Least Angle Regression

Brad Efron et.al.* (2004)
Stanford University

* Trevor Hastie, Ian Johnstone, Rob Tibshirani
The Beginning

It all started with this picture on page 330 in our *Elements of Statistical Learning (2001)*. This picture links boosting to the lasso, and is intended to explain how boosting fits models in high-dimensional space.
Adaboost Stumps for Classification

![Graph showing test misclassification error over iterations for Adaboost Stumps.](image)

- **Red line**: Adaboost Stump
- **Blue line**: Adaboost Stump shrink 0.1

Legend:
- Adaboost Stump
- Adaboost Stump shrink 0.1

**Axes:**
- **Y-axis**: Test Misclassification Error
- **X-axis**: Iterations

**Values:**
- Test Misclassification Error ranges from 0.24 to 0.36
- Iterations range from 0 to 1000

**Notes:**
- The graph compares the test misclassification error for Adaboost Stumps and Adaboost Stumps shrink 0.1 over iterations.
- The Adaboost Stump shrink 0.1 shows a lower error rate compared to the Adaboost Stumps.
Least Squares Boosting with Trees

*Elements of Statistical Learning (chapter 10)*

Response $y$, predictors $x = (x_1, x_2 \ldots x_p)$.

1. Start with function $F(x) = 0$ and residual $r = y$
2. Fit a CART regression tree to $r$ giving $f(x)$
3. Set $F(x) \leftarrow F(x) + \epsilon f(x)$, $r \leftarrow r - \epsilon f(x)$ and repeat steps 2 and 3 many times
Here is a version of least squares boosting for linear regression:
(assume predictors are standardized)

(Incremental) Forward Stagewise

1. Start with $r = y$, $\beta_1, \beta_2, \ldots \beta_p = 0$.  
2. Find the predictor $x_j$ most correlated with $r$  
3. Update $\beta_j \leftarrow \beta_j + \delta_j$, where $\delta_j = \epsilon \cdot \text{sign} \langle r, x_j \rangle$  
4. Set $r \leftarrow r - \delta_j \cdot x_j$ and repeat steps 2 and 3 many times

$\delta_j = \langle r, x_j \rangle$ gives usual forward stagewise; different from forward stepwise

Analogous to least squares boosting, with $trees=predictors$
Least Angle Regression — LAR

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. Increase $\beta_j$ in the direction of sign $\langle r, x_j \rangle$ 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 $\langle r, x_j \rangle = 0 \ \forall \ j$, i.e. OLS solution.
The LAR direction $\mathbf{u}_2$ at step 2 makes an equal angle with $\mathbf{x}_1$ and $\mathbf{x}_2$. 
- Maximal correlations decrease with steps.
- Lars/lasso decreases RSS optimally with increase in $||\beta||_1$.
- \textit{Dantzig Selector} decreases maximum $|\langle x_j, r \rangle|$ optimally with increase in $||\beta||_1$. (Candes and Tao, 2006)
\[ t = \sum |\hat{\beta}_j| \rightarrow \]
\[ t = \sum |\hat{\beta}_j| \rightarrow \]
LAR, Lasso and Forward Stagewise

• Not always identical

• In orthogonal predictor case: yes

• In hard to verify case of monotone coefficient paths: yes

• In general, almost!

• LAR algorithm can be simply modified to give both Lasso and Forward Stagewise paths.
Start with LAR. If a coefficient crosses zero, stop. Drop that predictor, recompute the best direction and continue. This gives the Lasso path

“Proof”: use KKT conditions for appropriate Lagrangian.

Informally:

\[
\frac{\partial}{\partial \beta_j} \left[ \frac{1}{2} \| y - X\beta \|^2 + \lambda \sum_j |\beta_j| \right] = 0
\]

\[\Leftrightarrow\]

\[\langle x_j, r \rangle = \lambda \cdot \text{sign}(\hat{\beta}_j) \quad \text{if } \hat{\beta}_j \neq 0 \text{ (active)}\]
Compute the LAR direction, but constrain the sign of the coefficients to match the correlations $\langle x_j, r \rangle$.

The incremental forward stagewise procedure approximates these steps, one predictor at a time. As step size $\epsilon \to 0$, can show that it coincides with this modified version of LAR.
The forward stagewise direction lies in the positive cone spanned by the (signed) predictors with equal correlation with the current residual.
The LARS algorithm computes the entire Lasso/FS/LAR path in same order of computation as one full least squares fit. When \( p \gg N \), the solution has at most \( N \) non-zero coefficients. Works efficiently for micro-array data (\( p \) in thousands).

Cross-validation is quick and easy.

Available for R (CRAN) and Splus.
Stein’s Lemma: \( df(\hat{\mu}) \overset{\text{def}}{=} \sum_{i=1}^{n} \text{cov}(\hat{\mu}_i, y_i)/\sigma^2 \)
Lasso or Forward Stagewise?

- Micro-array example (Golub Data). $N = 38$, $p = 7129$, response binary ALL vs AML
- Lasso behaves chaotically near the end of the path, while Forward Stagewise is smooth, stable, and mostly monotone.
LARS started a path frenzy!

- **Elasticnet**: (Zou and Hastie, 2005). Compromise between lasso and ridge: minimize $\sum_i(y_i - \sum_j x_{ij} \beta_j)^2$ subject to $\alpha||\beta||_1 + (1 - \alpha)||\beta||_2^2 \leq t$. Useful for situations where variables operate in correlated groups (genes in pathways).

- **Glmpath**: (Park and Hastie, 2005). Approximates the $L_1$ regularization path for generalized linear models. e.g. logistic regression, Poisson regression.

- **Grouped Lasso**: (Yuan and Li, 2006, following Lin and Zhang, 2002). Extends lasso to deal with groups of variables (e.g. dummy variables for factors):

  $$\max_{\beta} \ell(y; \beta) - \lambda \sum_{m=1}^{M} \gamma_m ||\beta_m||_2.$$
The $\ell_2$ norm ensures the vector of coefficients $\beta_m$ for each group are all zero or non-zero together. Using predictor-corrector methods we can construct the path for this criterion: Park & Hastie (2006), *Regularization path algorithms for detecting gene interactions*.

- **Bin Yu group**: Boosted Lasso and Composite Absolute Penalties.

- Rosset and Zhu (2004) discuss conditions needed to obtain piecewise-linear paths. A combination of piecewise quadratic/linear loss function, and an $L_1$ penalty, is sufficient.

- **Svmpath**: Entire regularization path for the SVM. Exploits previous point — loss is piecewise linear, penalty is quadratic (Hastie, Rosset, Tibshirani and Zhu, JMLR 2004)

- Bach and Jordan (2004) have path algorithms for Kernel estimation, and for efficient ROC curve estimation. The latter
is a useful generalization of the \texttt{svmpath} algorithm discussed above.

- Friedman and Popescu (2004) extend the incremental forward stagewise idea by updating all coefficients whose gradient exceeds a predefined threshold.

- fused lasso, graphical lasso, \ldots and many more.
Pathwise Coordinate Descent

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.

- Often each coordinate step is trivial. E.g. for lasso, it amounts to soft-thresholding, with many steps leaving $\hat{\beta}_j = 0$.

- Decreasing $\lambda$ slowly means not much cycling is needed.

- Coordinate moves can exploit sparsity. Currently we (JHF, TH, RT) fit a lasso logistic regression path (100 $\lambda$ values) with $N = 11K$ and $p = 4.8M$ and sparsity 0.0001 in < 2 minutes.
Conclusions

Brad: on behalf of all the path crazies, thanks for LARS and happy birthday!
Conclusions

Brad: on behalf of all the path crazies, thanks for LARS and happy birthday!
And by the way, a great movie ...
Conclusions

Brad: on behalf of all the path crazies, thanks for LARS and happy birthday!
And by the way, a great movie ...