Variable Selection in Random Forest with Application to Quantitative Structure-Activity Relationship

Vladimir Svetnik, Andy Liaw, and Christopher Tong

Biometrics Research, Merck & Co., Inc.
P.O. Box 2000 RY33-300, Rahway, NJ 07065, USA
{vladimir_svetnik, andy_liaw, christopher_tong}@merck.com

Abstract. A wrapper variable selection procedure is proposed for use with learning machines that generate a measure of variable importance, such as Random Forest. The procedure is based on iteratively removing low-ranking variables and assessing the learning machine performance by cross-validation. The procedure is implemented for Random Forest on some QSAR modeling examples from drug discovery and development. It is shown that the non-recursive version of the procedure outperforms the recursive version, and that the default Random Forest *mtry* function is usually adequate. The paper concludes with some comments about performance assessment and the dangers of using Random Forest's out-of-bag error estimate in a variable selection wrapper.

1 Introduction

In supervised learning problems involving very high dimensional data, it is often desirable to reduce the number of variables given to the learning machine [Guy03]. This is because the removal of irrelevant or "noise" variables may improve the performance of the learning machine. Furthermore, identifying only those variables that are important for classification or regression may help in the interpretation of the model. An important example is *quantitative structure-activity relationship* (QSAR) modeling [Eki00,Haw01,Liv00,Sto93]. Here, one posits that the biological activity of a chemical compound is a function of the compound's molecular structure (or other properties). The molecular structure is described using, for instance, topological descriptors like atom pairs [Car85]. One would like to use a compound's descriptors to predict its biological activity. In drug discovery and development programs, the number of available descriptors can often be much larger than the available number of compounds.

An optimal solution to the selection of the subset of important variables requires an exhaustive search over all possible subsets, which is computationally intractable. Therefore, one is compelled instead to seek a sub-optimal, "greedy" solution that is computationally feasible and has a good heuristic motivation. In any case, there may not be a unique solution if many variables are correlated with each other.

There are a number of approaches to solving the (unsupervised learning) problem of variable selection (a.k.a., feature selection) [Koh97]. One approach is called the "filter," which requires variable selection up front, independently of the learning machine. The learning machine is subsequently trained on the resultant reduced dimensional data set. An example is removing variables that do not pass a test of statistical significance. A second approach is called the "wrapper" approach. Here, the learning machine is "wrapped" by a loop that monitors its performance as variables are added or deleted. If the learning machine produces its own measure of variable importance, this importance measure can be incorporated into the wrapper algorithm as well. The wrapper approach is appealing because variable selection and supervised learning are essentially optimized together, as a complete learning system, rather than sub-optimized individually. Nonetheless, one should at least apply a weak filter to one's data (e.g., removing variables with no variance or that are collinear with other variables).

In this report, a wrapper approach is proposed for use with Random Forest, an ensemble learning machine recently introduced by Breiman [Bre01]. An application to a problem in drug discovery and development, QSAR modeling mentioned above, will be used to illustrate the procedure. In Section 2, a general procedure for wrapper variable selection is presented, with an implementation using Random Forest. In Section 3, this procedure is compared favorably with a recursive version of it. In Section 4, the optimization of Random Forest's tuning parameter *mtry* is considered in conjunction with variable selection. Finally, some discussions and conclusions are given. In particular, we discuss an "honest" assessment of performance for our procedure, as well as the hazards of using the Random Forest out-of-bag error estimate in a variable selection wrapper.

2 A procedure for wrapper variable selection

Here we introduce a general procedure for wrapper variable selection. It was designed with Random Forest in mind, but in principle it could be applied to any learning machine that produces a measure of variable importance. This measure of importance is used to rank the variables from most to least important, and the learning machine's performance can be monitored as more and more of the least important variables are iteratively removed. The essential idea of our procedure is to convert the problem of variable selection into a problem of parameter selection, where the parameter of interest is p', the number of important variables to use in the learning machine. This p' is found by cross-validation (CV), and when a final model is required, one trains the learning machine on all the data, re-ranks the variables, and selects the most important p' of them for use in a reduced dimensional model.

Here are the specific steps of the procedure.

1. Partition the data for k-fold cross-validation. (We usually take $k = 5$.)
2. On each CV training set, train the learning machine using all variables, to produce a ranking of the variables according to their importance. Record the CV test set predictions.

3. Remove the least important fraction (e.g., half) of the variables and train another learning machine on the remainders, again recording the CV test set predictions. Repeat this step of removing a fraction (e.g., half) of the variables until a small number (say, two) remain.
4. Aggregate the predictions from all k CV test sets and compute the aggregate error rate (or MSE) at each step down in number of variables.
5. Steps 1-4 are repeated 10-50 times to smooth out the variability; select p' that minimizes the curve of median error rate (or MSE) vs. number of variables. (See Fig. 1.)
6. Once p' is selected, train a learning machine now on all the data, producing a ranking of the variables, and take only the most important p' variables to input into the final learning machine. This final machine can now be used to predict future batches of data.

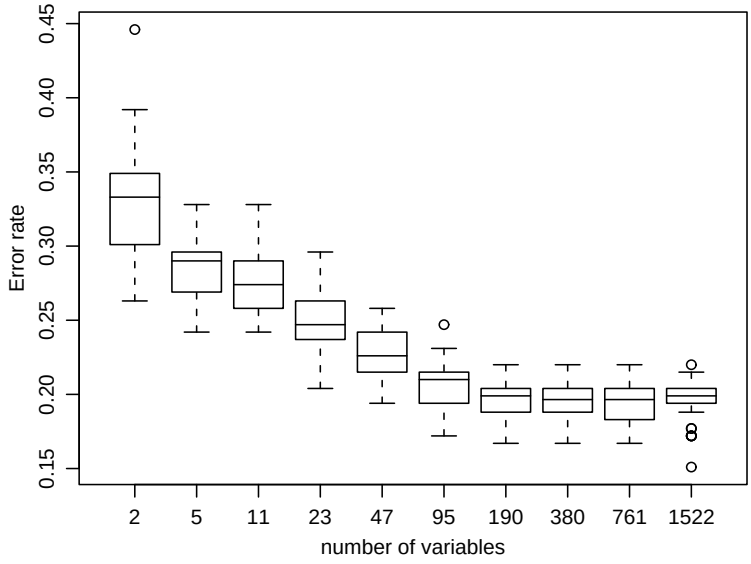

Fig. 1. Boxplots of 50 5-fold CV test error rates at each step of halving the important variables (P-gp data).

As an example, consider a QSAR modeling problem studied by Penzotti et al. [Pen02]. A collection of 186 drug compounds were evaluated for their P-glycoprotein (P-gp) transport activity, and were classified as P-gp substrates or non-substrates. To predict the compounds' classes, we used a set of 1522 atom pair (AP) descriptors to build Random Forest machines within 5-fold cross-validation. The result of Step 5 of the above procedure is shown in a set of boxplots (Fig. 1). In this case, the performance of Random Forest actually does not improve with variable selection, but remains steady until too many variables

are removed, at which point the performance steadily degrades. Arguably, one could reduce the number of variables to 190 without degrading performance.

Note that throughout this paper, Random Forest's margin-based variable importance measure is used. The results are very similar if one uses the Gini-based variable importance measure, since the atom pairs are binary descriptors. In general, the use of the Gini-based variable importance measure is problematic since it prefers variables that have a large number of distinct values over those with a smaller number of distinct values (Leo Breiman, personal communication).

3 Comparison of our wrapper with a recursive approach

A reasonable alternative to the procedure outlined above is one that is exactly the same, except that the variable importance is *recalculated* at step 3, producing a new ranking of the variables. This corresponds to a recursive feature elimination procedure similar to the one used by Guyon et al. [Guy02]. Our limited experience shows that this approach performs poorly, since it is much greedier than the non-recursive approach. A comparison of the performance of these two approaches is shown in Fig. 2 for the P-gp data mentioned earlier. Initially, there is not much difference between the non-recursive and recursive approaches; however, as variable reduction proceeds onward, the recursive approach produces much higher error rates.

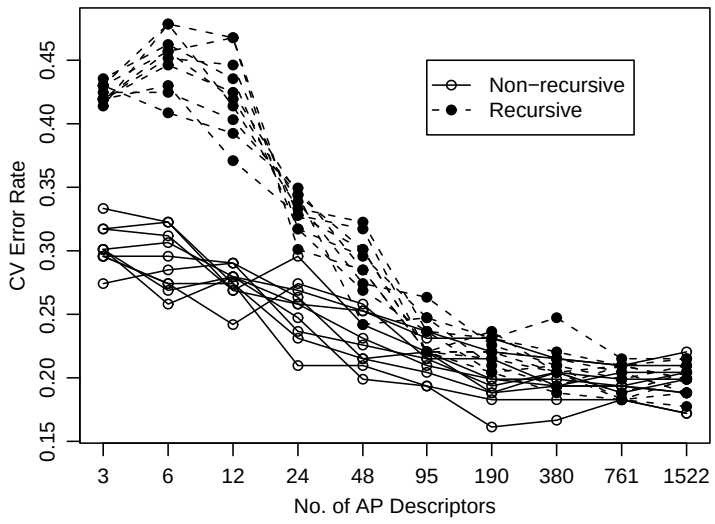

Fig. 2. Comparisons of the medians of CV test error rate performance of our wrapper variable selection procedure and a recursive version of it (P-gp data). The traces correspond to ten replications of 5-fold cross-validation.

The following regression QSAR problem presents an even more severe illustration. A set of 116 compounds were assayed for their binding affinity to the dopamine D_2 receptor, as measured by their $\log(IC_{50})$ [Gil92,She94]. We generated a set of 374 AP descriptors for this compound collection. The results of both the non-recursive and recursive procedures are shown in Fig. 3. The plot shows the recursive approach's almost immediate degradation of performance as the number of variables is reduced. Meanwhile, even the non-recursive procedure shows that there is actually no gain in performance as variables are removed.

Because of its inferior performance, we recommend avoiding the recursive procedure, and we will not discuss it further in this paper.

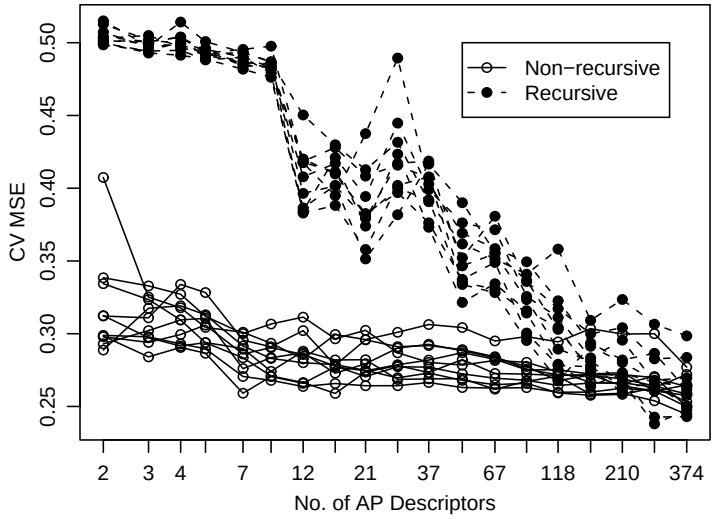

Fig. 3. Comparisons of the medians of CV test error rate performance of our wrapper variable selection procedure and a recursive version of it (dopamine receptor data). The traces correspond to ten replications of 5-fold cross-validation.

4 Optimization of Random Forest *mtry*

Random Forest has a tuning parameter, *mtry*, which is the number of variables randomly sampled at each node to be considered for splitting. We usually choose *mtry* to be some function of p, the number of variables in the input data. In the Random Forest software, the default for classification is $mtry = \sqrt{p}$ and for regression it is $mtry = p/3$ [Lia02]. In principle, one should use cross-validation to optimize the choice of the function $mtry = f(p)$. Three other potential choices for this function include $mtry = p$ (equivalent to bagging), $mtry = p/2$, and

$mtry = p/4$. (There are even examples where $mtry = 1$ is best; however, we don't expect this choice to work with AP descriptors, so we avoid its use here.)

When considering variable selection, p becomes the reduced number of variables, and in principle one should simultaneously optimize the number of variables and $mtry$. Once again, this would be computationally intractable. Instead, we have tried running our wrapper variable selection procedure using the four different mtry functions mentioned above. We usually find that there is not much difference between the performance for the four functions, although bagging seems to be the worst performer when it differs from the others. There are certainly exceptions (in one data set, not presented here, bagging is actually the best performer). The results for the P-gp data are shown in Fig. 4. In this case, the default $mtry$ function does seem to perform the best.

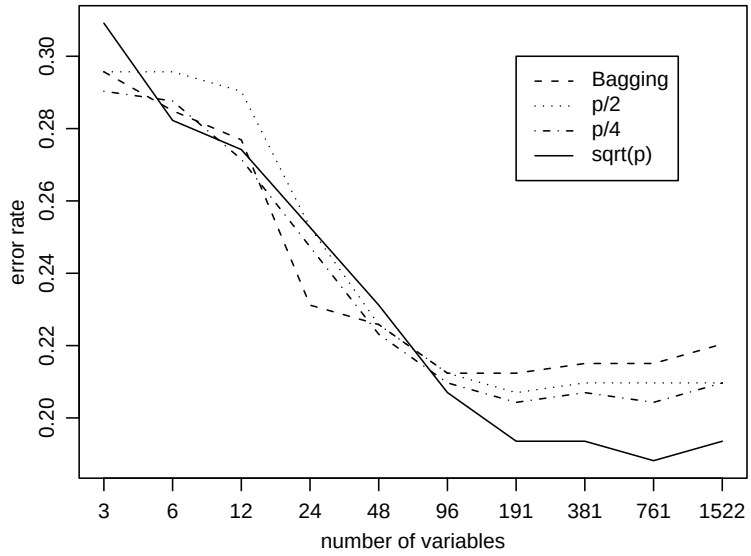

Fig. 4. Median CV test error rates at each step of halving the important variables, using different $mtry$ functions, for the P-gp data. Line segments connect the medians of 20 5-fold CV error rates.

5 Discussions and conclusions

In this report, a wrapper variable selection procedure is proposed based on using a measure of importance to rank variables for iterative removal. Cross-validation is used to select an optimal number of variables for training. The procedure is applied to Random Forest QSAR modeling in two example data sets. It is shown that the non-recursive version of the procedure outperforms the recursive version, and that the default $mtry$ function usually gives reasonably good performance.

Ambroise and McLachlan [Amb02] and Reunanen [Reu03] have argued compellingly that the assessment of performance of a learning machine with variable selection requires great care. *Both* the variable selection and the supervised training should be embedded within a CV procedure in order to obtain an honest assessment of the total learning system's performance. If the variable selection is done outside of CV, a *selection bias* is introduced. Following this principle, a performance assessment of our procedure requires that the entire algorithm be nested within another CV loop [Reu03]. For the P-gp data set, the median nested CV error rate based on 10 replications of 5-fold CV embedded within 10 replications of 10-fold CV is 0.191. This performance is actually quite consistent with Fig. 1, indicating that the selection bias is negligible in this example. Note that when using cross-validation for performance assessment, one should keep in mind the bias/variance properties of CV estimates [Dav97].

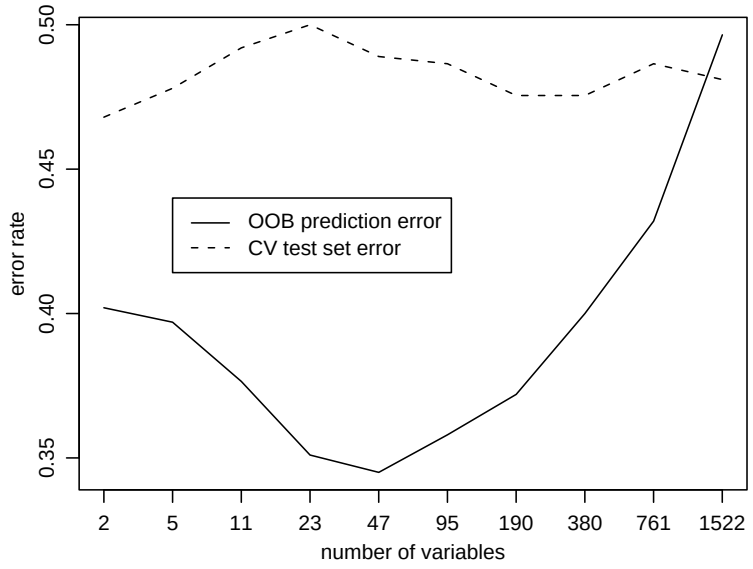

Fig. 5. Medians of CV test set and out-of-bag prediction error rates at each step of halving the important variables (P-gp data with randomized response vector). Line segments connect the medians of 20 5-fold CV error rates.

Nonetheless, carelessness can still lead to disaster, as illustrated in the following exercise. Suppose that in our procedure that instead of using CV to assess performance in the wrapper, we instead use Random Forest's out-of-bag (OOB) prediction error. Minimizing the curve of OOB prediction error turns out to be extremely misleading. We tried this with the P-gp data with the class labels *randomly scrambled*. We compared the CV Test set error and the OOB prediction error during variable selection (see Fig. 5). The CV Test set error does reflect what we expect: performance close to that of random guessing, regardless of

how many variables are used. On the other hand, the OOB prediction error rate incorrectly suggests that performance can be improved substantially by doing variable reduction. This shows that the use of out-of-bag error estimates for iterative performance assessment lends itself to severe overfitting. This is because the OOB error estimate for any reduced number of variables is contaminated by the initial variable ranking, which was based on *all* the data. The curve in Fig. 5 shows that with a sufficient number of variables remaining, Random Forest can do a good job of fitting to noise.

Acknowledgments We would like to thank Leo Breiman and Tom Dietterich for useful discussions; Peter Grootenhuis and Michelle Lamb for valuable correspondence regarding their data; and Bob Sheridan, Chris Culberson, and Brad Feuston for help with descriptor generation.

References

[Amb02] Ambroise, C., McLachlan, G. J.: Selection bias in gene extraction on the basis of microarray gene-expression data. *Proc. Natl. Acad. Sci. USA* **99** (2002) 6562–6566

[Bre01] Breiman, L.: Random Forests. *Machine Learning* **45** (2001) 5–32

[Car85] Carhart, R. E., Smith, D. H., Venkataraghavan, R.: Atom pairs as molecular features in structure-activity studies: definitions and applications. *J. Chem. Inf. Comput. Sci.* **25** (1985) 64–73

[Dav97] Davison, A. C., Hinkley, D. V.: *Bootstrap Methods and their Application.* Cambridge University Press, Sec. 6.4

[Eki00] Ekins, S. et al.: Progress in predicting human ADME parameters in silico. *J. Pharmac. Toxic. Meth.* **44** (2000) 251–272

[Gil92] Gilligan, P. J. et al.: Novel piperidine σ receptor ligands as potential antipsychotic drugs. *J. Med. Chem.* **35** (1992) 4344–4361

[Guy02] Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. *Machine Learning* **46** (2002) 389–422

[Guy03] Guyon, I., Elisseef, A.: An introduction to variable and feature selection. *J. Machine Learning Res.* **3** (2003) 1157–1182

[Haw01] Hawkins, D. M., Basak, S. C., Shi, X.: QSAR with few compounds and many features. *J. Chem. Inf. Comput. Sci.* **41** (2001) 663–670

[Koh97] Kohavi, R., John, G.: Wrappers for feature subset selection. *Artificial Intelligence* **97** (1997) 273–324

[Lia02] Liaw, A., Wiener, M.: Classification and regression by randomForest. *R News* **2/3** (2002) 18–22

[Liv00] Livingstone, D. J.: The characterization of chemical structures using molecular properties: a survey. *J. Chem. Inf. Comput. Sci.* **40** (2000) 195–209

[Pen02] Penzotti, J. E., Lamb, M. L., Evensen, E., Grootenhuis, P. D. J.: A computational ensemble pharmacophore model for identifying substrates of p-glycoprotein. *J. Med. Chem.* **45** (2002) 1737–1740

[Reu03] Reunanen, J.: Overfitting in making comparisons between variable selection methods. *J. Machine Learning Res.* **3** (2003) 1371–1382

[She94] Sheridan, R. P., Nachbar, R. B., Bush, B. L.: Extending the trend vector: the trend matrix and sample-based partial least squares. *J. Computer-Aided Molecular Design* **8** (1994) 323–340

[Sto93] Stone, M., Jonathan, P.: Statistical thinking and technique for QSAR and related studies. I. General theory. *J. Chemometrics* **7** (1993) 455–475