Why Classification Models Using Array Gene Expression Data Perform So Well: A Preliminary Investigation Of Explanatory Factors

C. F. Aliferis
Discovery Systems Laboratory
Department of Biomedical Informatics
Vanderbilt University
Nashville, TN 37232, USA

I. Tsamardinos
Discovery Systems Laboratory
Department of Biomedical Informatics
Vanderbilt University
Nashville, TN 37232, USA

P. Massion
Department of Medicine
Vanderbilt University
Nashville, TN 37232, USA

A. R. Statnikov
Discovery Systems Laboratory
Department of Biomedical Informatics
Vanderbilt University
Nashville, TN 37232, USA

D. Hardin
Department of Mathematics
Vanderbilt University
Nashville, TN 37232, USA

Abstract Results in the literature of classification models from microarray data often appear to be exceedingly good relative to most other domains of machine learning and clinical diagnostics. Yet array data are noisy, and have very small sample-to-variable ratios. What is the explanation for such exemplary, yet counter-intuitive, classification performance? Answering this question has significant implications (a) for the broad acceptance of such models by the medical and biostatistical community, and (b) for gaining valuable insight on the properties of this domain.

To address this problem we build several models for three classification tasks in a gene expression array dataset with 12,600 oligonucleotides and 203 patient cases. We then study the effects of: classifier type (kernel-based/non-kernel-based, linear/non-linear), sample size, sample selection within cross-validation, and gene information redundancy. Our analyses show that gene redundancy and classifier choice have the strongest effects on performance. Linear bias in the classifiers, and sample size (as long as kernel classifiers are used) have relatively small effects; train-test sample ratio, and the choice of cross-validation sample selection method appear to have small-to-negligible effects.

Keywords: Bioinformatics and Medicine, Gene Expression, Expression Data Analysis

1 Introduction

Microarray analysis has generated a spectrum of opportunities for the creation of diagnostic, therapeutic and prognostic models and for better understanding the biology of several major diseases. In various scientific forums, including clinical settings, observers (i.e., clinicians, statisticians, data mining analysts that primarily do not work first-hand with such datasets) often express reservations about the results reported in the literature in light of *prima facie* legitimate concerns about very small sample sizes and gene-to-sample ratios. There is also a growing concern in the data mining literature that published models are often over-fitted (e.g., by choosing classifier parameters manually in a non-blinded way to the test data, or by examining in a blinded fashion too many models using automated search procedures) [7,15]. To what extend are the reported results in microarray analysis an artifact of the evaluation methodologies employed? What methods should be employed to rule out the possibility of poor methodology inflating estimates of performance? And what factors explain these performance levels that, admittedly, may run counter to intuition?

In this research we explore several factors related to model-building for explaining performance of array-based

classification in the context of lung cancer, a disease in which we have a long-term research interest and previous model-building experience [1,2].

2 Data and Methods

Our approach consists of building diagnostic models under a variety of experimental settings and examining the sensitivity of final performance on the settings (i.e., the *explanatory factors*). In previous experiments we had developed computer models that use gene expression data to distinguish between metastatic and non-metastatic ("task I"), tumor and non-tumor ("task II"), and between histological subtypes of lung cancer ("task III") [1]. These models exhibit excellent classification performance (even with small subsets of genes when further using gene selection methods, and independently of the gene selection methods used). Here we build several alternative models to examine the effect of a number of explanatory factors and compare it against this (baseline) performance. Another way to describe our approach is that by making the baseline models "break down" we seek to obtain an understanding for the tolerance of such models to a number of relevant modeling factors.

First we note that we adopt the assumption that model builders typically apply cross-validation or other regularization methods to optimize gene selection and classification parameters, and to select among several models. In the case of cross-validation (the most wide-spread form of empirical regularization) this involves recursive (a.k.a. "nested") application of the principle "test on cases that were not used for training" to first perform gene selection, then optimize parameters for a classifier, and choose the best classifier [8]. Thus we do not examine the effect of omitting cross-validation since it is well-established in the data mining and statistical literature [7,8,15] that such an omission leads to overestimated performance, and in the bioinformatics literature some form of regularization is always used. Similarly we do not investigate the effect of estimating classification performance using accuracy (a.k.a, "0/1 loss", - a measure that is sensitive to the prior distribution of the classification categories [13]). Instead, we used the area under the ROC curve [13] (which is not affected by the prior distribution).

2.1 Data

We analyze the data of Bhattacharjee et al., which is a set of 12,600 gene expression measurements (Affymetrix oligonucleotide arrays) per subject, from 203 patients and normal subjects. The original study explored identification of new molecular subtypes and their association to survival. Hence the experiments presented here do not replicate or overlap with those of [3].

2.2 Evaluation Metric

We use our own Matlab implementation of computation of AUC using the trapezoidal rule [5]. Statistical comparisons among AUCs are performed using a paired Wilcoxon rank sum test [11].

2.3 Classifiers

We use linear and polynomial-kernel Support Vector Machines (LSVM, and PSVM respectively) [14], K-Nearest Neighbors (KNN) [8], and feed-forward Neural Networks (NNs) [9]. For SVMs we use the LibSVM base implementation [4] that implements Platt's algorithm [12], with C chosen from the set: {1e-14, 1e-3, 0.1, 1, 10, 100, 1000} and degree from the set {2, 3, 4}. For KNN, we chose k from the range [1,…,number_of_variables] using our own implementation of the algorithm (following [8]). For NNs we used the Matlab Neural Network Toolbox [6] with 1 hidden layer, number of units chosen (heuristically) from the set {2, 3,

5, 8, 10, 30, 50}, variable learning rate back propagation, performance goal=1e-8 (i.e., an arbitrary value very close to zero), a fixed momentum of 0.001, and number of epochs chosen from the range [100,…,10000]. The number of epochs in particular is optimised via special scripts with nested cross-validation during training such that training would stop when the error in an independent validation set would start increasing. To avoid overfitting, either in the sense of optimising parameters for classifiers, or in the sense of estimating final performance of the best classifier/gene set found [8] a nested cross-validation design is employed, in which the outer layer of cross-validation estimates the performance of the optimised classifiers while the inner layer chooses the best parameter configuration for each classifier). For the two tasks (adenocarcinoma-squamous, and normal-cancer) we use 5-fold cross-validation while for the metastatic-nonmetastatic task we use 7-fold cross-validation (since we had only 7 metastatic cases in the sample). To ensure optimal use of the available sample, we require that data splits are balanced (i.e., instances with the rarer of the two categories for each target would appear in the same proportion among random data splits).

2.4 Explanatory Factors

(a) *Overfitting*: we replace actual gene measurements by random values in the same range (while retaining the outcome variable values); (b) *Target class rarity*: we contrast performance in tasks with rare vs non-rare categories. (c) *Sample size*: we use samples from the set {40,80,120,160, 203} range (as applicable in each task). (d) *Predictor redundancy*: we replace the full set of predictors by random subsets with sizes in the set {500, 1000, 5000, 12600}. (e) *Train-test split ratio:* we use train-test ratios from the set {80/20, 60/40, 40/60} (for tasks II and III, while for task I modified ratios were used due to small number of positives, see Figure 1). (f) *Cross-validated fold construction*: we construct n-fold cross-validation samples retaining the proportion of the rarer target category to the more frequent one in folds with smaller sample, or, alternatively we ensure that all rare instances are included in the union of test sets (to maximize use of rare-case instances). (g) *Classifier type*: Kernel vs non-kernel and linear vs non-linear classifiers are contrasted. Specifically we compare linear and non-linear SVMs [13] (a prototypical kernel method) to each other and to KNN (a robust and well-studied non-kernel classifier and density estimator [8]).

3 Results

Figure 1 shows the performance achieved with all genes and standard n-fold cross-validation (baseline models). In all our experiments we found that the performance of polynomial SVMs was uniformly not different than the linear SVMs, thus to simplify presentation, only results for linear SVMs are reported in the remainder of the paper. One can also see that performance is not affected by the prior of the true category (compare tasks I, II, and III in Figure 1). Figure 2 shows that performance diminishes (i.e., returns to uninformative levels of AUC~0.5) when random gene measurements replace the true ones. We note that without cross-validation, and given the large number of genes-to-sample points, even random values such as the ones used here are sufficient to overfit the models (i.e., minimize empirical error while leaving generalization error to random guessing levels). We also note that deviations from 0.5 in the figure are due to sampling variation and due to use of the trapezoidal rule with a small test set to determine the AUC [5].

Figure 3 shows that even the smallest samples investigated (~40 arrays) are enough to give excellent classification

performance when all genes are included in the analysis (especially when using SVMs, much less so with KNN). Figure 4 shows that there is substantial redundancy in the classification-related information carried by the gene probes. Replacement of the full gene set by random subsets as small as 500 genes yields minor decrease in classification performance of SVM models (KNN's performance is more sensitive to this reduction in predictor number). Figure 5 shows that performance is mildly sensitive to the train-test sample ratio when the target category is rare (task I), and insensitive otherwise. In general, SVM methods are uniformly equally or more robust than KNN. Finally, as can be seen in Figure 6, n-fold cross-validation results are mildly sensitive (task I or sample=40) to whether in folds with smaller sample the rarer target category retains its proportion to the more frequent one, or all rare instances are included.

4 Conclusions and Discussion

Our experiments are limited in that we did not examine all possible explanatory factor combinations (due to the very large factorial space of possible models when one also considers nested cross-validation). Nevertheless, they do provide a number of interesting conclusions: first, we ruled out over-fitting as explanation of strong performance by using nested cross-validation and by establishing that substitution of actual gene values with random ones yields poor models. Our experiments support the conclusion that extensive gene redundancy and classifier characteristics are the most plausible explanation for the strong model performance. This is a reasonable

Classifier	Task I. Metastatic (7) – Nonmetastatic (132)	Task II. Cancer (186)-Normal (17)	Task III. Adenocarcinomas (139) - Squamous carcinomas (21)
SVMs	0.968 (139 cases)	0.996 (203 cases)	0.990 (160 cases)
KNN	0.926 (139 cases)	0.981 (203 cases)	0.976 (160 cases)

Figure 1: Area under the ROC curve with all genes and baseline setup.

Classifier	Task I. Metastatic (7) – Nonmetastatic (132)	Task II. Cancer (186)-Normal (17)	Task III. Adenocarcinomas (139) - Squamous carcinomas (21)
SVMs	0.584 (139 cases)	0.583 (203 cases)	0.572 (160 cases)
KNN	0.581 (139 cases)	0.522 (203 cases)	0.559 (160 cases)

Figure 2: Area under the ROC curve with random predictor values.

Classifier	Task I. Metastatic (7) – Nonmetastatic (132)	Task II. Cancer (186)-Normal (17)	Task III. Adenocarcinomas (139) - Squamous carcinomas (21)
SVMs	0.982 (40 cases), 0,982 (80 cases), 0.969 (120 cases)	1 (40 cases), 1 (80 cases), 1 (120 cases), 0.995 (160 cases)	0.981 (40 cases), 0.988 (80 cases), 0.980 (120 cases)
KNN	0.893 (40 cases), 0,832 (80 cases), 0.925 (120 cases)	1 (40 cases), 1 (80 cases), 0.993 (120 cases), 0.970 (160 cases)	0.916 (40 cases), 0.960 (80 cases), 0.965 (120 cases)

Figure 3: Area under the ROC curve when varying sample size.

Classifier	Task I. Metastatic (7) – Nonmetastatic (132)	Task II. Cancer (186)-Normal (17)	Task III. Adenocarcinomas (139) - Squamous carcinomas (21)
SVMs	0.944 (500 genes), 0.948 (1000 genes), 0.956 (5000 genes)	0.991 (500 genes), 0.989 (1000 genes), 0.995 (5000 genes)	0.982 (500 genes), 0.987 (1000 genes), 0.990 (5000 genes)
KNN	0.893 (500 genes), 0.893 (1000 genes), 0.941 (5000 genes)	0.959 (500 genes), 0.961 (1000 genes), 0.984 (5000 genes)	0.928 (500 genes), 0.955 (1000 genes), 0.965 (5000 genes)

Figure 4: Area under the ROC curve with random gene set selection of varying size.

Classifier	Task I. Metastatic (7) –Nonmetastatic (132)	Task II. Cancer (186)-Normal (17)	Task III. Adenocarcinomas (139) - Squamous carcinomas (21)
SVMs	0.915 (30/70), 0.938 (43/57), 0.954 (57/43), 0.962 (70/30), 0.968 (85/15)	0.997 (40/60), 0.996 (60/40), 0.996 (80/20)	0.989 (40/60), 0.990 (60/40), 0.990 (80/20)
KNN	0.782 (30/70), 0.833 (43/57), 0.866 (57/43), 0.901 (70/30), 0.990 (85/15)	0.960 (40/60), 0.962 (60/40), 0.976 (80/20)	0.960 (40/60), 0.962 (60/40), 0.976 (80/20)

Figure 5: Area under the ROC curve when varying train-test sample ratio.

Classifier	Task I. Metastatic (7) –Nonmetastatic (132)	Task II. Cancer (186)-Normal (17)	Task III. Adenocarcinomas (139) - Squamous carcinomas (21)
SVMs	0.890 (40 cases), 0.993 (80 cases), 0.965 (120 cases)	1 (40 cases), 1 (80 cases), 1 (120 cases), 0.995 (160 cases)	1 (40 cases), 1 (80 cases), 0.985 (120 cases)
KNN	0.918 (40 cases), 0.849 (80 cases), 0.80 (120 cases)	1 (40 cases), 0.96 (80 cases), 0.972 (120 cases), 0.982 (160 cases)	0.992 (40 cases), 0.960 (80 cases), 0.990 (120 cases)

Figure 6: Area under the ROC curve with alternative strategy for constructing cross-validation splits (i.e., use of all rare-category instances).

assumption in this domain, given that (a) manufacturers built chips with highly redundant and replicated gene probe sets, and that (b) intra-chip noise is not uniformly distributed [10]. Our experiments also suggest that sample size has relatively small effects (within the range examined); and that cross-validation design, train-test sample ratio, linear bias in the classifier (as long as kernel classifiers are used), and the choice of sample construction method appear to have small to negligible effects.

Another important factor that may account for good performance (but that not studied here) is that although the number

of samples measured in number of subjects is small relative to the number of variables (gene probes), the number of samples *measured in terms of aggregated cells per subject* is very large (i.e., up to several million cells per array). Thus within-patient intra-cell sampling variance will be minimized in array experiments from cell aggregates (while other sources of sampling variance such as sampling variation due to different cell environment from patient to patient will still be high due to the small number of patients). For example, sampling variation due to lack of cell-cycle synchronization will tend to be mitigated by the aggregation of cell measurements. We conclude with the remark that, in general, the dangers of overfitting/small sample are very real in high-dimensional datasets (of which array experiments are a prime example). Hence there is a (currently unmet) need for a *commonly accepted minimal set of sensitivity analyses and other best-of-practice guidelines* to safeguard against the possibility of performance overestimation in classification models. Methods such as the ones reported here may serve as a first step toward this direction. Further investigation in the context of many tasks/datasets is needed before such standards emerge, however.

5 Acknowledgments

Support for this research was provided in part by NIH grant LM 007613-01. Dr. Massion was supported by a grant from the American Lung Association.

6 References

[1] Aliferis C.F., et al. Machine Learning Models For Classification Of Lung Cancer and Selection of Genomic Markers Using Array Gene Expression Data (to appear in: FLAIRS 2003, special track AI in Medicine).

[2] Aliferis CF, Hardin D, Massion PP. Machine learning models for lung cancer classification using array comparative genomic hybridization. Proc AMIA Symp. 2002;7-11.

[3] Bhattacharjee, A., et al., Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses. Proc Natl Acad Sci USA, 2001. 98(24): 13790-5.

[4] Chang C.C., Lin, C.J, LIBSVM: a library for support vector machines (version 2.3). National Taiwan University.

[5] DeLong E., et al. Comparing the Areas under Two or More Correlated Receiver Operating Characteristic Curves: A Nonparametric Approach, Biometrics 44, 837-845, 1998 .

[6] Demuth, H. and M. Beale, Neural network toolbox user's guide. Matlab user's guide. 2001: The MathWorks Inc.

[7] Domingos P. Occam's two razors: the sharp and the blunt. In Proc. 4th Int Conf Knowledge Discovery and Data Mining, pages 37--43. AAAI Press, 1998.

[8] Duda, R.O., P.E. Hart, and D.G. Stork, Pattern Classification. 2001: John Wiley and Sons.

[9] Hagan, M.T., H.B. Demuth, and M.H. Beale, Neural network design. PWS Publishing; 1996.

[10] Kohane I.S. et al. Microarrays For An Integrative Genomics, MIT Press, 2000.

[11] Pagano M. et al. Principles of Biostatistics, Duxbury Thompson Learning, 2000.

[12] Platt J, Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines, Microsoft Research Technical Report MSR-TR-98-14, (1998).

[13] Provost F., T. Fawcett, and R. Kohavi. The case against accuracy estimation for comparing induction algorithms. In Proc. Fifteenth Intl. Conf. Machine Learning,

[14] Scholkopf B. et al. Advances in kernel methods: support vector learning. MIT Press; 1999.

[15] Schwarzer G, Vach W, On the misuses of artificial neural networks for prognostic and diagnostic classification in oncology. Stat Med. 2000 Feb 29;19(4):541-61.