$\hat{\theta}=\underset{\theta\in\{\theta_1,\ldots,\theta_m\}}{\arg\min}\ CV_r\left(\theta\right) \tag{6}$$

In the equations above, N represents the total sample size of the training data, $f_{\theta}^{-k(i)}(\cdot)$ is the function to estimate the coefficients, and $L(\cdot)$ is the loss function.

3.2. Nested cross-validation

Nested cross-validation for model selection is usually used in the case when multiple tuning parameters are estimated. In this approach, instead of generating only a single layer of K-folds, multiple layers of cross-validation loops are created. The numbers of multiple layers are determined by the numbers of tuning parameters to be estimated. If a parameter is tuned in inner loop, the value of this parameter is fixed, and assigned the fixed value in outer loop to estimate the additional tuning parameters.

In the outer layer of cross-validation, training data is partitioned into three folds (see Figure 1). Each fold will use two third of the training data (66.7% of original training) to train the model, and the remaining data (33.3% of original training) is used to estimate the CV error in the outer loop. In the inner layer of cross-validation, each fold will use two thirds of the training data generated by the outer layer ($66.7\% \times 66.7\% = 44.4\%$ of original training data), and the remaining data ($66.7\% \times 33.3\% = 22.2\%$ of original training) will be used to compute the CV error for the inner loop.

3.3. Model Selection Using Nested and Repeated Cross-Validation

We now introduce the details of our proposed method: nested and repeated cross-validation for classification model. The method has two steps: feature selection step and classification model construction step. In the proposed method, there are two layers of cross-validation, the training data is partitioned into K

folds of roughly equal size; this layer is called outer loop of cross-validation, and each dataset with Kth part removed is called inner training dataset, so there are K different inner training dataset; then, each inner training dataset is partitioned into V folds. Therefore, there are V sub-folds nested within each of the K folds. Figure 2 shows the process of our proposed method.

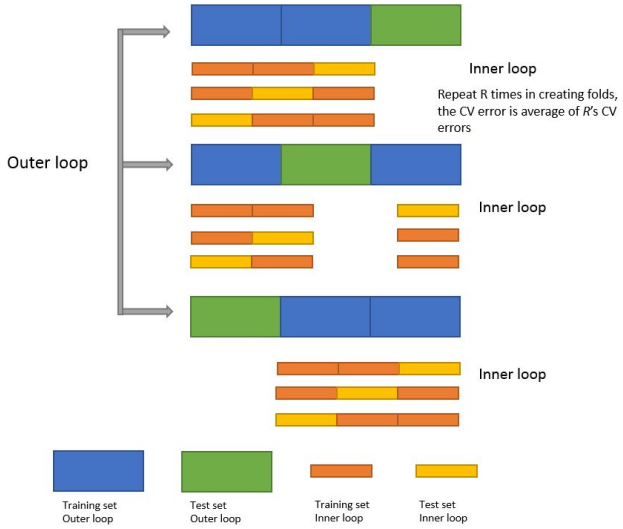

FIGURE 1: Showing the illustration of nested cross-validation, when $K, V = 3$.

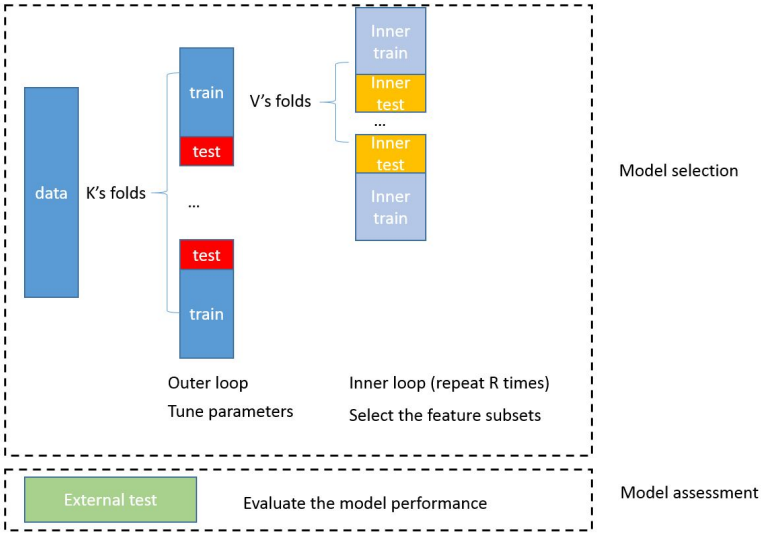

FIGURE 2: The flowchart showing the nested/repeated cross-validation in model selection.

The details of the algorithm are given as follow: The inner layer CV creates K times V models and determines the feature subset by combining all the models. The selected feature subset is then used in outer loop to estimate the tuning parameter. After the model is chosen, the model performance is evaluated using the held-out test data.

Next, an individual classifier (logistic regression via elastic net, SVM, and random forest) is used to train inner training dataset to select feature subsets using selection criteria (coefficient shrinkage method for logistic regression and variable ranking for SVM, and random forest). The classifier will select a set of informative feature subset. We then repeate cross-validation method to repeat the abovementioned step to re-partition inner training dataset to generate another V folds, R times. The individual classifier is also used to generate R different feature subsets. The final feature subset is determined using voting strategy, where any feature is selected more than 50% times ($> \frac{R}{2}$) is selected as the informative feature. After the feature subset is determined, the irrelevant features are removed and only the selected features are used in the next step. The step 2 is to build the final classification model in the outer loop. The simplified training data is used in this step, while the irrelevant features are removed, and only the selected features from step 1 are remaining. We build the final classification model using three different classification methods (logistic regression via elastic net, SVM, and random forest), the final classification model can predict the future outcome when new data is introduced, as well as evaluate the performance of selected model. The details of the proposed method is given as follows:

3.4. Step 1: Variable Selection

1. Divide the training dataset D into K folds of roughly equal size. For $k = 1$ to K, define data D^{-k} with k^{th} part removed for outer training data, and D^k with only k^{th} part remained for outer test data.

    a) Repeat the following steps R times (R is a predetermined number). Randomly divide dataset D^{-k} into V folds of roughly equal size. For $v = 1$ *to* V.

        i. Define V different data D^{-kv} with v^{th} part removed for inner training data, and D^{kv} with only v^{th} part remained for inner test data. For $m = 1$ *to* M (M is the number of grid value of the tuning parameters).

            A) Build statistical model $\hat{f}_{\theta_m} = \hat{f}(D^{-kv}; \theta_m)$.

            B) Apply $\hat{f}_{\theta_m}$ on inner test data D^{kv}, and compute the error using the loss function in inner test set.

$$Err_{\theta_m} = \sum_{i \in D^{-kv}} L\left(y_i, \hat{f}\left(D^{-kv}; \theta_m\right)\right)$$

        ii. Compute the V-fold cross-validation error for each m, therefore, there are m different CV errors. N_v is the number of samples in

inner loop for k^{th} part.

$$CV\left(\hat{f};\theta_m\right) = \frac{1}{N_v}\sum_{v=1}^{V}\sum_{i\in D^{-kv}} L\left(y_i, \hat{f}\left(D^{-kv};\theta_m\right)\right)$$

iii. By repeating the above step Rtimes, we derive CV error for the repeated cross-validation procedure for each m. N_v is the number of samples in inner loop for k^{th} part.

$$CV_R\left(\hat{f};\theta_m\right) = \frac{1}{N_v R}\sum_{r=1}^{R}\sum_{v=1}^{V}\sum_{i\in D^{-kv}} L\left(y_i, \hat{f}\left(D^{-kv};\theta_m\right)\right)$$

b) Determine the optimal value of tuning parameter from all possible m

$$\hat{\theta}_m = \underset{\theta\in\{\theta_1,\theta_m\}}{\arg\min}\, CV_R\left(\hat{f};\theta_m\right)$$

c) The optimal values of tuning parameters are then fixed in the objective function, and the objective function is minimized using gradient descent algorithm (Zhang 2004, Shalev-Shwartz, Singer, Srebro & Cotter 2011). When the final model is then chosen, and feature subset is determined by variable ranking method or coefficient shrinkage methods. Let $s(\cdot)$ be an indicator function, represented by:

$$s(x) = \begin{cases} 1 & \text{if } p_i \text{ is selected by the final model } i = 1, 2, \ldots, p \\ 0 & \text{if } p_i \text{ is not selected} \end{cases}$$

Then, the feature subset can be denoted as: $FS = \{s(p_1), s(p_2), \ldots, s(p_p)\}$, where for each of k-fold, we derive a "winner" feature subset, denoted as $FS_k = \{s(p_1), s(p_2), \ldots, s(p_p)\}$

2. For these K "winner" feature subsets, we compute the number of times that each feature is selected. Then, the final feature subset is defined as: $FS_{final} = \{fs(p_1), fs(p_2), \ldots, fs(p_p)\}$, where $fs(\cdot)$ is an indicator function, indicating whether the p^{th} feature is selected, and represented by

$$fs(x) = \begin{cases} 1 & \text{if } p_i \text{ is selected greater or equal to } \frac{K}{2} \text{ times, } i = 1, 2, \ldots, p \\ 0 & \text{if } p_i \text{ is selected less than } \frac{K}{2} \text{ times } i = 1, 2, \ldots, p \end{cases}$$

3. The previous step creates a subset of p' selected variables, where p'^{th} is the number of selected variables. The training data is subsetted for these selected variables for model building.

3.5. Step 2: Classification Model Building

1. Reduce the training dataset D to D', where $D' = (D; p')$. Only the variables selected in Step 1 are kept in D'

2. Using same fold that was generated in step 1. For $k = 1$ to K

   a) Define data $D'^{(-k)}$ with k^{th} part removed for training, and $D'^{(k)}$ that k^{th} part remained for test data. Repeat the following step R times (R is predetermined scaler, representing the repeat times). For $m = 1$ to M (M is the numbers of grid value of tuning parameters)

      i. Build statistical model $\hat{f}_{\theta_m} = \hat{f}(D'^{(-k)}; \theta_m)$

      ii. Apply $\hat{f}_{\theta_m}$ on inner test data $D'^{(k)}$, and compute the error using the loss function for each m.

$$Err_{\theta_m} = L\left(y_i, \hat{f}(D'^{(-k)}; \theta_m)\right)$$

   b) Compute the K-fold cross-validation error for each of the M values of the tuning parameters

$$CV\left(\hat{f}; \theta_m\right) = \frac{1}{N}\sum_{k=1}^{K}\sum_{i \in D'(-k)} L\left(y_i, \hat{f}(D'^{(-k)}; \theta_m)\right)$$

   c) Derive CV error for the repeated cross-validation procedure

$$CV_R\left(\hat{f}; \theta_m\right) = \frac{1}{KR}\sum_{r=1}^{R}\sum_{k=1}^{K}\sum_{i \in D'(-k)} L\left(y_i, \hat{f}(D'^{(-k)}; \theta_m)\right)$$

3. Determine the optimal value of tuning parameter from all possible m points

$$\hat{\theta} = \underset{\theta \in \{\theta_1, \ldots, \theta_m\}}{\arg\min}\, CV_R\left(\hat{f}; \theta\right)$$

4. The optimal value of tuning parameters is then fixed in the objective function, and the objective function is minimized by some optimization methods, such as gradient descent methods, in order to obtain the final model.

To sum up, the method to build and select the predictive model using repeated and nested cross-validation has more steps than standard single step cross-validation. The complete process is illustrated in Figure 2. The inner loop is created to select a candidate subset of features. While training the model in the inner loop, the V-folds are generated and repeated R times to alleviate the randomness of generation of each fold. This will reduce the variance. The outer loop will use subset of selected variables to build the final classification model. A simulation study has been presented evaluating the efficiency and comparing its performance with other standard methods. Also, the application of this approach has been presented with two real datasets.

4. Simulation Study

Suppose Y_i is a binary disease outcome, representing the normal cell or cancer cell for the i^{th} sample and suppose X_i is p-vector that represents the gene expression for the i^{th} sample. According to the nature of genetic pathology, there were several characteristics we needed to consider in our simulation study: (1) some genes are critical to the disease outcome, and those genes are differentially expressed between cancerous and non-cancerous cells; (2) a few genes may work as a group to influence the disease outcome and those genes are mutually correlated (Hira & Gillies 2015). We carry out a cross-sectional simulation study considering the above essential biological settings. We apply the aforementioned three classification and feature selection methods in the simulated data to assess the performance of the proposed methods and compare to the standard cross-validation method.

4.1. Generating the Predictors

We simulated our microarray data set with a fixed number of ($n = 100$) samples. We consider a small pool ($p = 2000$) and a large pool ($p = 5000$) of features. The simulated design matrix X consists of three groups of informative features and remaining are irrelevant features. The first group is the most important group, which has 1% of all p predictors. The numbers of the features of the three important feature groups are 1%, 2%, and 2% of all p predictors, respectively. We use three different strengths of correlations coefficient ($\rho = 0.3$, 0.5 and 0.8) for the genes (predictors) within the group but assume that the predictors between different groups are independent. Thus, we define that X_g, $g = 1, 2, 3$, indicating the gene expression for the three groups of important genes. The data is simulated from a multivariate normal distribution:

$$X_g \sim MVN\left(\mu_g, \Sigma_g\right), \quad g = 1,\ 2,\ 3 \tag{7}$$

where, $\mu_g = 0$, and $\Sigma_g = T^{\frac{1}{2}}\Gamma T^{\frac{1}{2}}$, where

$$\Gamma = \begin{bmatrix} 1 & \cdots & \rho \\ \vdots & \ddots & \vdots \\ \rho & \cdots & 1 \end{bmatrix}, \text{ and } T = \begin{bmatrix} \sigma^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma^2 \end{bmatrix},$$

ρ is the pre-determined correlation coefficient. The remaining 95% predictors are simulated from the standard normal distribution $X_i \sim N\left(0, 1\right)$, $i = (0.05p + 1), \ldots, p$. Then, we combine the X_g and X_i to create our final design matrix X. In reality, the structure of noise terms could be very complex. They can be mutually correlated and even correlated with the informative features. To investigate these complicated scenarios, the more complicated design is required. We do not address these situations in our simulation study.

4.2. Generating the Outcomes

We assume that Y follows a logistic regression with $Logit\left[P\left(Y_i = 1 \mid X_i, X_{true}\right)\right] = X_{true}\beta_{true}$, where X_{true} indicates a subset vector of "informative" variables of X_i. Therefore, the outcome Y_i is simulated from a Bernoulli distribution, where $Y_i \sim Bern(P_i)$. P_i is the $Pr(subject\ i\ has\ disease)$, where $P_i = \Pr\left(Y_i = 1 | X_i\right) = \frac{\exp\left(Z_i\right)}{1+\exp\left(Z_i\right)}$. The model of $Z_i = X_i\beta$ is used to derive the value of Z_i. X_i is the i^{th} vector of the design matrix as defined in the previous section, β is the vector of coefficients. The value of β is set to 5 for important feature group, 3 for secondary feature group, 2 for third feature group, and 0 for all the noise term, denoted as $\epsilon_i \sim N(0, 0.01)$.

In the simulation study, we consider the following six scenarios by considering the number of pool of variables (small and large), and within-group correlation (low, medium, and high). The final simulated data looks like as $D = \left\{\left(x_1, y_1\right), \left(x_2, y_2\right), \ldots, \left(x_n, y_n\right)\right\}, i = 1, 2, \ldots, n.$

After simulating the data for six cross-sectional scenarios, we apply three different methods to build the predictive model, including regularization methods with elastic net penalty, support vector machine, and random forest. The simulation study will investigate the following questions:

1. Whether applying repeated and nested cross-validation method improves the predictive performance than applying single cross-validation only.
2. Comparative study among three different methods to build the predictive model
3. Comparative study among six different data structures and correlation settings.

Table 1 presents the summary of AUC for three different predictive modelling methods: regularization methods with elastic net penalty, SVM, and random forest. Method 1 refers to the AUC for standard CV method. Method 2 refers to the AUC when method of repeated and nested CV is used. The last line of each scenario, labelled as "True" represents only true simulated variables are used in classification model building with standard cross-validation method. We consider the six different scenarios to investigate the performance when repeated and nested CV is used. Figure 3 presents the results using a box plot.

In the simulation study, we investigated the six scenarios to compare the model performance of building predictive model using standard cross-validation and using repeated and nested cross-validation. Table 1 summarizes the area under ROC curve (AUC) for the simulation study. We define the building predictive model using standard cross-validation as Method 1, whereas using repeated and nested cross-validation as Method 2. Table 1 shows that the AUC from Method 2 are consistently higher than AUC from Method 1, for three different statistical learning Methods (regularization Method, SVM, RF). This indicates that when Method 2 is used, the generalization error (test error) is lower than Method 1. Therefore, when Method 2 is applied, it provides a better estimated model than Method 1 is

used. In Figure 1, the gray bar represents the Method 2 and white bar represents the Method 1. The mean of AUC for Method 2 is consistently higher than AUC for Method 1.

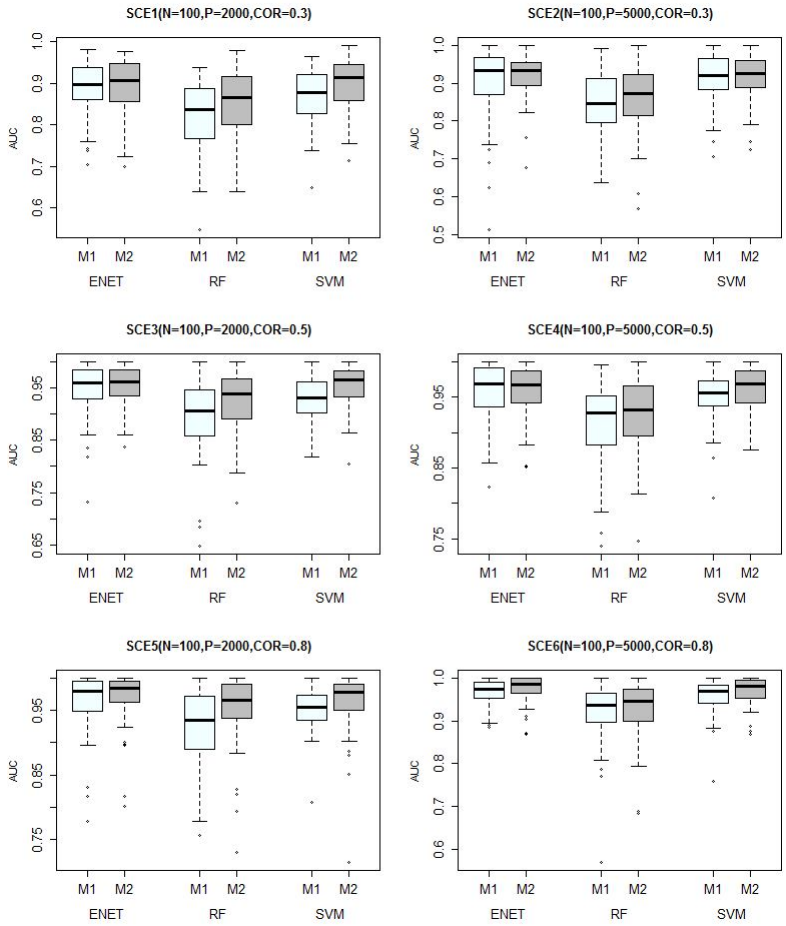

FIGURE 3: Boxplot of AUC comparing the simulation result. The white bar represents Method 1 (M1) and gray bar represents Method 2 (M2). M1 refers to the applying of standard cross-validation, whereas M2 refers to the applying of proposed method. The black line of each box is the mean of AUC. The six side-by-side box is the comparison of the AUC between using different models, including regularization methods via elastic net (ENET), SVM, and random forest(RF).

Table 1 also enables the comparative study for different statistical modelling strategies to build the predictive model. The overall AUC is the one of such criteria to compare among regularization Methods. The simulation study shows the regularization Methods with elastic net has the best prediction performance than other two modelling strategies. However, since the model performance is data-driven, the evidence is weak, and it can only justify that regularization methods

with elastic net has better predictive results for this specific simulated dataset. As well known, the SVM and random forest perform well when data is non-linear, thus, these two methods can be more appropriate when using in the real data having nonlinear trend.

Table 1: Summary of area under curve (AUC) for three feature selection methods for six different simulation scenarios.

	Elastic net	Support Vector Machine	Random Forest
Scenario 1, $n = 100$, $p = 2000$, correlation = 0.3			
Method 1	0.8856	0.8646	0.8215
Method 2	0.8930	0.8968	0.8532
True	0.9688	0.9767	0.9566
Scenario 2, $n = 100$, $p = 5000$, correlation = 0.3			
Method 1	0.9029	0.9151	0.8432
Method 2	0.9197	0.9153	0.8612
True	0.9736	0.9777	0.9518
Scenario 3, $n = 100$, $p = 2000$, correlation = 0.5			
Method 1	0.8823	0.8648	0.7983
Method 2	0.8823	0.8802	0.8381
TRUE	0.9612	0.9757	0.9483
Scenario 4, $n = 100$, $p = 5000$, correlation = 0.5			
Method 1	0.8900	0.8838	0.7936
Method 2	0.8905	0.8916	0.8457
TRUE	0.9719	0.9778	0.9520
Scenario 5, $n = 100$, $p = 2000$, correlation = 0.8			
Method 1	0.8922	0.9017	0.8570
Method 2	0.9171	0.9205	0.8840
TRUE	0.9774	0.9793	0.9537
Scenario 6, $n = 100$, $p = 5000$, correlation = 0.8			
Method 1	0.9324	0.9345	0.8877
Method 2	0.9422	0.9403	0.8989
TRUE	0.9874	0.9878	0.9648

The computation time of the proposed method, however, is longer than the standard cross validation method. Table 2 2 shows the comparison of the computation times for the proposed and standard cross validation methods for each of the elastic net, SVM and random forest using the simulated data. With the six-simulation scenarios, the proposed method took around 10-15 times more time than the standard method. The method with elastic net and random forest took around 10-12 times and SVM took around 13-15 times more time. Elastic net took shorter time as compared to Support Vector Machine and Random Forest. The computation time also depends on the dimension of the data as shown by scenarios 4-6 vs scenarios 1-3. As the dimension increases the computational time also increases. Overall, the better accuracy can be achieved at the cost of longer computation time. The computational burden can be minimized by using parallel and cloud computing.

TABLE 2: Table showing the comparison of the computation times (in seconds) between the standard and proposed method.

	Elastic net	Support Vector Machine	Random Forest
Scenario 1, $n = 100$, $p = 1000$, correlation $= 0.3$			
Method 1	2.225	6.299	19.37
Method 2	25.183	82.612	204.812
Scenario 2, $n = 100$, $p = 1000$, correlation $= 0.5$			
Method 1	1.718	4.809	16.076
Method 2	19.655	63.661	171.104
Scenario 3, $n = 100$, $p = 1000$, correlation $= 0.8$			
Method 1	1.952	5.385	16.733
Method 2	22.744	71.355	177.236
Scenario 4, $n = 100$, $p = 5000$, correlation $= 0.3$			
Method 1	3.151	24.001	89.263
Method 2	35.605	327.013	942.643
Scenario 5, $n = 100$, $p = 5000$, correlation $= 0.5$			
Method 1	3.555	25.364	91.98
Method 2	35.899	338.954	963.974
Scenario 6, $n = 100$, $p = 5000$, correlation $= 0.8$			
Method 1	3.174	23.584	83.389
Method 2	35.236	321.502	908.563

5. Application to Real Life Data

5.1. Application to Leukemia Gene Expression Data

Two important approaches of data analysis of microarray data include grouping the genes to discover broad patterns of biological process, and selecting important genes that are associated with disease. We use the gene expression dataset from leukemia and cervical cancer to investigate the performance of our proposed method.

The leukemia data, presented in Golub et al. (1999), consists of 47 patients with acute lymphoblastic leukemia (ALL) and 25 patients with acute myeloid leukemia (AML). Each of the 72 patients had a bone marrow samples obtained at the time of diagnosis. Furthermore, the observations have been assayed with Affymetrix Hgu6800 chips, resulting in 7129 gene expressions (Affymetrix probes). The Golub data set is possibly the most widely studied and cited microarray data set [6]. In this real data study, we also implement two different methods: Method 1 and Method 2 as mentioned above. The models are trained using training set (38 samples), the AUC and misclassification rate are calculated by using held-out test set (34 samples).

Figure 4 shows the comparison of AUC between two Methods using three statistical modelling approaches. The blue line is ROC for Method 1 whereas the red line is ROC for Method 2. The AUC values are shown at the bottom of right

corner. We can see that the AUC from Method 2 is higher than the AUC from Method 1, which indicates that Method 2 has better prediction performance than Method 1.

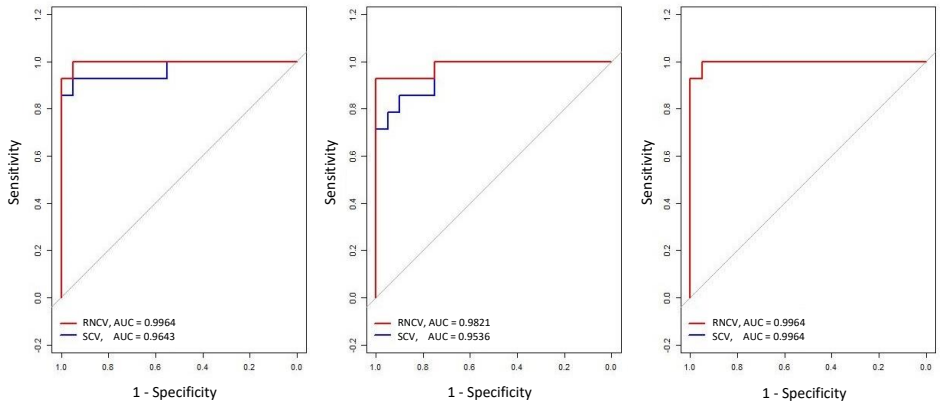

FIGURE 4: Comparison of AUC between two methods using three statistical modelling. The red line refers to the proposed repeated/nested cross-validation, whereas the blue line refers to standard cross-validation. In all three methods, the AUC from the proposed method has uniformly better than standard way.

Besides looking at ROC and AUC, the misclassification rate is also an important criteria to assess the model performance. The misclassification rate is computed as: $misclass.rate = \frac{FP+FN}{TP+TN+FP+FN}$. The terminologies are described in the table below:

TABLE 3: Cross-tabulation of true and predicted classification scenarios.

		Predicted	
		Positive	Negative
Actual	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

A total of 34 bone marrow test samples were used to compute the misclassification rate. Among the 34 samples, 20 samples are ALL, defined as positive class, and 14 samples are AML, defined as negative class. The predictive performance measurements can be estimated from Table 3, for example, the true positive (TP) can be explained as the predictive class is ALL and actual labelled class is also ALL. The misclassification rate then is calculated when the predictive performance measurements are known.

Table 4 compares AUC result between using single cross-validation (Method 1) in building the predictive model and using repeated and nested cross-validation (Method 2) in building the predictive model for leukemia cancer gene expression data. For both methods, three different classifiers are implemented into the framework. For Method 1, the misclassification rates for generalized linear model with elastic net penalty, SVM, and random forest are 23.5%, 11.8%, and 23.5%, respectively. In contrast, for Method 2, the misclassification rates for generalized

linear model with elastic net penalty, SVM, and random forest are 14.7%, 5.9%, and 11.8%, respectively. Therefore, to achieve more accurate prediction accuracy when new data is introduced, the predictive models built using repeated and nested cross-validation would be better.

Table 4: Misclassification rate for three different methods.

		TP	FN	FP	TN	Misclassification rate
Enet	Method 1	20	8	0	6	23.50%
	Method 2	20	5	0	9	14.70%
SVM	Method 1	20	4	0	10	11.80%
	Method 2	19	1	1	13	5.90%
RF	Method 1	20	8	0	6	23.50%
	Method 2	20	4	0	10	11.80%

5.2. Application to Cervical Cancer Gene Expression Data

Our second example is on The Cancer Genome Atlas (TCGA) studies on cervical cancer (TCGA Network 2017). Cervical cancer consists of primarily of two different types: squamous cell carcinoma and adenocarcinoma. A limited number also consists of both squamous and glandular cancer cells, termed as adenosquamous carcinoma. The data consists of 178 samples including 144 squamous cell carcinoma, 31 adenocarcinoma and 3 adenosquamous carcinoma. We excluded the 3 adenosquamous samples for this example. There were 20,533 genes assayed using RNAseq technology on those 175 subjects. After quality control and preprocessing 19,037 genes were left for the analyses. The main purpose of the use of this data was to assess the performance of our method rather than substantial analyses of the data. TCGA study has shown that the gene expression pattern is very different between the squamous and adenocarcinoma histology. In our implementation of the method, we aim to classify the two histologic subtypes based on the informative subset of gene expression data. We splitted the data into two pieces and 80% of the samples (140 samples) were used as training data and 20% (35 samples) were utilized to assess the model.

Table 5 shows the summary of the AUC and the accuracy/misclassification of classifying the two types of cervical cancer: squamous and adenosquamous carcinoma. The results are consistent with first example on leukemia gene expression data. The AUCs are higher with the proposed repeated and nested cross validation method as compared to the standard cross validation method. Similarly, the mis-sclassification rate is lower with the proposed method. The results demonstrate that the repeated and nested cross validation method performs better as compared to the standard k-fold cross validation method.

In addition, for both examples, we carried out the differential expression (DE) analyses of the genes between the two groups to select the genes prior to fitting our proposed method. Then, we used only those genes in our method in order to investigate how that would affect the results. The results showed that the relative efficiency of the Method 1 and Method 2 were exactly the same to that of using

all genes in both leukemia and cervical cancer examples. This was because our method selected only the subsets of DE genes when using all the genes in the model.

Table 5: Showing the AUC and misclassification rate for three different methods.

		AUC	TP	FN	FP	TN	Misclassification rate
Enet	Method 1	91.05	22	7	0	6	20.00
	Method 2	94.11	22	5	0	8	14.29
SVM	Method 1	85.47	22	6	0	7	17.14
	Method 2	88.23	21	4	0	10	11.43
RF	Method 1	87.25	22	8	0	5	22.86
	Method 2	91.44	22	6	0	9	16.22

6. Discussion

In this article, we explored a more robust cross validation method for variable selection and outcome classification. We also demonstrated its application using two gene expression datasets. The method can be applied to any type of high dimensional data where the concern is to classify the outcomes using a few important variables. The proposed method applies a repeated and nested cross-validation framework to build a predictive model and select the subsets of features for classification. The proposed approach completes the two important tasks: variable selection and outcome prediction. The outcome of the proposed method can be utilized further where the research question is concerned about predicting the outcome class using only a few important biomarkers.

There are several works done for the model selection and classification (Hernán dez & Correa 2009, Salazar 2012). Our proposed method uses a combination of repeated and nested cross-validation technique instead of standard cross-validation method. In our method, double layers of cross-validation are created. In the inner loop, we perform variable selection and determine the subset of informative variables, then, the subset of informative variables is used in the outer loop to estimate the parameters. After the parameters are estimated, the final model is then chosen with the cross-validation error minimized.

In the simulation study, we present different scenarios under the cross-sectional biological settings. The simulated dataset is used to build predictive models using three different statistical methods with two different cross-validation techniques including single cross-validation and repeated nested cross-validation. From the results of the simulation study, we have shown that our proposed method can provide better prediction accuracy in all three different statistical modeling approaches.

In the application, we used two gene expression datasets, the leukemia dataset from (Golub et al. 1999) and the TCGA cervical cancer. We used three different statistical modeling approaches including generalized linear model via elastic net penalty, SVM, and random forest for this classification task. We found that our

proposed method reduces the generalization error compared to the single cross-validation method.

The proposed method also has some limitations. Rather than using the normal K fold cross-validation for model selection, the nested cross-validation requires V folds nested in K fold, thus, the total $K \times V$ folds are generated for selecting the features and estimating the tuning parameters. Therefore, the computation time is significantly increased. There is trade-off between the accuracy and computational cost. However, with the development of modern computing facilities, the computational burden can be minimized using sophisticated technologies such as the parallel and cloud computing.

Our proposed method can be extended in several ways. (1) the result of feature selection from the predictive model determines a set of informative genes. When other critical clinical characteristics are collected, an integrative model can be created by combining the genes and those clinical covariates. (2) the cross-validation is a commonly used technique for model selection and model assessment. In our method, we use nested and repeated cross-validation to select the parameters and to perform model selection. It is also possible to extend the nested repeated cross-validation in model assessment and to estimate variation of the prediction accuracy.

In summary, we describe a framework for using nested and repeated cross-validation to perform feature selection and building a predictive classification model for high dimensional data. The proposed method is able to provide an improved prediction, and is also able to extract a subset of informative features from the pool of thousands of features.

Acknowledgements

We thank Golub et al. (1999) for the use of the data from leukemia studies. We also thank The Cancer Genome Atlas (TCGA) network for the cervical cancer gene expression data.

[Received: May 2019 — Accepted: December 2019]

References

Braga-Neto, U. M. & Dougherty, E. R. (2004), 'Is cross-validation valid for small-sample microarray classification?', *Bioinformatics* **20**(3), 374–380.

Breiman, L. (2001), 'Random Forest', *Machine Learning* **5**(32).

Cortes, C. & Vapnik, V. (1995), 'Support-Vector Networks', *Machine Learning* **45**(1), 5–32.

Dash, M. & Liu, H. (1997), 'Feature Selection for Classification', *Intell. Data Anal* **1**(3), 131–156.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loa, M. L., Downing, J. R., Caligiuri, M. A., Bloomfield, C. D. & Lander, E. S. (1999), 'Molecular classification of cancer: class discovery and class prediction by gene expression monitoring', *Science* **286**(5439), 531–537.

Guyon, I. (2006), *Feature extraction: foundations and applications*, Springer-Verlag, Berlin.

Hastie, T., Tibshirani, R. & H., F. J. (2009), *The elements of statistical learning: data mining, inference, and prediction*, 2nd edn, Springer, New York.

Hernán dez, F. & Correa, J. C. (2009), 'Comparison for three classification techniques', *Revista Colo mbiana de Estadística* **32**(2), 247–265.

Hira, Z. M. & Gillies, D. F. (2015), 'A review of feature selection and feature extraction methods applied on microarray data', *Advances in Bioinformatics* **13**.

Krstajic, D., Buturovic, L. J., Leahy, D. E. & Thomas, S. (2014), 'Cross-validation pitfalls when selecting and assessing regression and classification models', *Journal of cheminformatics* **6**(1), 10.

Kumar, V. & Minz, S. (2014), 'Feature Selection: A Literature Review', *Smart Computing Review* **4**(3), 211–229.

Lu, Y. & Han, J. W. (2003), 'Cancer classification using gene expression data', *Information Systems* **28**(4), 243–268.

Nguyen, M. H. & de la Torre F. (2010), 'Optimal feature selection for support vector machines', *Pattern Recognition* **43**(3), 584–591.

Pomeroy, S. L., Tamayo, P., Gaasenbeek, M., Sturla, L. M., Angelo, M., McLaughlin, M. E., Kim, J. Y. H., Goumnerova, L. C., Black, P. M., Lau, C. et al. (2002), 'Prediction of central nervous system embryonal tumour outcome based on gene expression', *Nature* **415**(6870), 436–442.

Saeys, Y., Inza, I. & Larranaga, P. (2007), 'A review of feature selection techniques in bioinformatics.', *Bioinformatics* **23**(19), 2507–2517.

Salazar, D. A. (2012), 'Comparison between SVM and Logistic Regression: Which One is Better to Discriminate? ', *Revista Colombiana de Estadística* **35**(2), 223–237.

Shalev-Shwartz, S., Singer, Y., Srebro, N. & Cotter, A. (2011), 'Pegasos: primal estimated sub-gradient solver for SVM', *Mathematical Programming* **127**(1), 3–30.

Stone, M. (1974), 'Cross-Validatory Choice and Assessment of Statistical Predictions', *Journal of the Royal Statistical Society* **36**(2), 111–147.

Strobl, C., Boulesteix, A.-L., Kneib, T., Augustin, T. & Zeileis, A. (2008), 'Conditional variable importance for random forests', *BMC bioinformatics* **9**(1), 307.

TCGA Network (2017), 'Integrated genomic and molecular characterization of cervical cancer', *Nature* **543**(7645), 378.

Trevino, V., Falciani, F. & Barrera-Saldana, H. A. (2007), 'DNA microarrays: a powerful genomic tool for biomedical and clinical research', *Molecular Medicine* **13**(9), 527–541.

Van't Veer, L. J., Dai, H., Van De Vijver, M. J., He, Y. D., Hart, A. A., Mao, M., Peterse, H. L., Van Der Kooy, K., Marton, M. J., Witteveen, A. T. et al. (2002), 'Gene expression profiling predicts clinical outcome of breast cancer', *nature* **415**(6871), 530.

Varma, S. & Simon, R. (2006), 'Bias in error estimation when using cross-validation for model selection', *BMC bioinformatics* **7**(1), 91.

Whelan, R., Watts, R., Orr, C. A., Althoff, R., Artiges, E., Banaschewski, T., Barker, G. J., Bokde, A. L. W., Büchel, C., Carvalho, F. M. et al. (2014), 'Neuropsychosocial profiles of current and future adolescent alcohol misusers', *Nature* **512**(7513), 185–189.

Zhang, L., Zhou, W., Velculescu, V. E., Kern, S. E., Hruban, R. H., Hamilton, S. R., Vogelstein, B. & Kinzler, K. W. (1997), 'Gene expression profiles in normal and cancer cells', *Science* **276**(5316), 1268–1272.

Zhang, T. (2004), Solving large scale linear prediction problems using stochastic gradient descent algorithms, *in* 'Proceedings of the twenty-first international conference on Machine learning', ACM, p. 116.

Zou, H. & Hastie, T. (2005), 'Regularization and variable selection via the elastic net ', *Journal of the Royal Statistical Society. Series B-Statistical Methodology* **67**, 301–320.