LMU
LUDWIG-
MAXIMILIANS-
UNIVERSITÄT
MÜNCHEN

INSTITUT FÜR STATISTIK

Christoph Bernau, Anne-Laure Boulesteix

Variable Selection and Parameter Tuning in High-Dimensional Prediction

Technical Report Number 076, 2009
Department of Statistics
University of Munich

http://www.stat.uni-muenchen.de

Variable Selection and Parameter Tuning in High-Dimensional Prediction

Christoph Bernau[1] and Anne-Laure Boulesteix[1,2]

[1] Department of Medical Informatics, Biometry and Epidemiology
University of Munich, Marchioninistr. 15, 81377 Munich, Germany
bernau@ibe.med.uni-muenchen.de, boulesteix@ibe.med.uni-muenchen.de

[2] Department of Statistics
University of Munich, Ludwigstr. 33, 80539 Munich, Germany

Abstract. In the context of classification using high-dimensional data such as microarray gene expression data, it is often useful to perform preliminary variable selection. For example, the k-nearest-neighbors classification procedure yields a much higher accuracy when applied on variables with high discriminatory power. Typical (univariate) variable selection methods for binary classification are, e.g., the two-sample t-statistic or the Mann-Whitney test.

In small sample settings, the classification error rate is often estimated using cross-validation (CV) or related approaches. The variable selection procedure has then to be applied for each considered training set anew, i.e. for each CV iteration successively. Performing variable selection based on the whole sample before the CV procedure would yield a downwardly biased error rate estimate. CV may also be used to tune parameters involved in a classification method. For instance, the penalty parameter in penalized regression or the cost in support vector machines are most often selected using CV. This type of CV is usually denoted as "internal CV" in contrast to the "external CV" performed to estimate the error rate, while the term "nested CV" refers to the whole procedure embedding two CV loops.

While variable selection and parameter tuning via internal CV have been widely investigated in the context of high-dimensional classification, it is still unclear how they should be combined if a classification method involves both variable selection and parameter tuning. For example, the k-nearest-neighbors method usually requires variable selection and involves a tuning parameter: the number k of neighbors. It is well-known that variable selection should be repeated for each external CV iteration. But should we also repeat variable selection for each *internal CV* iteration or rather perform tuning based on fixed subset of variables? While the first variant seems more natural, it implies a huge computational expense and its benefit in terms of error rate remains unknown.

In this paper, we assess both variants quantitatively using real microarray data sets. We focus on two representative examples: k-nearest-neighbors (with k as tuning parameter) and Partial Least Squares dimension reduction followed by linear discriminant analysis (with the number of components as tuning parameter). We conclude that the more natural but computationally expensive variant with repeated variable selection does not necessarily lead to better accuracy and point out the potential pitfalls of both variants.

Keywords: class prediction, variable selection, parameter tuning, nested cross-validation, genomics

1 Background

In the context of classification using high-dimensional data such as microarray gene expression data, it is often useful to perform preliminary variable selection. For example, the k-nearest-neighbors classification procedure yields a much higher accuracy when applied on variables with high discriminatory power. Typical (univariate) variable selection methods for binary classification are, e.g., the two-sample t-statistic or the Mann-Whitney test.

In small sample settings, the classification error rate is often estimated using cross-validation (CV) or related approaches. From now on, we denote the whole data sample as S, the CV folds as $T_1, \ldots, T_J$ (with $\cup_{j=1}^{J} T_j = S$ and $T_{j_1} \cap T_{j_2} = \emptyset$ for $j_1 \neq j_2$), and the corresponding CV learning sets as $L_j = S \setminus T_j$, for $j = 1, \ldots, J$. If the chosen classification method involves a preliminary variable selection step, this step has to be applied for each CV iteration anew, i.e. for each considered learning set L_j successively. Performing variable selection based on the whole sample S before the CV procedure would yield a downwardly biased error rate estimate.

CV may also be used to tune parameters involved in a classification method. For instance, the penalty in penalized regression is most often selected using CV. This type of CV is usually denoted as "internal CV" in contrast to the "external CV" performed to estimate the error rate, while the term "nested CV" refers to the whole procedure embedding two CV loops. Similarly to the external CV, we denote the internal fold for the jth external CV iteration and ith internal CV iteration as T_{ij} and the corresponding learning sets as $L_{ij} = L_j \setminus T_{ij}$, for $i = 1, \ldots, I$ and $j = 1, \ldots, J$.

While variable selection and CV have been widely investigated in the context of high-dimensional classification, it is still unclear how variable selection and parameter tuning should be combined if a classification method involves both variable selection and parameter tuning. For example, the k-nearest-neighbors method usually requires variable selection and involves a tuning parameter: the number k of neighbors. It is well-known that variable selection should be repeated for each external CV iteration based on L_j, yielding a variable subset $\mathcal{V}_j$. However, should we also repeat variable selection for each *internal CV* iteration based on L_{ij} or rather perform internal CV with the fixed subset $\mathcal{V}_j$?

In the latter variant, termed "V1" from now on, variables are selected based on the learning set L_j corresponding to the current external CV iteration. Hence, the well-known rule that variable selection should be performed without taking the test set into account is violated in the internal CV. This is because the subset of variables $\mathcal{V}_j$ is derived from $L_j = L_{ij} \cup T_{ij}$ that, from the point of view of internal CV, includes both the learning data set L_{ij} and test data set T_{ij}, for $i = 1, \ldots, I$. As outlined above, variable selection can also be repeated for each internal CV iteration based on L_{ij} only, yielding variant "V2". In this case, the variable subset varies in each internal CV iteration. We denote the subset of variables selected in the ith internal CV iteration and jth external CV iteration as $\mathcal{V}_{ij}$.

In variant V1, the error rates computed in the internal CV are expected to be lower than 50% even if the class membership Y is random. That is because the variable subset $\mathcal{V}_j$ is chosen to be associated with Y in L_j, for each $j = 1, \ldots, J$. In other words, V1 performs tuning based on downwardly biased estimates of the error rate. The relative performance of parameter values in internal CV - and thus the result of the tuning procedure - may also be affected by the fact that the internal test data sets T_{ij} were not disregarded while selecting the subset $\mathcal{V}_j$.

In this sense, variant V2 seems more natural and adequate. However, it implies a higher computational expense and its benefit over V1 in terms of error rate remains unknown. An additional potential pitfall is that the variables that are used for tuning (i.e. that are selected at each internal CV iteration) are different from those that are eventually used to construct the classifier. For example, if we perform penalized regression based on variables of different scale, it would obviously be wrong to use a penalty parameter that was chosen based on other variables. This example is probably exagerated and in this case the problem can be simply solved through appropriate scaling, but more subtle similar mechanisms may in general affect the accuracy of V2.

To our knowledge, V1 and V2 are both used in practice – most often rather implicitly and without much explanation. Their respective merits and pitfalls remain largely unexplored in the literature, although tuning issues are known to greatly affect accuracy in general. In this paper, we assess both variants V1 and V2 quantitatively using real microarray data sets. We focus on two representative examples: k-nearest-neighbors (with k as tuning parameter) and Partial Least Squares dimension reduction followed by linear discriminant analysis (with the number of components as tuning parameter). More precisely, we address the following questions: 1) Do V1 and V2 select the same parameter values? and, if yes, 2) Do the resulting classification accuracies differ substantially?

2 Methods and design of the study

2.1 Classification and variable selection methods

In this paper, V1 and V2 are compared for two completely different standard classification methods: the k-nearest-neighbors (kNN) algorithm with the number k of neighbors as tuning parameter, and Partial Least Squares (PLS) dimension reduction followed by linear discriminant analysis (PLS+LDA), with the number *ncomp* of PLS components as a tuning parameter. We refer to Boulesteix (2004) for details on PLS+LDA. For the purpose of reproducibility, we use the standardized implementations provided by the 'CMA' Bioconductor package (Slawski et al. (2008)). We consider the classical candidate parameter values $k = 1, 3, 5, 7, 9$ for kNN and $ncomp = 1, 2, \ldots, 10$ for PLS+LDA.

Tens of variable selection criteria have been proposed in the context of microarray-based (binary) classification. In this study, we focus on two univariate methods. The first one selects the p^* genes with the highest absolute value of the two-sample t-statistic. The second one is the criterion provided by the Recursive Feature Elimination (RFE) approach by Guyon et al. (2002) based on support vector machines. For computational reasons, the number of "iterations" is set to 1. These two procedures are implemented in the 'CMA' package. In this study, the number of selected variables is fixed to $p^* = 20, 50$ successively for the kNN method, and $p^* = 100, 500$ for the PLS+LDA method, which are all common choices.

2.2 Design of the comparison study

The study is based on two well-known real-life cancer data sets: the leukemia data set (Golub et al. (1999)) included in the 'CMA' Bioconductor package (Slawski et al. (2008)) yielding very good accuracies with most standard classification methods, and the colon cancer data sets included in the 'colonCA' Bioconductor package (Alon et al. (1999)) usually yielding error rates between 10% and 20% (see, e.g., Boulesteix (2004)).

The CV procedure is replicated several times using different random partitions, both in internal and external CV, which means that error rates are averaged over several random partitions instead of only one. This approach is commonly recommended to make the results more stable. In our study, external CV consists of 100 replications of J-fold CV with $J = 6$, whereas five replications of 3-fold internal CV are performed for tuning. In other words, external CV error rates are obtained by averaging over $100 \times 6 = 600$ folds, while parameter values are selected based on the average error rate over three replications of 5-fold CV, i.e. $3 \times 5 = 15$ internal folds.

3 Results

3.1 Do V1 and V2 select the same tuning parameter values?

At first we examine the tuning parameter values that are selected in the $100 \times J$ tuning runs by internal CV using both variants V1 and V2. Do V1 and V2 select the same parameter values?

In about half of the setups, V1 and V2 yield similar results. For example, the barplot depicted in Figure 1 (right panel) shows the frequency of selection of each candidate number of PLS components with $p^* = 500$ and RFE variable selection based on the colon data. The frequencies of selection do not differ substantially for V1 and V2. Clear differences between V1 and V2 are observed in the other half of the settings, with V2 consistently selecting more complex models. As illustrated in the left panel of Figure 1 for kNN with

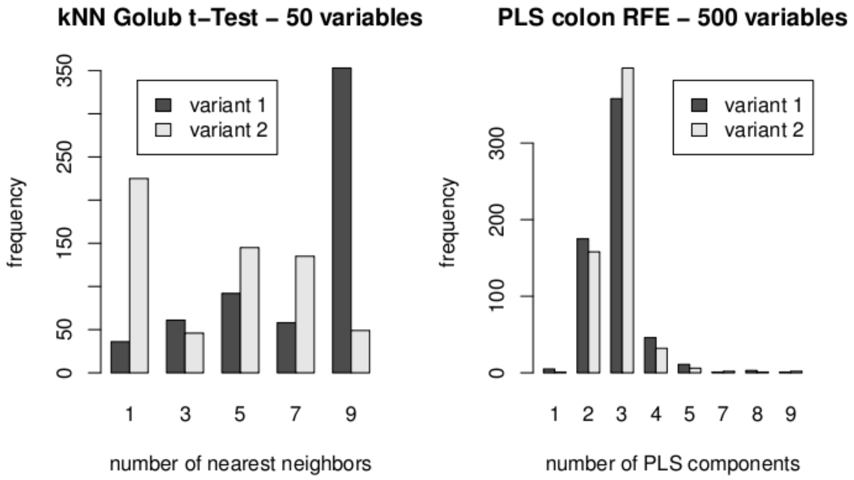

Fig. 1. Barplot of the frequency of selection of the candidate parameter values ($k = 1, 3, 5, 7, 9$ for kNN and $ncomp = 1, 2, \ldots, 10$ for PLS+LDA) for both variants V1 and V2 in two different illustrative setups. Whereas the right panel (PLS+LDA, RFE, $p^* = 500$, colon data) shows similar frequencies of selection for each candidate parameter value, obvious differences can be observed in the left panel (kNN, t-test, $p^* = 50$, Golub data). The barplots sum to $100 \times J = 600$.

t-test variable selection and $p^* = 50$ based on the Golub data, V2 noticeably selects smaller k values than V1 in kNN classification, i.e. more complex models. This general tendency of V2 to more complex models is also observed in several settings with PLS+LDA, where V2 selects higher numbers of PLS components.

The tendency of V1 to less complex models may be artificially enhanced by our convention that, if the lowest internal CV error rate is obtained with several parameter values, the least complex model is selected. Indeed, V1 performs tuning based on downwardly biased estimates of the error rate. With well-separated data sets such as Golub, it often occurs that all parameter values yield an error rate of 0%. In this case, the least complex model is selected, hence artificially increasing the frequency of selection of less complex parameter values (i.e. high k values or low $ncomp$ values). However, this artificial mechanism resulting from our convention can only explain a moderate part of the tendency of V2 to more complex models.

Roughly speaking, the higher complexity obtained with V2 can be partly explained as follows. With V2, the set of variables $\mathcal{V}_{ij}$ used in the jth external iteration and ith internal iteration is selected based on L_{ij} only. Thus, in the learning set L_{ij}, they are more strongly associated to the response Y than the variables $\mathcal{V}_j$ selected using the larger subsample L_j. As a consequence, a complex model fitted on L_{ij} based on variables $\mathcal{V}_{ij}$ is likely to perform better than a complex model fitted with the "worse" variables $\mathcal{V}_j$. This mechanism can probably partly explain why V2 leads to the selection of more complex models than V1. It is illustrated in Figure 2 which shows the prediction regions of the kNN classifier (based on only $p^* = 2$ for demonstration purposes) together with a scatterplot of the $p^* = 2$ selected variables. In this example, complex models ($k = 1$) in combination with V1 obviously yield overcomplex prediction regions leading to bad classification performance on the internal test data set T_{ij}.

3.2 Do the classification accuracies of V1 and V2 differ substantially?

In this section, we examine the differences in performance of V1 and V2 in external CV. On the whole, our conclusion is that both tuning variants yield approximately equal accuracies, with differences in accuracies smaller than 2.5% in all settings. As an example, the error rates obtained with the kNN method are summarized in Table 1. Similar differences are observed with PLS+LDA, see Table 2.

In the cases where V1 outperforms V2, the difference in performance is then largely due to the tendency of V1 to less complex models, as can be seen from the example depicted in Figure 3 (kNN with RFE, colon data). In this setup, the parameter value $k = 1$ (complex model) is not often selected by V1, and yields higher error rates than the other k values, regardless of whether it is chosen by V1 or V2. V1 thus seems to benefit from its tendency to less complex models.

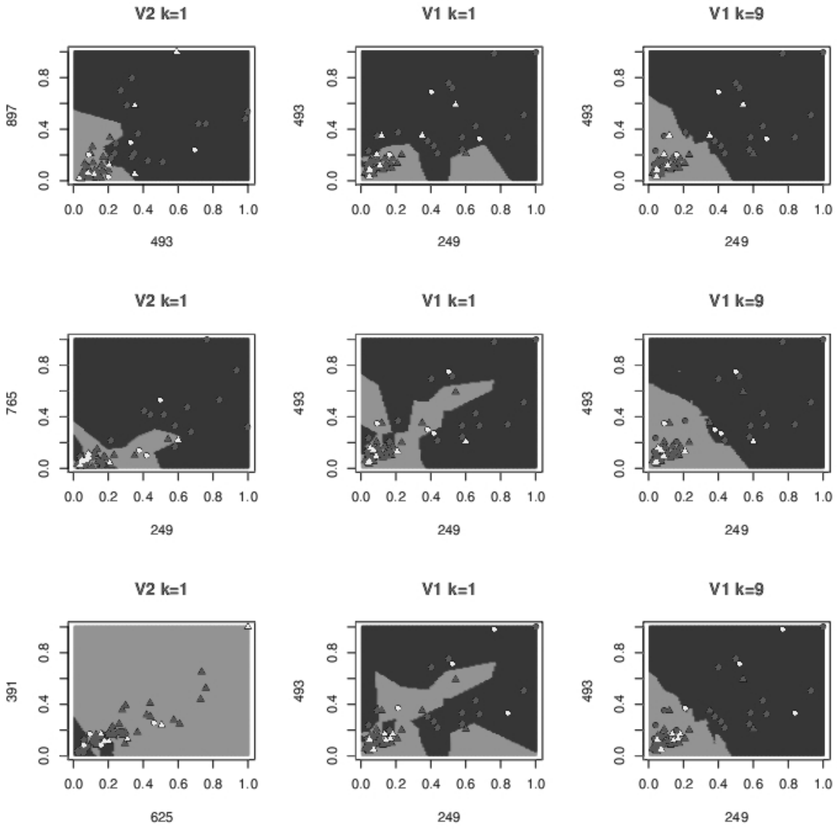

Fig. 2. Prediction regions of kNN classifiers with $p^* = 2$ variables for three internal CV iterations based on variants V1 and V2. Each row corresponds to a particular internal CV iteration. **1st column:** V2 with $k = 1$ neighbor. **2nd column:** V1 with $k = 1$ neighbor. **3rd column:** V2 with $k = 9$ neighbors. Circles stand for observations of class $Y = 0$, triangles stand for $Y = 1$. White symbols represent internal test observations from T_{ij}.

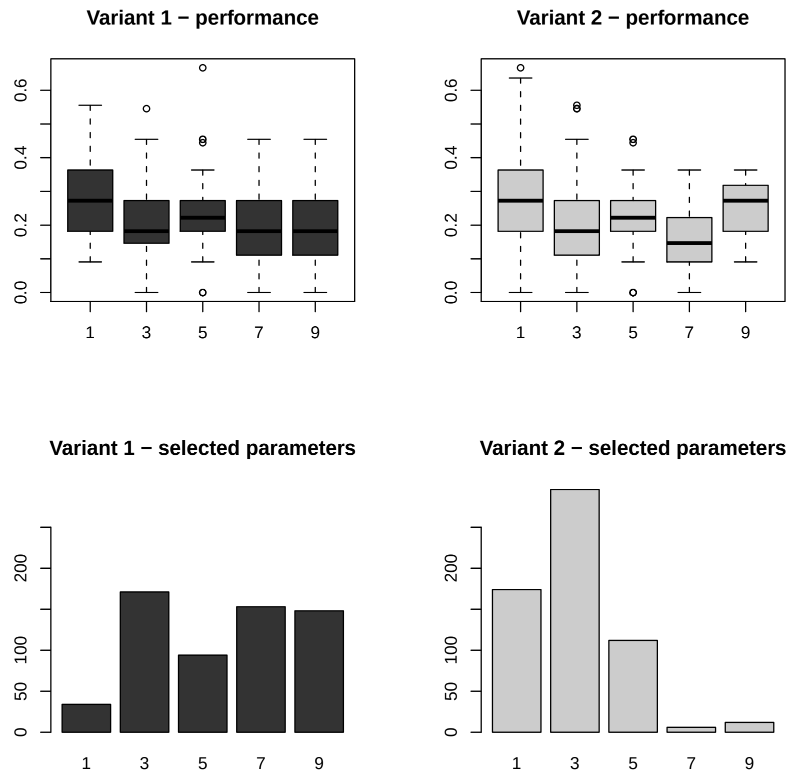

Fig. 3. kNN, RFE, $p^* = 20$, colon data. **Top:** Boxplots of the error rates in external CV for different values of k with V1 (left) and V2 (right). **Bottom:** Barplots of the frequencies of selection of the different k values with V1 (left) and V2 (right).

kNN		Golub data				colon cancer data			
		t-test		RFE		t-test		RFE	
		V1	V2	V1	V2	V1	V2	V1	V2
20 genes	mean MCR	7.8%	7.4%	5.8%	6.1%	16.8%	18.8%	21.6%	23.3%
	std. dev.	2.6%	2.8%	2.5%	2.9%	1.9%	2.4%	3.3%	4.1%
50 genes	mean MCR	5.9%	5.5%	1.9%	2.2%	16.4%	19.9%	16.9%	18.5%
	std. dev.	2.4%	2.7%	1.8%	1.7%	1.6%	1.9%	3.3%	3.0%

Table 1. Mean error rates (and standard deviations) with kNN using V1 and V2.

PLS LDA		Golub data				colon cancer data			
		t-test		RFE		t-test		RFE	
		V1	V2	V1	V2	V1	V2	V1	V2
100 genes	mean MCR	3.3%	4.0%	0.2%	1.2%	16.3%	14.3%	13.0%	12.3%
	std. dev.	1.7%	2.3%	0.7%	1.6%	2.5%	1.4%	2.3%	1.3%
500 genes	mean MCR	1.8%	2.3%	0.2%	1.1%	15.7%	15.1%	12.0%	12.1%
	std. dev.	1.7%	2.0%	0.6%	1.5%	2.5%	1.8%	1.4%	1.4%

Table 2. Overall mean MCRs and standard deviations obtained using PLS+LDA and various combinations of tuning variant, variable selection technique and data set. Reported standard deviations are computed based on mean MCRs of the 100 CV iterations.

4 Discussion

Our study shows that the two investigated tuning variants sometimes lead to clearly different tuning results. Variant V1 shows a general tendency to less complex models using both investigated data sets. Similar results are also obtained using further microarray data sets (data not shown). With regard to prediction accuracy, V1 and V2 yield similar accuracies and, in some settings, the seemingly inappropriate V1 approach even outperforms the more natural V2. Although V1 performs tuning based on severely biased internal CV error rates, the selected tuning parameter values yield acceptable accuracies in the settings considered in our study. Hence, the benefit of V2's higher computational expense in terms of prediction accuracy cannot be confirmed through our study. Let us add a few concluding remarks on both approaches:

- In some setups, we see that overcomplex models, which are frequent with V2, are associated with an increased error rate. This may be partly explained as follows. Since the L_{ij} are smaller than the L_j, it is easier to find variables that separate the two classes well in L_{ij} than in L_j. For these variables $\mathcal{V}_{ij}$, complex models perform well when fitted on L_{ij} – probably even better than when they are fitted on variables $\mathcal{V}_j$ in L_j and using larger data sets. This may partly explain why the tendency of V2 to more complex models seems to be a disadvantage.
- The variables used by V2 to construct the classifier in external cross-validation are not the same as those used for tuning in internal cross-validation. Beyond the examples considered in our paper, this may yield substantial problems in some cases, for instance when the tuning parame-

ter controls the ”amount of non-linearity” of a classifier. If some variables show linear relationships with Y while other substantially depart from linearity, performing parameter tuning and classifier fitting using different variables is obviously sub-optimal.

- Finally, we point out that V1 may show worse performance in data sets with well-separated classes (like the Golub data) if all parameter values yield an error rate of 0% in internal CV. This may often occur in practice, since the internal CV error rates are strongly downwardly biased in V1. In this case, no tuning is achieved by V1, while V2 often yields higher error rates that can be compared to perform parameter tuning.
- In conclusion, let us mention that, from a theoretical point of view, both variants V1 and V2 can be seen as imperfect approximations for a computationally unfeasible task. The correct approach would be to select the tuning parameter and the variable subset jointly from a multi-diensional grid in internal CV. Of course, an exhaustive search is unfeasible in high-dimensional data analysis. The development of simplified computationally efficient algorithms could be addressed in further research.

Acknowledgments

This project was supported by DFG project BO3139/2-1 and by the LMU-innovativ Project BioMed-S.

References

ALON, U., BARKAI, N., NOTTERMAN, K., GISH, D.A., YBARRA, S., MACK, D. and LEVINE, A.J. (1999): Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissue probed by oligonucleotide arrays. *PNAS 15, 6745-6750.*

BOULESTEIX, A.-L. (2004): PLS Dimension reduction for classification with microarray data. *Statistical Applications in Genetics and Molecular Biology 3,33.*

GOLUB, T.R., SLONIM, G.K., TAMAYO, P., HUARD, C., Gaasenbeek, M. MESIROV, J.P., COLLER, H., LOH, M.L., DOWNING, M.A., CALIGIURI, J.R., BLOOMFIELD, C.D. and LANDER, E.S. (1999): Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science 286, 531-537.*

GUYON, I., WESTON, J., BARNILL, S. et al. (2002): Gene selection for cancer classification using support vector machines. *Machine Learning 46, 389-422.*

SLAWSKI, M., DAUMER, M. and BOULESTEIX, A.-L. (2008): CMA a comprehensive Bioconductor package for supervised classification with high dimensional data. *BMC Bioinformatics 9, 439*.