Feature selection methods for SVM classification of microarray data

Mike Love

December 11, 2009

SVMs for microarray classification tasks

Linear support vector machines have been used in microarray experiments to predict a certain class of a sample tissue (e.g. healthy or tumor) using the mRNA expression of different genes. In measuring tens of thousands of genes, a microarray dataset includes many genes that are uninformative with respect to the classes. Finding subsets of genes for the SVM to train on could help improve the generalization of the algorithm and reveal a small set of relevant genes that could be used to build a cheaper diagnostic test.

Feature selection algorithms

I compare three methods of feature selection against running a linear SVM on the full dataset, on simulated data and an open microarray dataset.

Golub (1999) describes a weighted voting method with filter feature selection. The algorithm takes a certain number of genes that show the most extreme measure of a weight representing correlation between genes and the class labels. The measure of weight for gene i is:

$$c_i = \frac{\mu_i^+ - \mu_i^-}{\sigma_i^+ + \sigma_i^-}$$

where μ_i^+ is the mean of gene i for the positive class in the training data and σ_i^+ is the standard deviation of gene i for the negative class in the training data. A certain number of genes with the highest $|c_i|$ then contribute a vote to the total:

$$v_i = c_i(x_i - \frac{1}{2}(\mu_i^+ + \mu_i^-))$$

Here v_i is the vote from gene i and x_i is the value of gene i for a test case. The votes are then summed and if the total is positive then a positive class label is predicted. This method is similar to but not the same as a SVM algorithm.

Guyon (2002) and Zhang (2006) describe methods of backward search feature selection. Both methods involve starting with the full set of genes, picking a set of decreasing feature subset sizes, and eliminating batches of genes at each iteration by ranking their contribution. The measure of contribution in Guyon is w_i^2, the squared value of the weight vector for gene i from running SVM.

The measure in Zhang is $w_i(\mu_i^+ - \mu_i^-)$. The final subset size is chosen by k-fold cross-validation, which includes the feature selection steps as well as training the SVM. If multiple sizes tied for minimum CV error, I choose the one with more features, as choosing too few features can increase the error much more steeply than choosing too many. The final set of genes is chosen by the most frequently used genes at the step with the smallest CV error.

Simulated microarray data

I construct various sets of simulated data, using methods described in Zhang (2006). A set of informative genes are sampled from $N(\pm 0.25, 1)$ with the sign depending on which class the observation is from. A set of uninformative genes are sampled from $N(0, 1)$. To simulate outliers, some percent of the expression values for each sample are drawn from a Gaussian with the appropriate mean and 10 times the standard deviation. A training set with 100 observations and a test set with 1000 observations were created using this same procedure. 20 simulations were run for each setting of the parameters to assess variance. I varied the ratio of informative to uninformative genes, and the ratio of positive to negative classes in the training data.

Simulation results

I use the SMO algorithm presented in class with C=1, tolerance = 0.001 and max_passes = 10. For the method presented in Golub, I set the number of features to filter to 500. For the two backward search methods, I stepped through the subset sizes: [2000, 1500, 1000, 750, 500, 400, 300, 200, 100].

Here are results for different parameter settings. The final column indicates if the feature selection methods had test error with mean significantly different than the mean test error of the full SVM (two-tailed t-test at level $\alpha = .05$).

100 informative genes, 1900 uninformative genes, 1% outliers, balanced classes:

method	CV error (%)	test error (%)	features kept	number of SV	significant
full SVM	22.9 ± 5.2	23.8 ± 1.9	2000	80	NA
Golub	20.4 ± 5.8	23.1 ± 1.9	500	NA	
Guyon	18.0 ± 4.8	20.5 ± 2.9	100	44	*
Zhang	18.2 ± 5.1	20.8 ± 2.8	200	63	*

200 informative genes, 1800 uninformative genes, 1% outliers, balanced classes:

method	CV error (%)	test error (%)	features kept	number of SV	significant
full SVM	7.1 ± 2.9	9.6 ± 0.7	2000	80	NA
Golub	5.5 ± 2.6	9.7 ± 1.4	500	NA	
Guyon	5.4 ± 2.7	8.7 ± 1.2	500	78	*
Zhang	5.2 ± 2.4	8.8 ± 1.0	400	78	*

300 informative genes, 1700 uninformative genes, 1% outliers, balanced classes:

method	CV error (%)	test error (%)	features kept	number of SV	significant
full SVM	2.1 ± 1.2	3.4 ± 0.7	2000	80	NA
Golub	2.3 ± 1.5	3.3 ± 0.6	500	NA	
Guyon	1.5 ± 1.5	3.3 ± 0.6	500	77	
Zhang	1.7 ± 1.4	3.5 ± 1.1	500	77	

200 informative genes, 1800 uninformative genes, 1% outliers, unbalanced classes (25% positive, 75% negative):

method	CV error (%)	test error (%)	features kept	number of SV	significant
full SVM	22.9 ± 1.5	21.0 ± 1.0	2000	80	NA
Golub	18.5 ± 3.1	19.5 ± 2.2	500	NA	*
Guyon	16.7 ± 3.1	14.9 ± 2.3	100	37	*
Zhang	16.8 ± 3.1	14.9 ± 2.7	100	36	*

Microarray data

To test the methods on real microarray data, I used an openly available dataset published by Alon (1999). The data include 2000 of the genes with the largest minimal intensity across 40 tumor and 22 normal colon samples. The number of samples is not large enough to have a training set and a separate test set, so instead I compared the methods using out-of-bootstrap error. For each method, I trained the SVM on a bootstrap sample from the 62 observations, and tested on the observations that did not appear in the bootstrap sample. This was repeated for 20 bootstrap samples. As with cross-validation in the simulations, the feature selection loops were embedded within the bootstrapping loop.

method	out-of-boot error (%)	features kept	number of SV	significant
full SVM	19.5 ± 5.8	2000	39	NA
Golub	33.8 ± 14.7	500	NA	* (worse)
Golub	21.1 ± 11.1	20	NA	
Guyon	18.3 ± 6.7	1500	41	
Zhang	18.3 ± 7.9	400	33	

Conclusions

For the simulation data, the backwards search feature selection methods have statistically significant reduction in cross-validation and test error when there are few informative genes, although the differences are not large for the balanced class data. This agrees with the conclusion from Nilsson (2006) that the SVM performs well even with many uninformative features, making large decreases in predictive error through feature selection hard to achieve. All three feature selection algorithms gave significant improvements with the unbalanced simulated data, and the reduction in error was large for backwards search methods (29% decrease in test error).

The two backwards search methods performed very similarly (both better than the method from

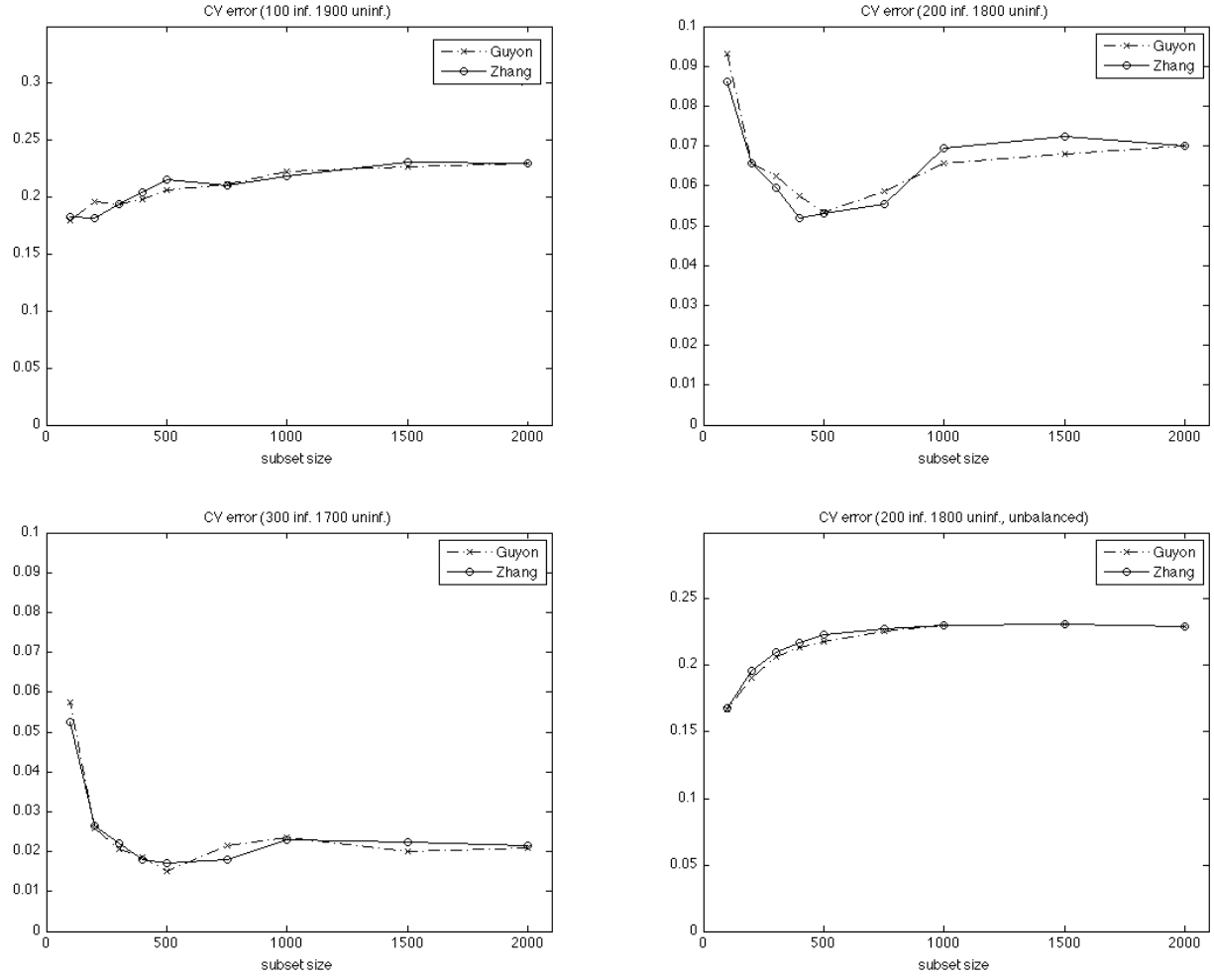
Golub) on the simulated datasets. The number of support vectors always decreased as the number of features was reduced in the backwards search methods, which might imply better generalization results.

For the actual microarray data, the two backwards search methods had similar error to the full SVM, but in this case the Golub method performed worse. I set the Golub method to pick only 20 genes after trying on 500 genes and having high prediction error. The minimum number selected for the Guyon and Zhang methods can be arbitrary in cases like this when the error curve is fairly flat over the various subset sizes.

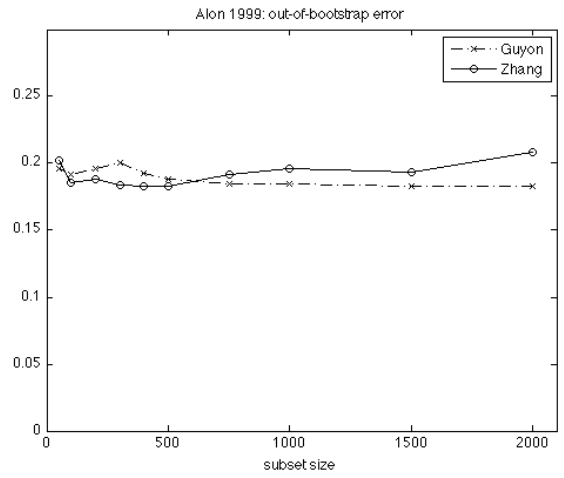
The plots of cross-validation error indicate that the predictive errors often increase more steeply with the loss of any informative features than they do with the gain of uninformative features. These plots might provide, for empirical data, a rough sense of the number of "relevant" genes for a certain contrast of sample classes.

Figures

Plots of CV error from the simulated data are the averages over 20 simulations.



Plot of out-of-bootstrap error for the Alon (1999) data is the average over 20 bootstrap samples.



References

1. Alon, U., Barkai, N., Notterman, D.A., Gish, K., Ybarra, S., Mack, D., Levine, A.J.: Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. Procedings of the National Academy of Sciences 96 (1999) 6745-6750
2. Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J., Coller, H., Loh, M., Downing, J., Caligiuri, M., Bloomfield, C., Lander, E.S.: Molecular classifiation of cancer: class discovery and class prediction by gene expression monitoring. Science 286 (1999) 531-537
3. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. Machine Learning 46 (2002) 389-422
4. Nilsson, R., Pena, J.M., Bjorkegren, J., Tegner, J.: Evaluating Feature Selection for SVMs in High Dimensions. Springer Berlin / Heidelberg (2006)
5. Zhang, X., Lu, X., Shi, Q., Xu, X., Leung, H.E., Harris, L.N., Iglehart, J.D., Miron, A., Liu, J.S., Wong, W.H.: Recursive SVM feature selection and sample classification for mass-spectrometry and microarray data. BMC Bioinformatics 7 (2006)