“What Was It That Made Generalized Validation Cross?” or, a brief history of tuning, and illustrated with an application to the LASSO-Patternsearch Algorithm.

Title courtesy of S. Stigler

Grace Wahba

LASSO-Patternsearch Algorithm based on joint work with Weiliang Shi, Stephen Wright, Kristine Lee, Ronald Klein and Barbara Klein

The University of Chicago June 7, 2007

These slides at

`http://www.stat.wisc.edu/~wahba/` → TALKS

Papers/preprints at

`http://www.stat.wisc.edu/~wahba/` $->$ TRLIST

Abstract

We begin with a few historical remarks about what might be called the regularization class of statistical model building methods, which include penalized likelihood, support vector machines, robust and quantile nonparametric regression, etc., etc, and the problem of tuning them, spending a little too much time on methods related to Generalized Cross Validation. After that we discuss an approach to variable and pattern selection given very large attribute vectors, based on the LASSO (that is, l_1 penalties) that differs from most approaches to this problem in that it is a mostly global, rather than a sequential, or greedy algorithm, for finding patterns in the data that most influence an outcome.

Regularization Class of Statistical Models

- $y \in \mathcal{Y}$: The observations, $y_1, \cdots, y_n$.
- $x \in \mathcal{X}$: The attribute vectors, $x(1), \cdots, x(n)$.
- $f \in \mathcal{H}$: The model, to be found, relates $x \in \mathcal{X}$ to $y \in \mathcal{Y}$. $\mathcal{H}$ is the class in which f is to be found.
- $\mathcal{C}(y, f)$: Measures goodness of fit of the model to the data.
- $J_\lambda(f)$: Penalty functional on f, constrains complexity/degrees of freedom of the model.

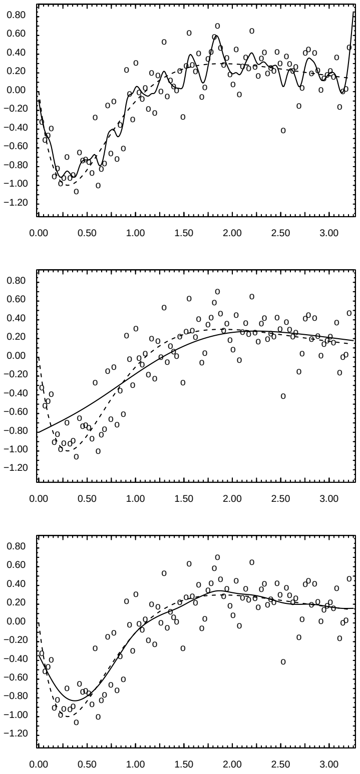
The model f is found as the solution to: min $f \in \mathcal{H}$:

$$\sum_{i=1}^{n} \mathcal{C}(y_i, f(x(i)) + J_\lambda(f).$$

The (set of) parameter(s) λ controls the tradeoff between fit and complexity, a. k. a bias-variance in some contexts.

One simple example leads to the cubic smoothing spline.

- $y \in R$
- $x \in [0, 1]$
- $f \in W_2^2$ (Sobolev space of functions with square integrable second derivative),
- $\mathcal{C}(y, f) = (y - f(x))^2$
- $J_\lambda(f) = \int_0^1 (f''(x))^2 dx$



On the right: Top: λ too small; Middle λ too big; Bottom λ just right, chosen by Generalized Cross Validation *GCV*. (Golub, Heath and Wahba, 1979, Craven and Wahba, 1979).

Varieties of Cost Functions (Univariate Case).

	$\mathcal{C}(y,f)$
Regression:	
Gaussian data	$(y-f)^2$
Bernoulli, $f = log[p/(1-p)]$	$-yf + log(1+e^f)$
Other exponential families	other log likelihoods
Data with outliers	robust functionals
Quantile functionals	$\rho_q(y-f), \rho_q(\tau) = \tau(q - I(\tau \leq 0))$
Classification: $y \in \{-1, 1\}$	
Support vector machines	$(1-yf)_+$, $(\tau)_+ = \tau, \tau \geq 0, 0$ otherwise
Other "large margin classifiers"	e^{-yf} and other functions of yf

Multivariate (vector-valued y) versions of the above.

Penalty Functionals

	$J_\lambda(y, f)$
Quadratic (RKHS) Penalties: $x \in \mathcal{T}$, some domain, can be very general. $f \in \mathcal{H}_\mathcal{K}$, a reproducing kernel Hilbert space of functions, characterized by some positive definite function $K(s,t)$, $s, t \in \mathcal{T}$.	$\lambda \|f\|^2_{\mathcal{H}_\mathcal{K}}$, etc.
l_p **Penalties:** $x \in \mathcal{T}$, some domain, can be very general. $f \in$ span $\{B_r(x), r = 1, \cdots, N\}$, a specified set of basis functions on $\mathcal{T}$. $f(x) = \sum_{r=1}^N c_r B_r(x)$	$\lambda \sum_{r=1}^N \|c_r\|^p$

Various combinations of RKHS and ℓ_p penalties are possible.

$\mathcal{T}$:

- Anything you can define a positive definite function K on.
- Anything you can define a set of basis functions $\{B_r\}$ on.
- Many generalities: $x = (x_1 : x_2), \quad x_1 \in \mathcal{T}^1, x_2 \in \mathcal{T}^2$, $f = f_1 + f_2 \qquad J_\lambda(f) = J^1_{\lambda_1}(f_1) + J^2_{\lambda_2}(f_2)$, etc., etc., etc....

Some Tuning References

Not complete. May not be the earliest reference. Not guaranteed.

- Unbiased Risk C. Mallows. Some comments on C_p. *Technometrics*, 15:661–675, 1973.
- AIC H. Akaike. A new look at the statistical identification model. *IEEE Trans. Auto. Control*, 19:716–723, 1974.
- Leaving-out-one G. Wahba and S. Wold. A completely automatic French curve. *Commun. Stat.*, 4:1–17, 1975.
- GCV-illposed G. Wahba. Practical approximate solutions to linear operator equations when the data are noisy. *SIAM J. Numer. Anal.*, 14:651–667, 1977.
- Unbiased Risk M. Hudson. A natural identity for exponential families with applications in multiparameter estimation. *Ann. Statist.*, 6:473–484, 1978.

- BIC G. Schwartz. Estimating the dimension of a model. *Ann. Statist.*, 6:461–464, 1978.
- GCV G. Golub, M. Heath, and G. Wahba. Generalized cross validation as a method for choosing a good ridge parameter. *Technometrics*, 21:215–224, 1979.
- GCV P. Craven and G. Wahba. Smoothing noisy data with spline functions: estimating the correct degree of smoothing by the method of generalized cross-validation. *Numer. Math.*, 31:377–403, 1979.
- G. Wahba. A comparison of GCV and GML for choosing the smoothing parameter in the generalized spline smoothing problem. *Ann. Statist.*, 13:1378–1402, 1985.
- Randomized Trace D. Girard. A fast ‘Monte-Carlo cross-validation’ procedure for large least squares problems with noisy data. *Numer. Math.*, 56:1–23, 1989.

- Randomized Trace M. Hutchinson. A stochastic estimator for the trace of the influence matrix for Laplacian smoothing splines. *Commun. Statist.-Simula.*, 18:1059–1076, 1989.

- GACV D. Xiang and G. Wahba. A generalized approximate cross validation for smoothing splines with non-Gaussian data. *Statistica Sinica*, 6:675–692, 1996.

- GACV-multiple outcomes X. Lin. Smoothing spline analysis of variance for polychotomous response data. Technical Report 1003, PhD thesis, Department of Statistics, University of Wisconsin, Madison WI, 1998. Available via G. Wahba's website.

- SVM T. Joachims. Estimating the generalization performance of an SVM efficiently. In *Proceedings of the International Conference on Machine Learning*, San Francisco, 2000. Morgan Kaufman.

- GACV-SVM G. Wahba, Y. Lin, and H. Zhang. Generalized approximate cross validation for support vector machines. In A. Smola, P. Bartlett, B. Scholkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 297–311. MIT Press, 2000.

- GACV-clustered outcomes F. Gao, G. Wahba, R. Klein, and B. Klein. Smoothing spline ANOVA for multivariate Bernoulli observations, with applications to ophthalmology data, with discussion. *J. Amer. Statist. Assoc.*, 96:127–160, 2001.

- SVM O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46:131–159, 2002.

- GACV-l_1 H. Zhang, G. Wahba, Y. Lin, M. Voelker, M. Ferris, R. Klein, and B. Klein. Variable selection and model building via likelihood basis pursuit. *J. Amer. Statist. Assoc.*, 99:659–672, 2004.

- GACV-multicat-SVM Y. Lee, Y. Lin, and G. Wahba. Multicategory support vector machines, theory, and application to the classification of microarray data and satellite radiance data. *J. Amer. Statist. Assoc.*, 99:67–81, 2004.
- B.Efron. The estimation of prediction error: Covariance penalties and cross-validation. *J. Amer. Statist. Assoc.*, 81:619–642. (with discussion), 2005.
- GACV-l_1,BGACV W. Shi, G. Wahba, S. Wright, K. Lee, R. Klein, and B. Klein. LASSO-Patternsearch algorithm with application to ophthalmalogy data. Technical Report 1131, Department of Statistics, University of Wisconsin, Madison WI, 2006.
- M. Yuan. GACV for quantile smoothing splines. *Comp. Stat. Data Anal.*, 50:813–829, 2006.

The LASSO-Patternsearch Algorithm:

Data $\{y_i, x(i)\}, i = 1, \cdots, n$

y_i - response of ith subject with p attributes:
$x(i) = (x_1(i), x_2(i), \cdots, x_p(i))$

$y_i, x_1(i), x_2(i), \cdots, x_p(i) \in \{0, 1\}^{p+1}$. (special case and p large)

Define $p(x) = Prob(y = 1|x)$
$f(x) = log(p(x)/(1 - p(x)))$ [*the logit a.k.a log odds ratio*].
$p(x) = e^{f(x)}/(1 + e^{f(x)})$.

The negative log likelihood $\mathcal{C}(y, f)$ for n observations is

$$\mathcal{C}(y, f) = \sum_{i=1}^{n} -y_i f(x(i)) + b(f(x(i)), \quad b(f) = log(1 + e^f).$$

The logit $f(x)$ is modeled as

$$f(x) = \sum_{\ell=0}^{N} c_\ell B_\ell(x)$$

where the B_ℓ are given basis functions. The B_ℓ depend on $x = x_1, ..., x_p$ in some specified way. The $\{c_\ell\}$ are found by minimizing

$$\mathcal{C}(y, f) + J_\lambda(f)$$

where

$$J_\lambda(f) = \lambda \sum_{\ell=1}^{N} |c_\ell| \qquad l_1 \ \ penalty.$$

LASSO-Patternsearch involves a large to very large number N of basis functions. The ℓ_1 penalty $\sum_{\ell=1}^{N}|c_\ell|$ has the property that many smaller c_ℓ will be set to 0, depending on λ. For given λ, the c_ℓ can be found numerically in MATLAB for moderate size N, but that will not be good enough for our purposes. For the LASSO-Patternsearch the basis functions will be all products of the x_r up to order q:

$$B_{j_1,j_2,..,j_r}(x) = \prod x_{j_1}x_{j_2}...x_{j_r}, r = 1, \cdots, q.$$

Thus, $B_{j_1,j_2,\ldots,j_r}(x) = 1$ if x is a p-vector which has ones in each of the $j_1, j_2, \cdots, j_r$ positions, and $B_{j_1,\ldots,j_r}(x) = 0$ otherwise. The number N of basis functions is then

$$N = \binom{p}{0} + \binom{p}{1} + \binom{p}{2} + ... + \binom{p}{q}.$$

For $q = p$, (all possible patterns), $N = 2^p$.

Note that the conditional distribution of one Bernoulli random variable y given p other Bernoulli random variables $x_1, \cdots, x_p$ has 2^p paramteters and can be expanded in complete generality in these basis functions. The representation will be most compact, however, if all the risky variables are coded with the risky direction as 1.

A special purpose algorithm which can handle N up to 4000 on our 3.4 GHz cpu and 4Gb memory workstation is in Shi *et. al.* (The data analysis described later took just 5 seconds.)

How to choose λ?

We will target the Kullback-Liebler (KL) distance between two distributions η and η_λ where η is the true distribution and η_λ is an estimated distribution:

$$KL(\eta, \eta_\lambda) = E_\eta log \frac{\eta}{\eta_\lambda}.$$

For example, in the Gaussian case, suppose $\eta \sim \mathcal{N}(\mu, 1)$ and $\eta_\lambda \sim \mathcal{N}(\mu_\lambda, 1)$, then the KL distance $KL(\eta, \eta_\lambda) = \frac{1}{2}(\mu - \mu_\lambda)^2$, leading to minimizing predictive mean square error as the target.

In the Bernoulli case we use the $GACV$ to choose λ, which targets the KL distance for members of an exponential family with no nusiance parameters. $GACV$ is derived starting with the Comparative KL distance (CKL), which ignores that part of KL distance not depending on λ. The result for Bernoulli data is next.

Notation: $f(x(i)) \rightarrow f_i$; $f_\lambda(x(i) \rightarrow f_{i\lambda}$; $p(x(i) \rightarrow p_i$; $p_{i\lambda}(1 - p_{i\lambda}) \rightarrow \sigma^2_{i\lambda}$. $H = \{h_{ij}\}$ is inverse Hessian. (like the influence matrix in the Gaussian case) with iith entry h_{ii}, $W = diag(\sigma_{i\lambda})$.

$$\begin{aligned}
KL(f, f_\lambda) &= \sum_{i=1}^{n} E_f log[\frac{\mathcal{L}(f_i)}{\mathcal{L}(f_{i\lambda})}] \\
CKL(f, f_\lambda) &= \sum_{i=1}^{n} [-p_i f_{i\lambda} + b(f_{i\lambda})] \\
CV(\lambda) &= \sum_{i=1}^{n} [-y_i f_{i\lambda}^{[-i]} + b(f_{i\lambda})] \\
ACV(\lambda) &= \sum_{i=1}^{n} [-y_i f_{i\lambda} + b(f_{i\lambda})] + \sum_{i=1}^{n} [\frac{y_i(y_i - \mu_{i\lambda})}{(1 - \sigma^2_{i\lambda} h_{ii})}] h_{ii} \\
GACV(\lambda) &= \sum_{i=1}^{n} [-y_i f_{i\lambda} + b(f_{i\lambda})] + tr H \frac{\sum_{i=1}^{n} y_i(y_i - \mu_{i\lambda})}{tr[I - (W^{1/2} H W^{1/2})]}.
\end{aligned}$$

Notation: $f(x(i)) \rightarrow f_i$; $f_\lambda(x(i) \rightarrow f_{i\lambda}$; $p(x(i) \rightarrow p_i$; $p_{i\lambda}(1-p_{i\lambda}) \rightarrow \sigma^2_{i\lambda}$. $W = diag(\sigma_{i\lambda})$. $H = \{h_{ij}\}$ is inverse Hessian, Here $H = B_*(B_*'WB_*)^{-1}B_*'$, B_* is the $n \times N_{B0}$ design matrix for the N_{B0} non-zero c_ℓ in the model.

$$\begin{aligned} ACV(\lambda) &= \sum_{i=1}^{n}[-y_i f_{i\lambda} + b(f_{i\lambda})] + \sum_{i=1}^{n}[\frac{y_i(y_i-\mu_{i\lambda})}{(1-\sigma^2_{i\lambda}h_{ii})}]h_{ii} \\ GACV(\lambda) &= \sum_{i=1}^{n}[-y_i f_{i\lambda} + b(f_{i\lambda})] + trH\frac{\sum_{i=1}^{n} y_i(y_i-\mu_{i\lambda})}{tr[I-(W^{1/2}HW^{1/2})]}. \\ GACV(\lambda) &= \sum_{i=1}^{n}[-y_i f_{\lambda i} + \log(1+e^{f_{\lambda i}})] + trH\frac{\sum_{i=1}^{n} y_i(y_i-p_{\lambda i})}{n(1-\frac{1}{n}N_{B0})}, \end{aligned}$$

The LASSO-Patternsearch has the following steps:

Step 1. Minimize $\mathcal{C}(y, f) + \lambda \sum_{\ell=1}^{N} |c_\ell|$, choose λ by $GACV$.

Step 2. Enter all basis functions with $\ell : |c_\ell| > 0$ into a parametric logistic regression model:

$$f(x) = \sum_{\ell : c_\ell > 0} a_\ell B_\ell(x)$$

and fit.

Step 3. Select all ℓ for which a_ℓ are significant at the $q\%$ level, to fit the final model:

$$f(x) = \sum_{\ell : a_\ell \quad significant} b_\ell B_\ell(x).$$

q is another tuning parmeter, chose q by $BGACV$. (What is BGACV?) Examine patterns (B_ℓ)'s with significant b_ℓ's.

Step 4. Interpret, demonstrate "significance after data mining".

What is BGACV?

$$\begin{aligned} GACV(\lambda) &= \sum_{i=1}^{n}[-y_i f_{i\lambda} + b(f_{i\lambda})] + trH\frac{\sum_{i=1}^{n} y_i(y_i-\mu_{i\lambda})}{tr[I-(W^{1/2}HW^{1/2})]}. \\ BGACV(\lambda) &= \sum_{i=1}^{n}[-y_i f_{i\lambda} + b(f_{i\lambda})] + \frac{\log n}{2}trH\frac{\sum_{i=1}^{n} y_i(y_i-\mu_{i\lambda})}{tr[I-(W^{1/2}HW^{1/2})]}. \end{aligned}$$

$GACV$ targets KL distance. Not the same as as selecting the ‘true’ model. $GACV$ appears to be conservative in that it almost never misses a ‘true’ pattern (basis function), at the expense of including noise patterns. Similar argument: replace AIC with BIC by replacing a 2 by $\log n$ in front of the degrees of freedom. As a heuristic, considering loss of a ‘true’ pattern worse than inclusion of a noise pattern, while trying to have it both ways, a BGACV criteria is employed as a second stage. Interesting theoretical issues here.

Application to progression of myopia from the Beaver Dam Eye Study, BDES 1 to BDES2, $n = 876$ records of persons aged 60-69 at BDES1. A person whose 'worse eye' scored at a decrease of .75 Diopters or more is labeled $y = 1$, and 0 otherwise. About 13% of this group was scored $y = 1$. $p = 7$ predictor variables.

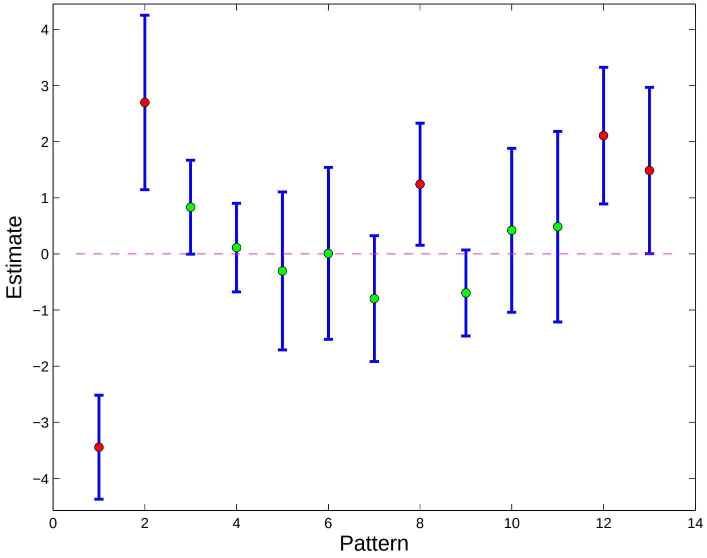
Table 1: Trial Variables and Cutpoints

	variable	description	binary cut point (higher risk) $X = 1$)
X_1	sex	sex	Male
X_2	inc	income	< 30
X_3	jomyop	juvenile myopia	< 21
X_4	catct	cataract	4-5
X_5	pky	packyear	>30
X_6	asa	aspirin	not taking
X_7	vtm	vitamin	not taking

Step 1. $p = 7$ variables, $q = p = 7$, $N = 2^7$ basis functions, minimize $\mathcal{C}(y, f) + \lambda \sum_{\ell=1}^{128} |c_\ell|$, choose λ by GACV.

Twelve patterns survived Step 1.

Step2. Parametric logistic regression for patterns surviving Step 1: Enter the patterns surviving Step 1 into a parametric logistic regression model:



The result for the 12 patterns is above: Confidence intervals depicted reflect Step 3

Step 3. Select all ℓ for which a_ℓ are significant at the $q\% = 96.92\%(BGACV)$ level, to fit the final model. The patterns passing this test are:

1. Constant
2. catct (Cataract)
8. pky vtm (Packyear $>$ 30 and not taking vitamins)
12. sex inc jomyop asa (Male, low income, juvenile myopia, not taking aspirin)
13. sex inc catct asa (Male, low income, cataract, not taking aspirin)

Step 3.(continued) Fit the final model with the five patterns significant at the 96.92% (BGACV) level.

$$f(x) = \sum_{\ell : a_\ell \quad significant} b_\ell B_\ell(x).$$

The (refitted) model is

$$f(catct, pky, vtm, sex, inc, jomyop, asa)$$

$$-3.29 + 2.42 * cact + 1.18 * pky * vtm$$

$$+ 1.84 * sex * inc * jomyop * asa + 1.08 * sex * inc * cat * asa.$$

Step 4. Having done some ”data mining”, the investigators can go back and look at classes of people who may not have been examined separately before. For example:

catct	pky	not take vitamins	risk of progression
1	1	1	$17/23 = 0.7391$
1	1	0	$7/14 = 0.5000$
0	1	1	$22/137 = 0.1606$
0	1	0	$2/49 = 0.0408$
1	0	1	$18/51 = 0.3529$
1	0	0	$19/36 = 0.5278$
0	0	1	$22/363 = 0.0606$
0	0	0	$13/203 = 0.0640$

Looking at the smokers: (1, 1, 1, 1):

Looking at the smokers: smokers with cataract are relatively protected by taking vitamins, and smokers without cataract are also relatively protected by taking vitamins. For non smokers taking or not taking vitamins makes no (significant) difference.

Physiologically meaningful - recent literature suggests:
a) Certain vitamins are good for eye health.
b) Smoking depletes the serum and tissue vitamin level, especially Vitamin C and Vitamin E.

(Although as usual, a "randomized controlled clinical trial would provide the best evidence of any effect of vitamins on progression of myopia in smokers")

To check on the “significance” of the patterns, randomly scramble the ys while keeping the x’s fixed, and apply the entire LASSO-Patternsearch algorithm to see how often false patterns are generated. Repeat 600 times. (Statistical theory is not clear on properties of multistep procedures)

Detection of noise patterns found in scrambled data compared to observed p values:

Log p values of the patterns found (out of 600) are plotted (l. to r. top to bottom) for observed patterns of size 1,2,3,4. Red lines are for the observed p-values for *catct*, *pky vtm*, none, and *sex inc jomyop asa* (lower) and *sex inc catct asa* (upper).

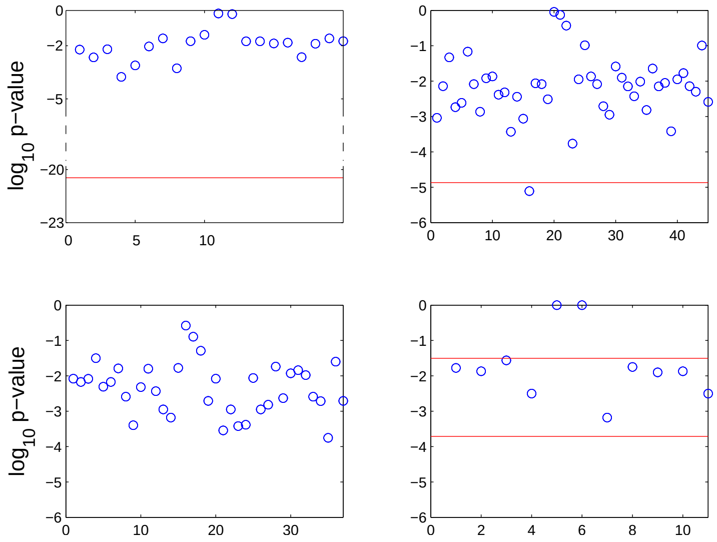


Figure 1: Upper red line suggests that *sex inc catct asa* is borderline significant.

Genetic Data (realistic simulation, not ours, not released yet)

y = phenotype, x = SNPs, alleles, covariates, after coding as 1 or 0, 9192 variables.

Train: 1500 cases, 2000 controls
Tune: 1500 cases, 2000 controls
Test: 1500 cases, 2000 controls.

Pre-screen step: 9192 variables reduced to $N = 2559$ basis functions for the LASSO step. Final model has 8 main effects and 3 interactions. Using $p = .5$ as a classifier, a competitive 12.6% error rate was obtained. Identified a SNP near most of the genes that were used to generate the data.