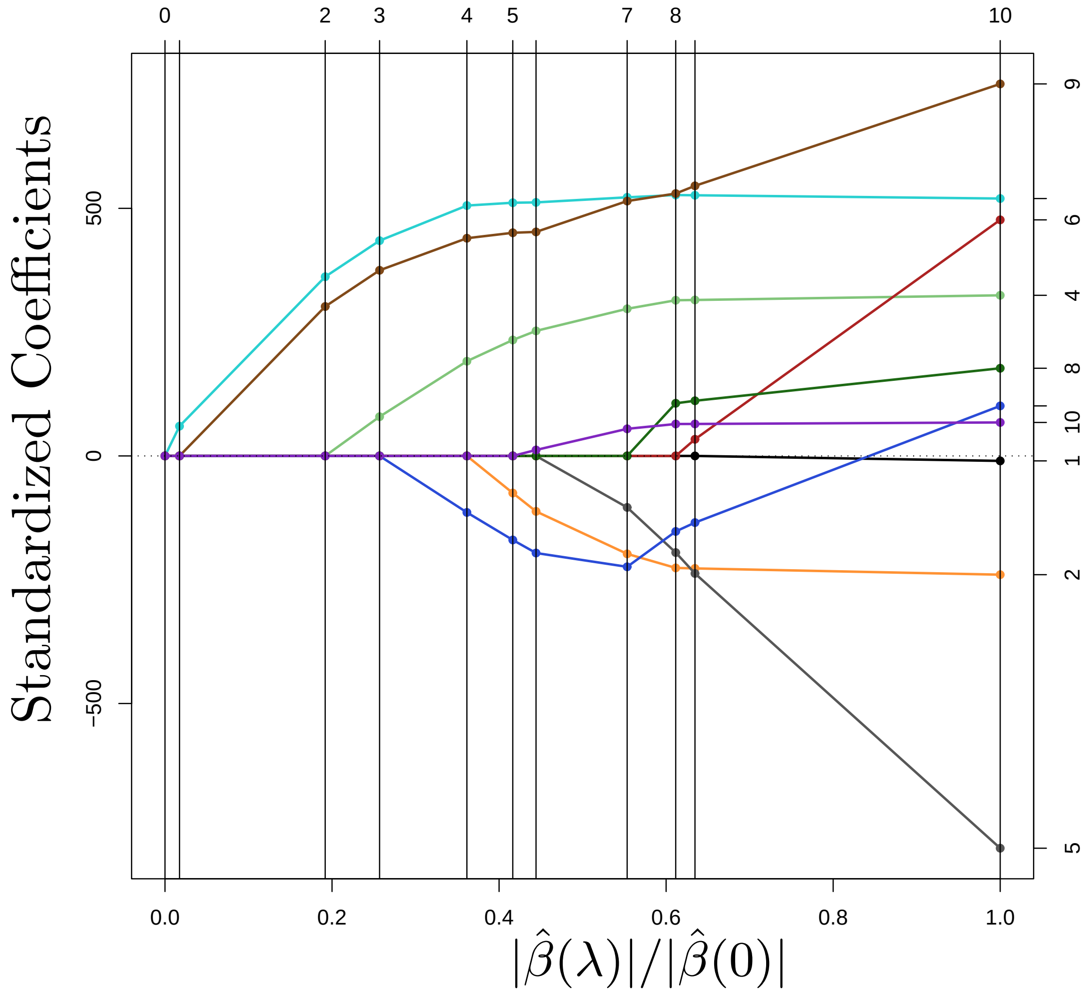
Fast Regularization Paths via Coordinate Descent

Trevor Hastie

Stanford University

joint work with Jerry Friedman and Rob Tibshirani.

Lasso Coefficient Profile



Lasso: $\hat{\beta}(\lambda) = \text{argmin}_\beta \sum_{i=1}^{N}(y_i - \beta_0 - x_i^T\beta)^2 + \lambda||\beta||_1$

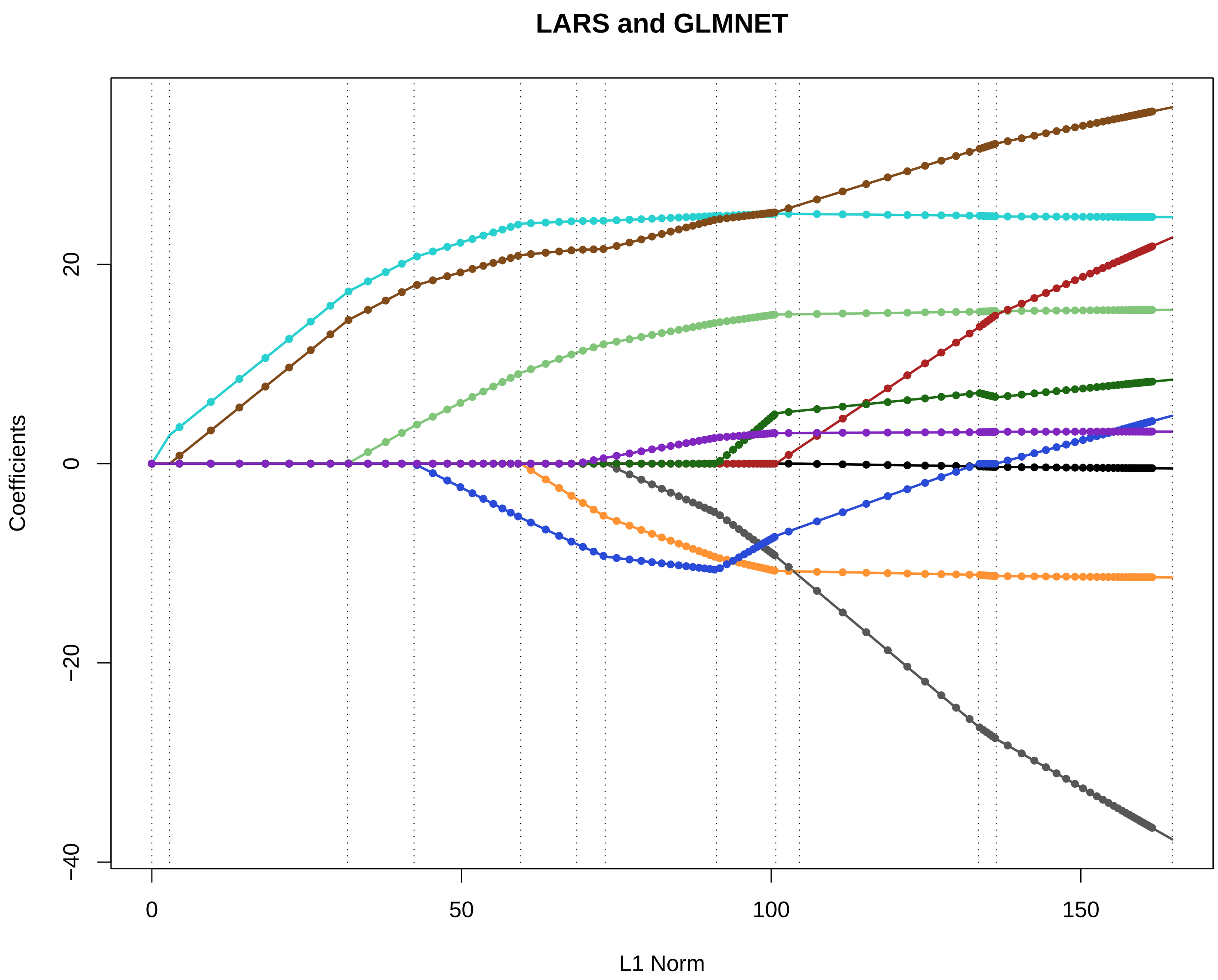
History

- In 2001 the LARS algorithm (Efron et al) provides a way to compute the entire lasso coefficient path efficiently at the cost of a full least-squares fit.
- Efficient path algorithms allow for easy and exact cross-validation and model selection.
- 2001 – present: path algorithms pop up for a wide variety of related problems: grouped lasso, support-vector machine, elastic net, quantile regression, logistic regression and glms, Cox proportional hazards model, Dantzig selector, ...
- Many of these do not enjoy the piecewise-linearity of LARS, and sieze up on very large problems.

Coordinate Descent

- Solve the lasso problem by coordinate descent: optimize each parameter separately, holding all the others fixed. Updates are trivial. Cycle around till coefficients stabilize.
- Do this on a grid of λ values, from λ_{max} down to λ_{min} (uniform on log scale), using warms starts.
- Can do this with a variety of loss functions and additive penalties.

Coordinate descent achieves dramatic speedups over all competitors, by factors of 10, 100 and more.


LARS and GLMNET
Coefficients
L1 Norm
−40
−20
0
20
0
50
100
150

Speed Trials

Competitors:

`lars` As implemented in R package, for squared-error loss.

`glmnet` Fortran based R package using coordinate descent — topic of this talk. Does squared error and logistic (2- and K-class).

`l1logreg` Lasso-logistic regression package by Koh, Kim and Boyd, using state-of-art interior point methods for convex optimization.

`BBR/BMR` Bayesian binomial/multinomial regression package by Genkin, Lewis and Madigan. Also uses coordinate descent to compute posterior mode with laplace prior—the lasso fit.

Based on simulations (next 3 slides) and real data (4th slide).

Linear Regression — Dense Features

	Average Correlation between Features					
	0	0.1	0.2	0.5	0.9	0.95
	$N = 5000,\ p = 100$					
glmnet	0.05	0.05	0.05	0.05	0.05	0.05
lars	0.29	0.29	0.29	0.30	0.29	0.29
	$N = 100,\ p = 50000$					
glmnet	2.66	2.46	2.84	3.53	3.39	2.43
lars	58.68	64.00	64.79	58.20	66.39	79.79

Timings (secs) for **`glmnet`** and **`lars`** algorithms for linear regression with lasso penalty. Total time for 100 λ values, averaged over 3 runs.

Logistic Regression — Dense Features

	Average Correlation between Features					
	0	0.1	0.2	0.5	0.9	0.95
	$N = 5000,\ p = 100$					
glmnet	7.89	8.48	9.01	13.39	26.68	26.36
l1lognet	239.88	232.00	229.62	229.49	223.19	223.09
	$N = 100,\ p = 5000$					
glmnet	5.24	4.43	5.12	7.05	7.87	6.05
l1lognet	165.02	161.90	163.25	166.50	151.91	135.28

Timings (seconds) for logistic models with lasso penalty. Total time for tenfold cross-validation over a grid of 100 λ values.

Logistic Regression — Sparse Features

	0	0.1	0.2	0.5	0.9	0.95
			$N = 10,000,\ p = 100$			
glmnet	3.21	3.02	2.95	3.25	4.58	5.08
BBR	11.80	11.64	11.58	13.30	12.46	11.83
l1lognet	45.87	46.63	44.33	43.99	45.60	43.16
			$N = 100,\ p = 10,000$			
glmnet	10.18	10.35	9.93	10.04	9.02	8.91
BBR	45.72	47.50	47.46	48.49	56.29	60.21
l1lognet	130.27	124.88	124.18	129.84	137.21	159.54

Timings (seconds) for logistic model with lasso penalty and sparse features (95% zeros in X). Total time for ten-fold cross-validation over a grid of 100 λ values.

Logistic Regression — Real Datasets

Name	Type	N	p	**glmnet**	**l1logreg**	**BBR BMR**
			Dense			
Cancer	14 class	144	16,063	2.5 mins	NA	2.1 hrs
Leukemia	2 class	72	3571	2.50	55.0	450
			Sparse			
Internet ad	2 class	2359	1430	5.0	20.9	34.7
Newsgroup	2 class	11,314	777,811	2 mins	3.5 hrs	

Timings in seconds (unless stated otherwise). For Cancer, Leukemia and Internet-Ad, times are for ten-fold cross-validation over 100 λ values; for Newsgroup we performed a single run with 100 values of λ, with $\lambda_{min} = 0.05\lambda_{max}$.

A brief history of coordinate descent for the lasso

1997 Tibshirani's student Wenjiang Fu at U. 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 external examiner at PhD oral of Anita van der Kooij (Leiden) who uses coordinate descent for elastic net. Friedman, Hastie + Tibshirani revisit this problem. Others have too — Krishnapuram and Hartemink (2005), Genkin, Lewis and Madigan (2007), Wu and Lange (2008), Meier, van de Geer and Buehlmann (2008).

Coordinate descent for the lasso

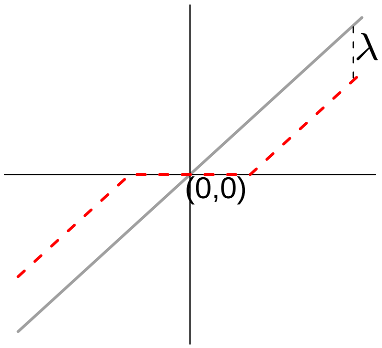
$$\min_\beta \frac{1}{2N}\sum_{i=1}^N (y_i - \sum_{j=1}^p x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^p |\beta_j|$$

Suppose the p predictors and response are standardized to have mean zero and variance 1. Initialize all the $\beta_j = 0$.

Cycle over $j = 1, 2, \ldots, p, 1, 2, \ldots$ till convergence:

- Compute the partial residuals $r_{ij} = y_i - \sum_{k \neq j} x_{ik}\beta_k$.
- Compute the simple least squares coefficient of these residuals on jth predictor: $\beta_j^* = \frac{1}{N}\sum_{i=1}^N x_{ij} r_{ij}$
- Update β_j by *soft-thresholding*:

$$\begin{aligned} \beta_j &\leftarrow S(\beta_j^*, \lambda) \\ &= \text{sign}(\beta_j^*)(|\beta_j^*| - \lambda)_+ \end{aligned}$$



Why is coordinate descent so fast?

Naive Updates: $\beta_j^* = \frac{1}{N}\sum_{i=1}^N x_{ij} r_{ij} = \frac{1}{N}\sum_{i=1}^N x_{ij} r_i + \beta_j$, where r_i is current model residual; $O(N)$. Many coeficients are zero, and stay zero. If a coefficient changes, residuals are updated in $O(N)$ computations.

Covariance Updates: $\sum_{i=1}^N x_{ij} r_i = \langle x_j, y\rangle - \sum_{k:|\beta_k|>0} \langle x_j, x_k\rangle \beta_k$ Cross-covariance terms are computed once for active variables and stored (helps a lot when $N \gg p$).

Sparse Updates: If data is sparse (many zeros), inner products can be computed efficiently.

Active Set Convergence: After a cycle through p variables, we can restrict further iterations to the *active set* till convergence + one more cycle through p to check if active set has changed. Helps when $p \gg N$.

Warm Starts: We fits a sequence of models from $\lambda_{\max}$ down to $\lambda_{\min} = \epsilon\lambda_{max}$ (on log scale). λ_{max} is smallest value of λ for which all coefficients are zero. Solutions don't change much from one λ to the next. Convergence is often faster for entire sequence than for single solution at small value of λ.

FFT: Friedman + Fortran + Tricks — no sloppy flops!

Binary Logistic Models

Newton Updates: For binary logistic regression we have an outer Newton loop at each λ. This amounts to fitting a lasso with weighted squared error-loss. Uses weighted soft thresholding.

Multinomial: We use a symmetric formulation for multi- class logistic:

$$\Pr(G = \ell|x) = \frac{e^{\beta_{0\ell}+x^T\beta_\ell}}{\sum_{k=1}^K e^{\beta_{0k}+x^T\beta_k}}.$$

This creates an additional loop, as we cycle through classes, and compue the quadratic approximation to the multinomial log-likelihood, holding all but one classes parameters fixed.

Details Many important but tedious details with logistic models. e.g. if $p \gg N$, cannot let λ run down to zero.

Elastic-net Penalty

Proposed in Zou and Hastie (2005) for $p \gg N$ situations, where predictors are correlated in groups.

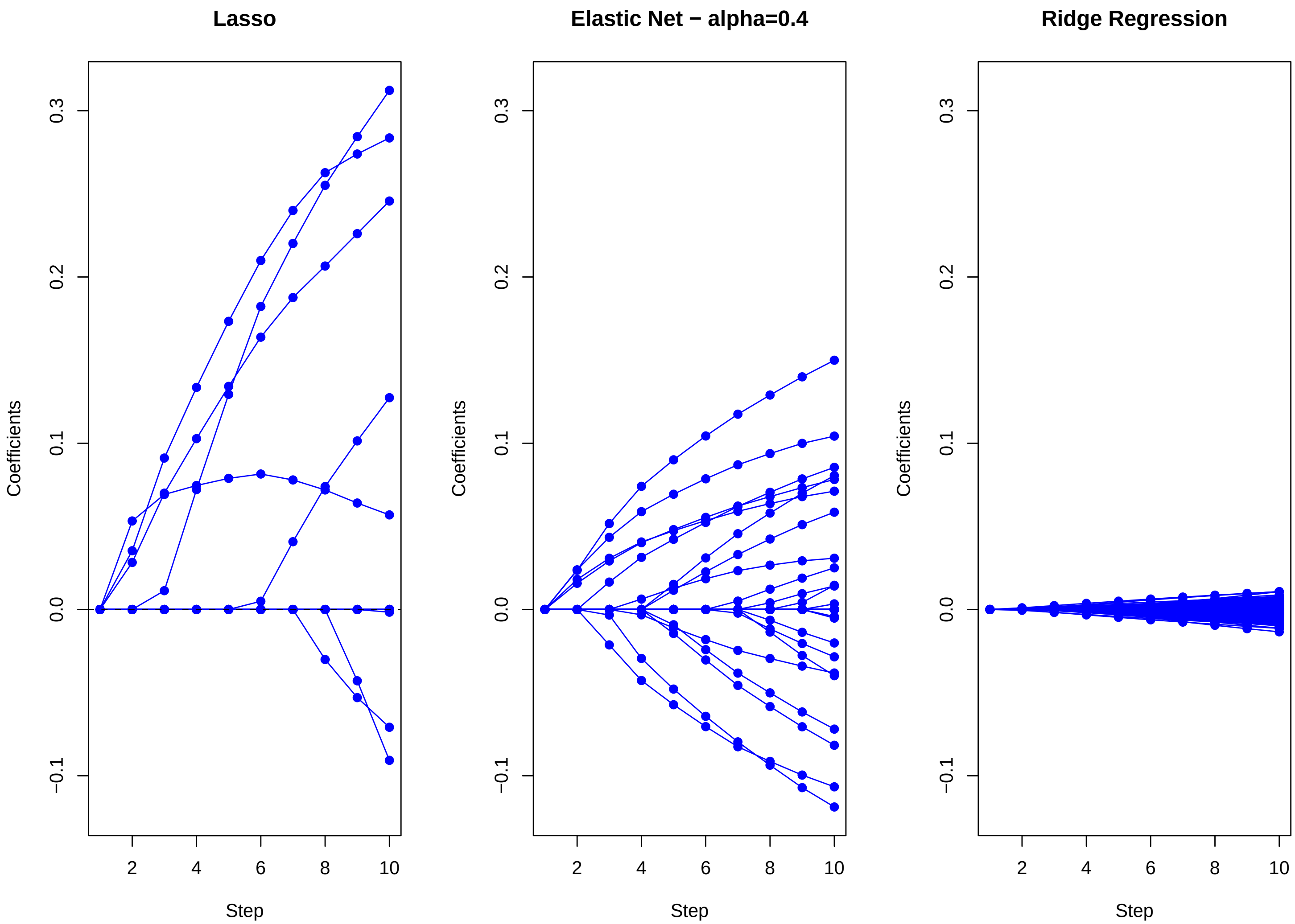
$$P_\alpha(\beta) = \sum_{j=1}^{p} \left[\tfrac{1}{2}(1-\alpha)\beta_j^2 + \alpha|\beta_j|\right].$$

α creates a compromise between the *lasso* and *ridge*.

Coordinate update is now

$$\beta_j \leftarrow \frac{S(\beta_j^*,\ \lambda\alpha)}{1+\lambda(1-\alpha)}$$

where $\beta_j^* = \frac{1}{N}\sum_{i=1}^N x_{ij}r_{ij}$ as before.



Leukemia Data, Logistic, N=72, p=3571, first 10 steps shown

Multiclass classification

Microarray classification: 16,063 genes, 144 training samples 54 test samples, 14 cancer classes. Multinomial regression model.

Methods	CV errors out of 144	Test errors out of 54	# of genes used
1. Nearest shrunken centroids	35 (5)	17	6520
2. L_2-penalized discriminant analysis	25 (4.1)	12	16063
3. Support vector classifier	26 (4.2)	14	16063
4. Lasso regression (one vs all)	30.7 (1.8)	12.5	1429
5. K-nearest neighbors	41 (4.6)	26	16063
6. L_2-penalized multinomial	26 (4.2)	15	16063
7. Lasso-penalized multinomial	17 (2.8)	13	269
8. Elastic-net penalized multinomial	22 (3.7)	11.8	384

6–8 fit using `glmnet`

Summary

Many problems have the form

$$\min_{\{\beta_j\}_1^p} \left[R(y, \beta) + \lambda \sum_{j=1}^{p} P_j(\beta_j) \right].$$

- If R and P_j are convex, and R is differentiable, then coordinate descent converges to the solution (Tseng, 1988).
- Often each coordinate step is trivial. E.g. for lasso, it amounts to soft-thresholding, with many steps leaving $\hat{\beta}_j = 0$.
- Decreasing λ slowly means not much cycling is needed.
- Coordinate moves can exploit sparcity.

Other Applications

- Undirected graphical models — learning dependence structure via the lasso. Model the inverse covariance in the Gaussian family with L_1 penalties applied to elements. Modified lasso algorithm, which we solve by coordinate descent (FHT 2007).
- Grouped lasso (Yuan and Lin, 2007, Meier, Van de Geer, Buehlmann, 2008) — each term $P_j(\beta_j)$ applies to *sets* of parameters:

$$\sum_{j=1}^{J} ||\beta_j||_2.$$

  Leads to a block-updating form of coordinate descent.
- CGH modeling and the fused lasso. Here the penalty has the

form

$$\sum_{j=1}^{p} |\beta_j| + \alpha \sum_{j=1}^{p-1} |\beta_{j+1} - \beta_j|.$$

This is not additive, so a modified coordinate descent algorithm is required (FHT + Hoeffling 2007).