Mathematisches Institut Oberwolfach

# The LASSO-Patternsearch Algorithm: Finding “patterns in a haystack”

Grace Wahba

Joint work with Weiliang Shi, Steve Wright, Kristine Lee, Ronald Klein and Barbara Klein, to appear, Statistics and Its Interface (2008)

June 29-July 5, 2008

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

Papers/preprints including this one at
`http://www.stat.wisc.edu/~wahba/` $->$ TRLIST

## Abstract

We describe the LASSO-Patternsearch (LPS) algorithm, a two step procedure whose core applies a LASSO ($\ell_1$ penalized likelihood) to Bernoulli ($\{0,1\}$) response data $y$ given a very large attribute vector $x$ from a sparse multivariate Bernoulli distribution. Sparsity here means that the conditional distribution of $y$ given $x$ is assumed to have very few terms, but some may be of higher order (patterns). We propose the BGACV for Bernoulli data to tune the LASSO and a final parametric fit for variable selection,, which is based on a prior belief in sparsity. An algorithm which can handle a very large number ($2 \times 10^6$) of candidate terms in a global optimization scheme is given, and it is argued that global methods have a certain advantage over greedy methods in the variable selection problem. Applications to demographic and genetic data are described.

## Outline

1. The LASSO - $\ell_1$ penalties.
2. The multivariate Bernoulli distribution: Bernoulli response $y \in \{0,1\})$, Bernoulli attributes $x \in (\{0,1\},\cdots,\{0,1\})$.
3. Tuning: GACV and BGACV for Bernoulli responses.
4. LASSO-Patternsearch (LPS) core algorithm for global variable selection.
5. The post LASSO step.
6. Examples: demographic study, genetic data.
7. Summary and conclusions.

# The LASSO

Given $y_i, x(i)$, where $x(i) = (x_1(i), \cdots, x_p(i)$, we find $f = f(x)$ to minimize

$$I_\lambda(y, f) = \mathcal{L}(y, f) + \lambda J(f)$$

where $\mathcal{L}(y, f)$ is $\frac{1}{n}$ times the negative log likelihood of $y$ given $x$ (or some other measure of the fit of $y$ to $f$), $J(f)$ is a penalty functional on $f$ and $\lambda$ is a tuning parameter. If

$$f(x) = \mu + \sum c_\ell B_\ell(x),$$

for some specified basis functions, and the penalty functional is

$$J(f) = \sum_\ell |c_\ell|,$$

then the solution is called the LASSO (Tibshirani, 1996).(A similar idea was proposed by Chen, Donoho and Saunders, 1998).

Copious literature exists for the LASSO with Gaussian data

$$\mathcal{L}(y, f) = \sum (y_i - f(x(i))^2,$$

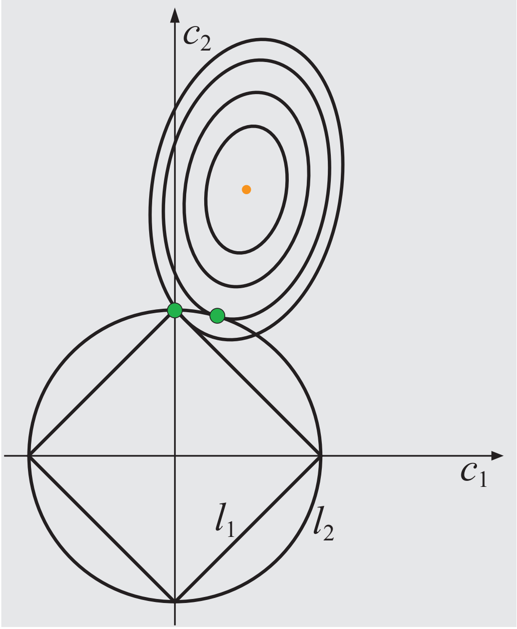
as well as for the support vector machine

$$\mathcal{L}(y, f) = \sum (1 - y_i f(x(i))_+.$$

However there are relatively few results for Bernoulli data, $y_i \in \{0, 1\}$, $f = \log[p/(1-p)]$ with the log likelihood

$$\mathcal{L}(y, f) = \sum_{i=1}^{n} [-y_i f(x(i)) + \log(1 + e^{f(x(i))})],$$

specifically with respect to choosing the tuning parameter $\lambda$.



The $\ell_1$ penalty $\sum_\ell |c_\ell|$ is known to give a sparse representation - as $\lambda$ increases, an increasing number of the $c_\ell$ become 0. This is in contrast to the $\ell_2$ penalty $\sum_\ell c_\ell^2$, in which, typically the $c_\ell$ tend to all be non-zero. The choice of $\lambda$ is an important issue in practice.

# The Multivariate Bernoulli Distribution

Let $(x_0, x_1, \cdots, x_p)$ be a $p+1$ dimensional vector of possibly correlated Bernoulli random variables. The most general form $p(x_0, x_1, \cdots, x_p)$ of the joint density is (Whittaker 1990)

$$p(0,0,\cdots,0)^{[\pi_{j=0}^{p}(1-x_j)]} p(1,0,\cdots,0)^{[x_0 \pi_{j=1}^{p}(1-x_j)]} \ldots p(1,1,\cdots,1)^{[\pi_{j=0}^{p} x_j]}.$$

We can extract from this the conditional logit $f$, which appears in the likelihood

$$f(x) = \log[p(x)/(1-p(x))] = \log\left(\frac{prob(y=1|x)}{(1-prob(y=1|x)}\right)$$

as

$$f(x_1, \cdots x_p) = \log p(1, x_1, \cdots, x_p) - \log p(0,, x_1, \cdots, x_p).$$

After some reorganization the logit has the representation

$$f(x_1, \cdots, x_p) = \mu + \sum_{j=1}^{p} c_j B_j(x) + \sum_{1<j<p} c_{jk} B_{jk}(x) + \cdots + c_{12\ldots p} B_{12\ldots p}(x)$$

where $B_j(x) = x_j, B_{jk} = x_j x_k$, and so forth, and the $c$'s are parameters to be estimated. Note that $B_{j_1 j_2 \cdots j_r}$ is 1 if $x_{j_1}, x_{j_2}, \cdots, x_{j_r}$ are all 1 and 0 otherwise.

## Patterns

We will call $B_{j_1 j_2 .. j_r}(x)$ an $r$th order pattern. Let $q$ be the highest order considered. Then there will be $N_B = \sum_{\nu=0}^{q} \binom{p}{\nu}$ patterns. If $q = p$, there is a complete set of $N_B = 2^p$ patterns (including the constant function $\mu$), spanning all possible patterns. If $p = 7$, say then there will be $2^7 = 128$ patterns and standard software can be used to solve the Bernoulli LASSO problem. If $p = 1,000$ there are over half a million unknown coefficients just considering "main effects" and order 2 patterns, so special purpose software is required. To consider all possible patterns of order three requires preprocessing to reduce the number of candidates. In any case, however, the $\ell_1$ penalty, with $\lambda$ chosen with variable selection in mind, is ideally suited to find those (assumed) small number of patterns which influence $y$.

# Tuning

The Comparative Kullback-Liebler ($CKL$) distance: Letting $f$ and $f_\lambda$ be the true and estimated parameters of the distribution, then the $KL$ distance, defined as

$$KL(f, f_\lambda) = E_f \log\left(\frac{G(y, f)}{G(y, f_\lambda)}\right)$$

is a measure of how close the fitted distribution $G(y, f_\lambda)$ is to the true, but unknown distribution $G(y, f)$. $CKL(\lambda)$ is $KL(\lambda)$ minus any terms not dependent on $\lambda$. In the Gaussian case with known $\sigma^2$ $CKL(\lambda) = \sum \frac{1}{2\sigma^2}(f(x(i) - f_\lambda(x(i))^2$, a. k. a. $PMSE$. In the Bernoulli case

$$CKL(\lambda) = \sum -p(x(i))f_\lambda(x(i)) + log(1 + e^{f_\lambda(x(i))}).$$

where $p = e^f/(1 + e^f)$.

In the Gaussian case, when $y = f(x) + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$ with $\sigma^2$ known, there is the well known unbiassed risk estimate for $PMSE(\lambda)$ (Mallows(1973), Hudson(1978)). In Poisson, Gamma, Binomial with $m > 2$ and other distributions, unbiassed estimates of $CKL(\lambda)$ are well known, (Hudson(1978),Wong(2006)). In the Gaussian case with $\sigma^2$ unknown, an exact unbiassed estimate for $PMSE(\lambda)$ is not available but a good approximation to the minimizer is available ($GCV$, for example). Similarly, in the Bernoulli case Wong shows that there is no unbiassed estimator for $CKL(\lambda)$ but a good approximation for finding the minimum (GACV) has been given in Xiang and Wahba(1996) based on a leaving-out-one argument.

The $GACV$ was derived from a leaving-out-one argument on the $CKL$. In the LASSO-Patternsearch case it has a simple form:

$$GACV(\lambda) = \sum_i -y_i f_\lambda(x(i)) + log(1+e^{f_\lambda(x(i))}) + trH \frac{\sum_i y_i(y_i - p_\lambda(x(i)))}{(n - N_{B*})}$$

where $H = B*(B*'WB*)^{-1}B*'$ with $B*$ is the design matrix for the basis functions in the fitted model and $W$ is the $n \times n$ diagonal matrix with $ii$ th element the estimated variance $\sigma_\lambda(i)$. $N_{B*}$ is the number of non-zero coefficients in the fitted model.

# BGACV

It has long been understood in the Gaussian data-quadratic penalty case that tuning for $PMSE$ is not the same as tuning for variable selection, indeed, this is the difference between the well known $AIC$ (prediction) and $BIC$ (variable selection with prior belief in sparsity). (George(2000), Leng, Lin and Wahba(2006)). Where the $AIC$ has a 2 in front of the degrees of freedom, the $BIC$ has a $\log n$. For $\log n > 2$, $BIC$ gives a bigger $\lambda$. For the present work being specifically targeted to obtain sparse models, will will do the same thing, giving

$$BGACV(\lambda) = \sum_i -y_i f_\lambda(x(i)) + log(1 + e^{f_\lambda(x(i))}) + \frac{\log n}{2} trH \frac{\sum_i y_i(y_i - p_\lambda(x(i))}{(n - N_{B*})}$$

(Theorem?).

## Global LASSO Algorithm (thanks Steve Wright)

($2 \times 10^6$ unknowns.) The minimization problem has the form

$$\min_c I_\lambda(y, f) \equiv \min I_\lambda(c) := F(c) + \lambda \|c\|_1.$$

We describe a separable approximation algorithm with projected Newton acceleration. Generates a sequence of iterates $c^k$, $k = 1, 2, \ldots$.

At iteration $k$, form a linear model with damping, to solve for candidate step $d$:

$$\min_d F(c^k) + \nabla F(c^k)^T d + \frac{1}{2}\alpha_k d^T d + \lambda \|c^k + d\|_1.$$

Larger $\alpha_k \Rightarrow$ shorter step.

Can be solved trivially as it is separable in the components of $d$: $O(n)$ operations.

Get an estimate of the zero set $\mathcal{C}_k := \{i \mid c_i^k + d_i = 0\}$.

First-order step: keep doubling $\alpha_k$ until the step gives a decrease in $I_\lambda$, that is, $I_\lambda(c^k + d) < I_\lambda(c^k)$. Take this kind of step if $\mathcal{C}_k$ is large (more than 1000 elements, say).

Otherwise try to accelerate by taking a Newton step in the components that are not in $\mathcal{C}_k$ — the ones that are apparently nonzero at the solution.

Define the non-zero set $\mathcal{S}_k := \{1, 2, \ldots, n\} \setminus \mathcal{C}_k$ and define the projected Hessian

$$H_k := \left[ \frac{\partial^2 F(c^k)}{\partial c_i \partial c_j} \right]_{i \in \mathcal{S}_k, j \in \mathcal{S}_k}.$$

Compute projected Newton step

$$d_{\mathcal{S}_k} = -H_k^{-1} [\nabla F(c^k)]_{\mathcal{S}_k}.$$

Do a line search in this direction (but curtail it if any of the components in $\mathcal{S}_k$ change sign, that is, cross zero).

Get further savings by not evaluating the whole vector $\nabla F(c^k)$.

At iteration $k$, define working set $\mathcal{W}_k$ to contain

- components $i$ with $c_i^k \neq 0$; and
- some random fraction of the other components (say, 1% or 5%).

Evaluate only the $\mathcal{W}_k$ components of $\nabla F(c^k)$; set the other components of the step to zero: $d_i = 0$ for $i \notin \mathcal{W}_k$.

Still need to evaluate the full vector $\nabla F(c^k)$ to check optimality.

To solve for 21 values of $\lambda$ on a 3 GHz dual-core PC reqired

- a few seconds for 128 basis functions
- 4.5 minutes for 403,000 basis functions.

## Step 2: Parametric Linear Logistic Regression

The $N_B*$ patterns surviving the LASSO step are entered into a linear logistic regression model using `glmfit` in MATLAB and final pattern selection is then carried out by backward elimination. (Step 2). This is a heuristic step which has given us good results in a large number of simulations and a small number of applications. It was initially motivated by observing that putting the entire set of $N_B*$ surviving patterns in a linear logistic regression resulted in patterns whose coefficients were not significantly different from zero. One of the $N_B*$ patterns is removed, the model is fit, and the $BGACV$ score is computed. The pattern that gives the best score to the model after being taken out is removed from the model. This continues to the end and the final model is chosen from the pattern set with the best tuning score.

# Global vs. Greedy Algorithms

Conjecture: It is harder to pick out the right patterns if variables that are not directly related to the response are more highly correlated with those that are. Loosely speaking this is verified by the "Irrepresentable Condition" of Zhao and Yu(2006), which can be thought of as a measure of how much unimportant variables are correlated with important ones. This condition is almost necessary and sufficient for the oracle property.

Conjecture: The LASSO-Patternsearch algorithm does better than greedy competitors in these hard cases.

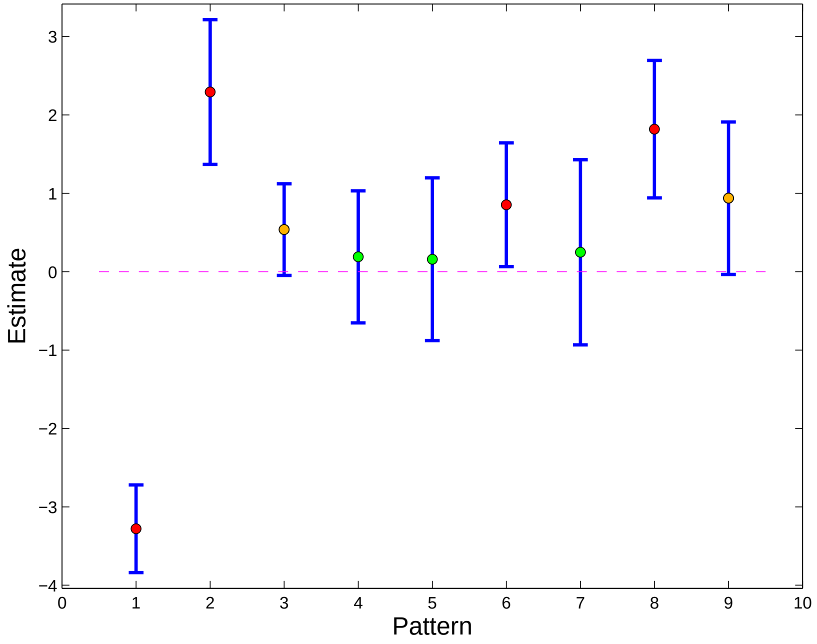
## Example 1: Risk of myopic change in a demographic study.

Applied to to “myopic change” from the Beaver Dam Eye Study, BDES 1 to BDES 2, five years apart. $n = 876$ records of persons aged 60-69 at BDES 1. A person whose “worse eye” scored at a decrease of .75 diopters or more is labeled $y = 1$, and 0 otherwise. Which variables or clusters of variables are predictive of this outcome? Consider seven variables of possible interest and want to see if there are high order interactions among the variables. The continuous variables are dichotomized so as to be able to do this.

Table 1: Trial Variables and Cutpoints

| | variable | description | binary cut point<br>(higher risk)<br>($X = 1$) |
|---|---|---|---|
| $X_1$ | 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 |

There are $2^7$ possible subsets (clusters) of variables that could be important.



Eight patterns (plus $\mu$) that survived the LASSO (Step 1), entered in a parametric linear logistic regression. Patterns numbered 2,6,8 and 9 passed Step 2, giving the final model

$$\begin{aligned} f &= -2.84 + 2.42 \cdot catct + 1.11 \cdot pky \cdot vtm \\ &\quad +1.98 \cdot sex \cdot inc \cdot jomyop \cdot asa + 1.15 \cdot sex \cdot inc \cdot catct \cdot asa. \end{aligned}$$

The significance levels for the coefficients of the four patterns in this model can be formally computed and are, respectively 3.3340e-21, 1.7253e-05, 1.5721e-04, and 0.0428.

## Smokers, Vitamins and Cataracts

Recalling “smoking = 1”, “not taking vitamins = 1” “having cataract(s) = 1”, we can read from this model:

$$\begin{aligned} f &= -2.84 + 2.42 \cdot catct + 1.11 \cdot pky \cdot vtm \\ &\quad +1.98 \cdot sex \cdot inc \cdot jomyop \cdot asa + 1.15 \cdot sex \cdot inc \cdot catct \cdot asa. \end{aligned}$$

Letting variable indicate “0”: we see that smokers with cataract or without cataract are protected by taking vitamins, $B_{pky \cdot vtm} = 0$. For non-smokers $B_{pky \cdot vtm} = 0$, taking or not taking vitamins makes no (significant) difference.

## Smokers, Vitamins and Cataracts

This result is physiologically meaningful-recent literature suggests:

a) Certain vitamins are good for eye health.

b) Smoking depletes the serum and tissue vitamin level, especially Vitamins C and E.

“However, our data are observational and subject to uncontrolled confounding. A randomized controlled clinical trial would provide the best evidence of any effect of vitamins on myopic change in smokers. ” (R. Klein in Shi *et al (2008))*

## Example 2. Rheumatoid Arthritis and SNPS in a Generative Model From GAW 15

The 15th Genetic Analysis Workshop (GAW 15) provided an extensive simulation data set of cases and controls with simulated single nucleotide polymorphisms (snp's) related to the risk of rheumatoid arthritis. This was an opportuniity to apply LPS to a set of large genetic attribute vectors with a known (hopefully realistic!) architecture, and compare the results with the description of the architecture generating the data. There were 1500 cases and 2000 controls, and 100 replicates. For the analysis we used the 674 snps in chromosome 6 along with three environmental variables, age, sex and smoking. Older than 55, female and smoking are the risky ($y = 1$) attributes. Most of the snps have three levels, normal, one variant allele and two variant alleles so two dummy variables were created to code this.

# Example 2. Rheumatoid Arthritis and SNPS in a Generative Model From GAW 15

A screen step one variable at a time with a (generous) passing criteria of at least one $p$ value less than 0.05 resulted in 72 snps, sex and smoking. Using all main effects and second order patterns from these 72 snps, sex and smoking resulted in 10371 basis functions. After some correction for an obvious miscoding in the GAW data, a model with five main effects and five two factor interactions was fit. (Next slide, above the double line). The LPS found the important variables.

# Simulated and Fitted Models, With and Without Third Order Patterns

| | Variable 1 | Level 1 | Variable 2 | Level 2 | Coef | Est |
|---|---|---|---|---|---|---|
| Main effects | constant | - | - | - | -4.8546 | -4.6002 |
| | $smoking$ | - | - | - | 0.8603 | 0.9901 |
| | $SNP6\_153$ | 1 | - | - | 1.8911 | 1.5604 |
| | $SNP6\_162$ | 1 | - | - | 2.2013 | 1.9965 |
| | $SNP6\_154$ | 2 | - | - | 0.7700 | 1.0808 |
| Second order patterns | $sex$ | - | $SNP6\_153$ | 1 | 0.7848 | 0.9984 |
| | $sex$ | - | $SNP6\_154$ | 2 | 0.9330 | 0.9464 |
| | $SNP6\_153$ | 2 | $SNP6\_154$ | 2 | 4.5877 | 4.2465 |
| | $SNP6\_153$ | 1 | $SNP6\_553$ | 2 | 0.4021 | 0 |
| | $SNP6\_154$ | 2 | $SNP6\_490$ | 1 | 0.3888 | 0 |
| | | Added | | | | |
| Third order pattern | $sex \cdot SNP6\_108\_2 \cdot SNP6\_334\_2$ | | | | 3 | 2.9106 |

Simulated model adapted from GAW 15 analysis. “Coef” is simulated coefficients. LPS run with main effects and two factor terms. “Est” is the estimated coefficients. LPS run with third order patterns (403,594 patterns) resulted in the same fit. Then a third order pattern added to the model, LPS successfully fitted it.

## Summary and Conclusions

Results can be generalized in several ways. The message is:

1. The log linear expansion of the multivariate Bernoulli distribution is good way to think about Bernoulli responses with attribute vectors consisting of bits.

2. Bernoulli data can tuned in a manner appropriate for Bernoulli data. BGACV is good for the variable selection problem with Bernoulli data.

3. An algorithm for fitting a very large number of unknowns simultaneously in a LASSO model is available.

4. It is argued that global methods based on simultaneous fits are better at fitting correlated predictors in the variable selection problem than greedy methods.

5. Firm theoretical foundations for items 2. and 4. would be nice to have.