The Lasso: a retrospective

Robert Tibshirani
Stanford University

RSS annual meeting 2010, Brighton, U.K.

Work here represents collaborations with many people, especially Trevor Hastie, Brad Efron, Jerome Friedman, Daniela Witten, Holger Hoefling, Pei Wang, Ryan Tibshirani

Email: tibs@stanford.edu
http://www-stat.stanford.edu/~tibs
Outline

- Review of lasso and its history
- Computational approaches
- Recent developments: applications and generalizations to other problems
- Discussion: challenges

**Not covering:** theoretical results - eg asymptotic recovery of true model- interesting! Buhlmann, Candes, Donoho, Meinshausen, Wainwright, Yu,... But not enough time.
I would like to honor:

Leo Breiman  Paul Tseng  John Nelder
1928-2005  1959-2009(?)  1924-2010

For GLMs,
the GLIM language,
A pioneer in statistical modelling
Regression shrinkage and selection via the Lasso

Tibshirani, JRSSB 1996

- Outcome variable $y_i$, for cases $i = 1, 2, \ldots n$, features $x_{ij}$, $j = 1, 2, \ldots p$

- Minimize

$$\sum_{i=1}^{n}(y_i - \sum_{j} x_{ij} \beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

- Equivalent to minimizing sum of squares with constraint $\sum |\beta_j| \leq s$.

- Similar to ridge regression, which has constraint $\sum j \beta_j^2 \leq t$

- Lasso does variable selection and shrinkage; ridge regression, in contrast only shrinks.
Picture of Lasso and Ridge regression

\[ \hat{\beta} \]

\[ \hat{\beta} \]

\( \beta_1 \)

\( \beta_2 \)

\( \hat{\beta} \)

\( \hat{\beta} \)

\( \hat{\beta} \)

\( \hat{\beta} \)

\( \hat{\beta} \)

lasso

ridge
More $\ell_q$ norms

Lasso uses $q = 1$, the value closest to subset selection ($q = 0$) that yields a convex problem.
Example: Prostate Cancer Data

\[ y_i = \log(\text{PSA}), \quad x_{ij} \text{ measurements on a man and his prostate} \]
History of the idea

- Lasso is regression with an $\ell_1$ norm penalty. $\ell_1$ norms have been around for long time!

- Most direct influence: Leo Breiman’s **garotte**. Idea is to minimize

$$\sum_{i=1}^{n} (y_i - \sum_{j} c_j x_{ij} \hat{\beta}_j)^2 \text{ subject to } c_j \geq 0, \sum_{j=1}^{p} c_j \leq t$$

where $\hat{\beta}_j$ are usual least squares estimates.

- This is undefined when $p > N$ (not a hot topic in 1995!) so I just combined the two stages into one (as a Canadian I also wanted a gentler name).

- Other related work around the same time: Frank and Friedman discussed **Bridge regression** (earlier -1993) using a penalty $\lambda \sum |\beta_j|^{\gamma}$, with both $\lambda$ and $\gamma$ estimated from the data. Chen, Donoho and Saunders (1998)- **Basis pursuit**.
submitted to JRSSB. Fairly positive reviews; one round of revisions.

idea did not get much attention until years later

why?
  – computation in 1996 was slow compared to today;
  – algorithms for lasso were black boxes, and not statistically motivated; (until LARS)
  – full statistical and numerical advantages of sparsity not appreciated
  – large data problems (in $N, p$ or both) were rare;
  – community did not have the R language for fast, easy sharing of new software tools
Citation counts from google scholar

![Graph showing the number of citations over years]

- Number of citations: 0, 200, 400, 600

The graph illustrates a significant increase in the number of citations over the years from 1996 to 2008.
... including a control group
Computational advances

- Original lasso paper used an off-the-shelf quadratic program solver. Doesn’t scale well. Not transparent.

- **LARS algorithm** (Efron, Hastie, Johnstone, Tibshirani 2002) gives an efficient way to solve the lasso, and connects the lasso to forward stagewise regression. Same algorithm is contained in the homotopy approach of Osborne, Presnell and Turlach (2000).

- **Coordinate descent** algorithms are extremely simple and fast, and exploit the assumed sparsity.

- There’s an explosion of activity in the optimization community in Nesterov’s **first order methods**.
Least Angle Regression — LAR (Efron et al 2002)

Like a “more democratic” version of forward stepwise regression.

1. Start with $r = y$, $\hat{\beta}_1, \hat{\beta}_2, \ldots \hat{\beta}_p = 0$. Assume $x_j$ standardized.

2. Find predictor $x_j$ most correlated with $r$.

3. Instead of simply entering the predictor, go slower: Increase $\beta_j$ in the direction of sign(corr($r, x_j$)) until some other competitor $x_k$ has as much correlation with current residual as does $x_j$.

4. Move $(\hat{\beta}_j, \hat{\beta}_k)$ in the joint least squares direction for $(x_j, x_k)$ until some other competitor $x_\ell$ has as much correlation with the current residual

5. Continue in this way until all predictors have been entered. Stop when corr($r, x_j$) = 0 $\forall j$, i.e. OLS solution.
LARS

\[ \sum |\hat{\beta}_j| \rightarrow \]

\[ |\hat{c}_{k,j}| \rightarrow \]

\( \hat{c}_k \)
Pathwise coordinate descent for the lasso (yr ?)

- Coordinate descent: optimize one parameter (coordinate) at a time.

- How? suppose we had only one predictor. Problem is to minimize

\[
\sum_{i}(y_i - x_i\beta)^2 + \lambda|\beta|
\]

- Solution is the soft-thresholded estimate

\[
\text{sign}(\hat{\beta})(|\beta| - \lambda)_+
\]

where \(\hat{\beta}\) is usual least squares estimate.

- Idea: with multiple predictors, cycle through each predictor in turn. Compute residuals \(r_i = y_i - \sum_{j \neq k} x_{ij}\hat{\beta}_k\) and apply soft-thresholding, pretending that our data is \((x_{ij}, r_i)\).
• Turns out that this is coordinate descent for the lasso criterion

$$\sum_i (y_i - \sum_j x_{ij} \beta_j)^2 + \lambda \sum |\beta_j|$$

• like skiing to the bottom of a hill, going north-south, east-west, north-south, etc.

• **Too simple?!**
A brief history of coordinate descent for the lasso

- 1997: Tibshirani’s student Wenjiang Fu at University of Toronto develops the “shooting algorithm” for the lasso. Tibshirani doesn’t fully appreciate it.

- 2002: Ingrid Daubechies gives a talk at Stanford, describes a one-at-a-time algorithm for the lasso. Hastie implements it, makes an error, and Hastie + Tibshirani conclude that the method doesn’t work.

- 2006: Friedman is the external examiner at the PhD oral of Anita van der Kooij (Leiden) who uses the coordinate descent idea for the Elastic net. Friedman wonders whether it works for the lasso. Friedman, Hastie + Tibshirani start working on this problem. See also Wu and Lange (2008).
Pathwise coordinate descent for the lasso

- Start with large value for $\lambda$ (very sparse model) and slowly decrease it
- most coordinates that are zero never become non-zero
- coordinate descent code for Lasso is just 73 lines of Fortran!
When does coordinate descent work?


If

\[ f(\beta_1 \ldots \beta_p) = g(\beta_1 \ldots \beta_p) + \sum h_j(\beta_j) \]

where \( g(\cdot) \) is convex and differentiable, and \( h_j(\cdot) \) is convex, then coordinate descent converges to a minimizer of \( f \).

Non-differential part of loss function must be separable.
Extensions

- Pathwise coordinate descent can be generalized to many other models: logistic/multinomial for classification, graphical lasso for undirected graphs, fused lasso for signals.
- Its speed and simplicity are quite remarkable.
- *glmnet* R package now available on CRAN. Handles lasso regression, logistic regression, multinomial regression, Cox model
Example: Logistic regression

- Outcome $Y = 0$ or $1$; Logistic regression model

$$\log\left(\frac{Pr(Y = 1)}{1 - Pr(Y = 1)}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 \ldots$$

- Criterion is binomial log-likelihood + absolute value penalty

- Example: sparse data. $N = 50,000$, $p = 700,000$.

- State-of-the-art interior point algorithm (Stephen Boyd, Stanford), exploiting sparsity of features: 3.5 hours for 100 values along path
Example: Logistic regression

- Outcome $Y = 0$ or $1$; Logistic regression model
  \[ \log\left(\frac{Pr(Y = 1)}{1 - Pr(Y = 1)}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 \ldots \]

- Criterion is binomial log-likelihood + absolute value penalty

- Example: sparse data. $N = 50,000$, $p = 700,000$.

- State-of-the-art interior point algorithm (Stephen Boyd, Stanford), exploiting sparsity of features: \textbf{3.5 hours} for 100 values along path

- \textit{Pathwise coordinate descent: 1 minute}
## Generalizations and variants of the lasso

<table>
<thead>
<tr>
<th>Method</th>
<th>Authors</th>
<th>Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>grouped lasso</td>
<td>Yuan and Lin</td>
<td>$\sum_g</td>
</tr>
<tr>
<td>elastic net</td>
<td>Zou and Hastie</td>
<td>$\lambda_1 \sum</td>
</tr>
<tr>
<td>fused lasso</td>
<td>Tibs et. al</td>
<td>$\lambda \sum</td>
</tr>
<tr>
<td>adaptive lasso</td>
<td>Zou</td>
<td>$\lambda_1 \sum w_j</td>
</tr>
<tr>
<td>graphical lasso</td>
<td>Fried., Hastie, Tibs</td>
<td>loglik $+\lambda</td>
</tr>
<tr>
<td>Dantzig selector</td>
<td>Candes and Tao</td>
<td>min $X^T(y - X\beta)</td>
</tr>
<tr>
<td>near monotonic reg.</td>
<td>Tibs, Hoef. and Tibs</td>
<td>$\sum (\beta_j - \beta_{j+1})_+$</td>
</tr>
<tr>
<td>matrix completion</td>
<td>Candes &amp; Tao</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Maz., Hastie, Tibs</td>
<td></td>
</tr>
<tr>
<td>multivariate methods</td>
<td>Jolliffe, Witten and many others</td>
<td>$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Lasso methods can shed light on more traditional ones

- graphical lasso for fitting a sparse Gaussian graph - based on

\[ \text{Gaussian loglikelihood} + \lambda \| \Sigma^{-1} \|_1 \]

Gives method for **graph selection**- determining which edges to include.

**Bonus**: special case of graphical lasso gives a new simple method for fitting a graph with **pre-specified** edges (structural zeroes in \( \Sigma^{-1} \)). Details in *Elements of Statistical Learning, 2nd ed*

- near-isotonic regression- details next
Gaussian graphical model

Edge between $x_i$ and $x_j$ is included if $(\Sigma^{-1})_{ij}$ is non-zero.
Near Isotonic regression

Ryan Tibshirani, Holger Hoefling, Rob Tibshirani (2010)

- generalization of isotonic regression: data sequence
  \[ y_1, y_2, \ldots, y_n. \]

  minimize \( \sum (y_i - \hat{y}_i)^2 \) subject to \( \hat{y}_1 \leq \hat{y}_2 \ldots \)

  Solved by Pool Adjacent Violators algorithm.

- Near-isotonic regression:

  \[
  \beta_\lambda = \arg\min_{\beta \in \mathbb{R}^n} \frac{1}{2} \sum_{i=1}^{n} (y_i - \beta_i)^2 + \lambda \sum_{i=1}^{n-1} (\beta_i - \beta_{i+1})_+, \]

  with \( x_+ \) indicating the positive part, \( x_+ = x \cdot 1(x > 0). \)
Near-isotonic regression—continued

• Convex problem. Solution path $\hat{\beta}_i = y_i$ at $\lambda = 0$ and culminates in usual isotonic regression as $\lambda \to \infty$. Along the way gives near monotone approximations.

• Simple algorithm that computes the entire path of solutions, a modified version of the well-known pool adjacent violators.

• We show that the degrees of freedom is the number of “plateaus” in the solution. Using results from Ryan Tibshirani’s PhD work with Jonathan Taylor.
Toy example

\[
\begin{align*}
\lambda &= 0 & \lambda &= 0.25 \\
\lambda &= 0.7 & \lambda &= 0.77
\end{align*}
\]
Global warming data
Sparse canonical correlation analysis


- Data matrices $X_1(n \times p)$, $X_2(n \times q)$ representing two sets of measurements on $n$ individuals
- Goal: find sparse sets of weights $w_1, w_2$ to maximize $\text{Corr}(X_1w_1, X_2w_2)$.
- formally

$$\text{maximize}_{w_1, w_2} w_1^TX_1^TX_2w_2$$

such that $\|w_1\|_1 \leq 1, \|w_2\|_1 \leq 1, \|w_1\|_1 \leq c_1, \|w_2\|_1 \leq c_2$.

- Bi-convex problem; simple alternating algorithm
- R package PMA (penalized multivariate analysis) and Excel addin Correlate
Example- Lymphoma data

\( \mathbf{X}_1 = \text{CGH (Copy number)}, \ \mathbf{X}_2 = \text{gene expression} \)

We apply sparse CCA by chromosome:

\[ \hat{\mathbf{w}}_1 = \]

\[ \hat{\mathbf{w}}_2 = \text{sets of weights for gene expression} \]
The matrix completion problem

- Data $X_{m \times n}$, for which only a relatively small number of entries are observed. The problem is to “complete” or impute the matrix based on the observed entries. Eg the Netflix database (see next slide).

- For a matrix $X_{m \times n}$ let $\Omega \subset \{1, \ldots, m\} \times \{1, \ldots, n\}$ denote the indices of observed entries. Consider the following optimization problem:

$$
\begin{align*}
\text{minimize} & \quad \text{rank}(Z) \\
\text{subject to} & \quad Z_{ij} = X_{ij}, \ \forall (i, j) \in \Omega \\
\end{align*}
$$

Not convex!
<table>
<thead>
<tr>
<th></th>
<th>Lord of the rings</th>
<th>Pretty Woman</th>
<th>Harry Potter</th>
<th>Pulp Fiction</th>
<th>Kill Bill</th>
<th>Blue velvet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Daniela</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Genevera</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>2</td>
<td>?</td>
<td>1</td>
</tr>
<tr>
<td>Larry</td>
<td>1</td>
<td>?</td>
<td>2</td>
<td>5</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Jim</td>
<td>?</td>
<td>?</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>Andy</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>?</td>
<td>?</td>
<td>5</td>
</tr>
</tbody>
</table>
• Make the problem convex by replacing “rank” with nuclear norm (sum of the singular values):

\[
\begin{align*}
\text{minimize} & \quad \|Z\|_* \\
\text{subject to} & \quad Z_{ij} = X_{ij}, \forall (i,j) \in \Omega
\end{align*}
\]  \hspace{1cm} (2)

(Fazel 2002).

• This criterion is used by Candes et al 2009. Fascinating work on conditions for exact reconstruction.

• But this criterion requires the training error to be zero. This is too harsh and can overfit!

• Instead we use penalized reconstruction error:

\[
\begin{align*}
\text{minimize} & \quad \sum_{(i,j) \in \Omega} (Z_{ij} - X_{ij})^2 + \lambda \cdot \|Z\|_* \\
\end{align*}
\]  \hspace{1cm} (3)
Idea of Algorithm

(Mazunder, Hastie, Tibshirani 2010)

1. impute the missing data with some initial values
2. compute the singular value decomposition (SVD) of the current imputed matrix, and soft-threshold the singular values:
3. reconstruct the SVD and hence obtain new imputations for missing values
4. repeat steps 2,3 until convergence

\[ \hat{Z} = UD_{d_i}V^T \rightarrow UD_{S(d_i,\lambda)V^T} \] (4)
### Timings

| $(m, n)$         | $|\Omega|$ | true rank | SNR | effective rank | time(s)       |
|------------------|------------|-----------|-----|----------------|---------------|
| $(3 \times 10^4, 10^4)$ | $10^4$     | 15        | 1   | (13, 47, 80)   | (41.9, 124.7, 305.8) |
| $(10^5, 10^5)$   | $10^4$     | 15        | 10  | (5, 14, 32, 62) | (37, 74.5, 199.8, 653) |
| $(10^5, 10^5)$   | $10^5$     | 15        | 10  | (18, 80)       | (202, 1840)   |
| $(5 \times 10^5, 5 \times 10^5)$ | $10^4$     | 15        | 10  | 11             | 628.14        |
| $(5 \times 10^5, 5 \times 10^5)$ | $10^5$     | 15        | 1   | (3, 11, 52)    | (341.9, 823.4, 4810.75) |
| $(10^6, 10^6)$   | $10^5$     | 15        | 1   | 80             | 8906          |
• lasso ($\ell_1$) penalties are useful for fitting a wide variety of models

• New algorithms allow application of these models to large datasets, exploiting sparsity for **both statistical and computation gains**.

• **Challenge**: we are statisticians, not computer scientists. This is a fun area to work in, but we should not invent new models/algorithms just for the sake of it. We should focus on developing tools and understanding their properties, to help us to do **better science**.