BIOINFORMATICS

Vol. 00 no. 00 2007
Pages 1–3

WilcoxCV: an R package for fast variable selection in cross-validation

Anne-Laure Boulesteix [a]*

[a]Sylvia Lawry Centre for Multiple Sclerosis Research, Hohenlindenerstr. 1, D-81677 Munich, Germany.

ABSTRACT

Summary: In the last few years, numerous methods have been proposed for microarray-based class prediction. Although many of them have been designed especially for the case $n \ll p$ (much more variables than observations), preliminary variable selection is almost always necessary when the number of genes reaches several tens of thousands, as usual in recent data sets. In the two-class setting, the Wilcoxon rank sum test statistic is, with the t-statistic, one of the standard approaches for variable selection. It is well-known that the variable selection step must be seen as a part of classifier construction and, as such, be performed based on training data only. When classifier accuracy is evaluated via cross-validation or Monte-Carlo cross-validation, it means that we have to perform p Wilcoxon or t-tests for each iteration, which becomes a daunting task for increasing p. As a consequence, many authors often perform variable selection only once using all the available data, which can induce a dramatic underestimation of error rate and thus lead to misleadingly reporting predictive power. We propose a very fast implementation of variable selection based on the Wilcoxon test for use in cross-validation and Monte Carlo cross-validation (also known as random splitting into learning and test sets). This implementation is based on a simple mathematical formula using only the ranks calculated from the original data set.

Availability: Our method is implemented in the freely available `R` package `WilcoxCV` which can be downloaded from the Comprehensive R Archive Network at
http://cran.r-project.org/src/contrib/Descriptions/WilcoxCV.html.

Contact: boulesteix@slcmsr.org

1 INTRODUCTION

Many applied and methodological articles have been devoted to class prediction based on high-dimensional microarray data with applications to, e.g., molecular cancer diagnosis or prediction of response to therapy. In this context, it is common practice to perform univariate variable selection before constructing a classifier, even if the chosen classification method can handle a large number of predictors. In binary classification, it is usual to rank genes according to the p-value obtained in, e.g., the t-test for two independent samples and related methods or the Wilcoxon rank sum test, also known as the Mann-Whitney test (Dettling and Bühlmann, 2003; Boulesteix and Tutz, 2006). The genes with the smallest p-values are then selected and used for classifier construction. In contrast to the t-test, the Wilcoxon rank sum test is robust against outliers, which are frequent in microarray data, and does not require normal distribution of the expression levels within both classes. This is an important advantage, since normality of gene expression data is often questionable, even after normalization. Wilcoxon-based variable selection is reported to perform very well in one of the most extensive comparison studies on microarray-based classification (Lee *et al.*, 2005).

The performance of classification methods is commonly evaluated by cross-validation (CV) or Monte-Carlo cross-validation (MCCV). In m-fold cross-validation, the n observations are divided into m (approximately) equally sized groups. In the k-th cross-validation iteration, the k-th group is considered as test data set, whereas the remaining $m-1$ groups form the learning set which is used for classifier construction. This classifier is then used to predict the observations from group k. After the m iterations, the error rate is estimated as the proportion of misclassified observations. An important special case is leave-one-out cross-validation (LOOCV), where $m = n$. Monte-Carlo cross-validation (also denoted as subsampling or random splitting in the literature) also consists of several iterations in which the data set is split into learning and test sets. In contrast to cross-validation, the test sets are not chosen to form a partition of the whole data set but drawn randomly (without replacement) from the n observations at each iteration. The number of iterations N_{iter} is fixed by the user and can be as high as computationally feasible, leading to a more robust estimation than cross-validation. The size ratio between learning and test data sets is also fixed by the user. Usual choices are, e.g., 2:1, 4:1 or 9:1. Repeated cross-validation is another robust procedure (Braga-Neto and Dougherty, 2004) which consists of averaging the results obtained in cross-validation for different partitions. Braga-Neto and Dougherty (2004) and Molinaro *et al.* (2005) review and compare procedures for estimating the error rate of a classifier, including those mentioned in the present article and other like bootstrap sampling.

Procedures such as CV and MCCV are commonly applied for both estimation and optimization purposes. When used for estimation, the goal of CV and MCCV is to evaluate the performance of the considered classifier on independent data, which is a major topic in all medical articles on microarray-based prediction. In the context of optimization, CV and MCCV aim at selecting the best combination of method parameters based on a learning set. These parameters are then used to predict observations from the test set. Method parameters may include, e.g., the number of components in PLS (Boulesteix and Strimmer, 2007) and other dimension reduction methods (Dai *et al.*, 2006) or the penalty parameter in penalized logistic regression (Zhu and Hastie, 2004). When reporting the accuracy of a classification method, it is particularly important to

*to whom correspondence should be addressed: boulesteix@slcmsr.org

	$n = 30$	$n = 50$	$n = 100$
$LOOCV$	$72/\mathit{1.5}$	$130/\mathit{2.5}$	$270/\mathit{5.9}$
$MCCV\ 9:1$ $N_{iter} = 100$	$250/\mathit{4.7}$	$250/\mathit{5.3}$	$270/\mathit{8.3}$

Table 1. Time (in seconds) needed by the standard approach i) (normal font) and our novel algorithm ii) (italic) as output by the function `system.time`.

perform such a CV-based optimization step, denoted as inner loop by Statnikov *et al.* (2005).

In many articles using a cross-validation procedure (either for error rate estimation or for parameter optimization), it is unclear whether and when preliminary variable selection was performed, although bias due to too early variable selection are well documented (Ambroise and, 2002). When LOOCV is used for error estimation, selecting variables using all n observations instead of considering variable selection as a part of classifier construction leads to downwardly biased estimation. Apparently good performing classifiers maybe produced even when predictors are not associated with class membership, yielding 'noise discovery' (Ioannidis, 2005). When LOOCV is used to determine the optimal parameter value of a given method, for instance the number of components in PLS dimension reduction (Dai *et al.*, 2006; Boulesteix and Strimmer, 2007), performing variable selection with all available observations may favor sparse models.

We argue that computational expense is the main reason for variable selection to be often (spuriously) performed only once using all available observations. We propose an implementation of variable selection based on the Wilcoxon rank sum test in the context of CV and MCCV which solves this problem by using a simple mathematical formula. It outputs the Wilcoxon test statistics for all CV or MCCV iterations simultaneously in much less time than if the Wilcoxon test were applied successively in all iterations.

2 IMPLEMENTATION

Let us consider a sample $(x_i, y_i)_{i=1,\dots,n}$, where y_i denotes the binary class membership ($y_i = 0, 1$) and x_i the expression level of observation i for the considered gene. For simplicity, we omit the index g ($g = 1, \dots, p$) of the gene. Let R_i denote the rank of observation i. The Wilcoxon rank sum test tests the equality of the medians in two independent samples (here, the samples defined by $y_i = 0$ and $y_i = 1$). The test statistic is given as $W = \sum_{i:y_i=0} R_i$, which is the sum of the ranks of observations from class $y_i = 0$. The p-value of the test is derived from the exact null-distribution of W (for very small samples) or from the asymptotic result

$$\frac{W - n_0(n+1)/2}{\sqrt{n_0 n_1 (n+1)/12}} \sim_{H_0} \mathcal{N}(0, 1), \qquad (1)$$

where n_0 and n_1 are the numbers of observations with $y_i = 0$ and $y_i = 1$, respectively. In CV or MCCV, we denote as T_k ($k = 1, \dots, N_{iter}$) the set of the n_{T_k} observations included in the test set for the k-th iteration. For example, we have $N_{iter} = m$, $\cup_{k=1}^{m} T_k = \{1, \dots, n\}$ and $T_{k_1} \cap T_{k_2} = \emptyset\ \forall k_1 \neq k_2$ in m-fold cross-validation. In the special case of leave-one-out cross-validation, T_k is defined as $T_k = \{k\}$ and $N_{iter} = n$. Let W_k denote the Wilcoxon rank sum test statistic obtained based on the sample $(x_i, y_i)_{i \notin T_k}$ including all observations except those from T_k. We derive a new simple formula allowing to compute W_k, $k = 1, \dots, N_{iter}$ simultaneously. Let $R_{i,k}$ denote the rank of observation i in the k-th iteration, i.e. in the sample $(x_i, y_i)_{i \notin T_k}$, with the convention $R_{i,k} = 0 = (R_i - R_i)$ if $i \in T_k$. For $i \notin T_k$, we have

$$R_{i,k} = R_i - \sum_{j \in T_k} I(R_j < R_i). \qquad (2)$$

We obtain

$$W_k = \sum_{i:y_i=0} R_{i,k} = \sum_{i:y_i=0} R_i - \sum_{i:y_i=0, i \in T_k} R_i - \sum_{i:y_i=0,\ i \notin T_k} \sum_{j \in T_k} I(R_j < R_i). \qquad (3)$$

This formula is based on the R_i ($i = 1, \dots, n$) only. Hence, it allows to compute the W_k simultaneously very efficiently. Computation of the p-values and ordering of the genes can then be carried out based on the standardized statistic W_k^* which is asymptotically normally distributed:

$$W_k^* = \frac{W_k - n_{0,k}(n - n_{T_k} + 1)/2}{\sqrt{n_{0,k}(n - n_{T_k} - n_{0,k})(n - n_{T_k} + 1)/12}}, \qquad (4)$$

where $n_{0,k}$ denotes the number of observations with $y_i = 0$ when observations from T_k are removed. In the k-th iteration, the best genes are those with the highest $|W_k^*|$ values.

Run time comparison

We compared the time needed to order 1000 genes in CV and MCCV by i) running the function `wilcox.test` for each CV or MCCV iteration, ii) using our novel efficient algorithm as implemented in the function `wilcox.selection.split` from our `R` package `WilcoxCV`. Results are given in Table 1 for different values of n and two different procedures: LOOCV and MCCV with size ratio 9:1 and $N_{iter} = 100$ iterations. As can be seen from Table 1, the new algorithm reduces computation time dramatically (up to a factor 50) compared to the standard approach (carrying out the Wilcoxon rank sum test for each iteration).

ACKNOWLEDGMENT

I thank Martin Daumer, Korbinian Strimmer and Elisabeth Gnatowski for critically reading the manuscript. This work was supported by the Porticus Foundation in the context of the International School for Technical Medicine and Clinical Bioinformatics.

REFERENCES

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

Boulesteix,A.-L., Tutz,G. (2006) Identification of interaction patterns and classification with applications to microarray data, *Computational Statistics and Data Analysis*, **50**:783–802.

Boulesteix,A.-L., Strimmer,K. (2007) Partial Least Squares: A versatile tool for the analysis of high-dimensional genomic data, *Briefings in Bioinformatics*, **8**:32–44.

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

Dai,J. J., Lieu,L., Rocke,D. (2006) Dimension reduction for classification with gene expression data, *Statistical Applications in Genetics and Molecular Biology*, **5**:6.

Dettling,M., Bühlmann,P. (2003) Boosting for Tumor Classification with Gene Expression Data, *Bioinformatics*, **19**:1061–1069.

Ioannidis,J. P. (2005) Microarrays and molecular research: noise discovery, *The Lancet*, **365**:488–492.

Lee,J. W., Lee,J. B., Park,M., Song,S. H. (2005) An extensive comparison of recent classification tools applied to microarray data, *Computational Statistics and Data Analysis*, **48**:869–885.

Molinaro,A., Simon,R., Pfeiffer,R. M. (2005) Prediction error estimation: a comparison of resampling methods, *Bioinformatics*, **21**:3301–3307.

Statnikov,A., Aliferis,C. F., Tsamardinos,I., Hardin,D., Levy,S. (2005) A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis, *Bioinformatics*, **21**:631–643.

Zhu,J., Hastie,T. (2004) Classification of gene microarrays by penalized logistic regression, *Biostatistics*, **5**:427–443.