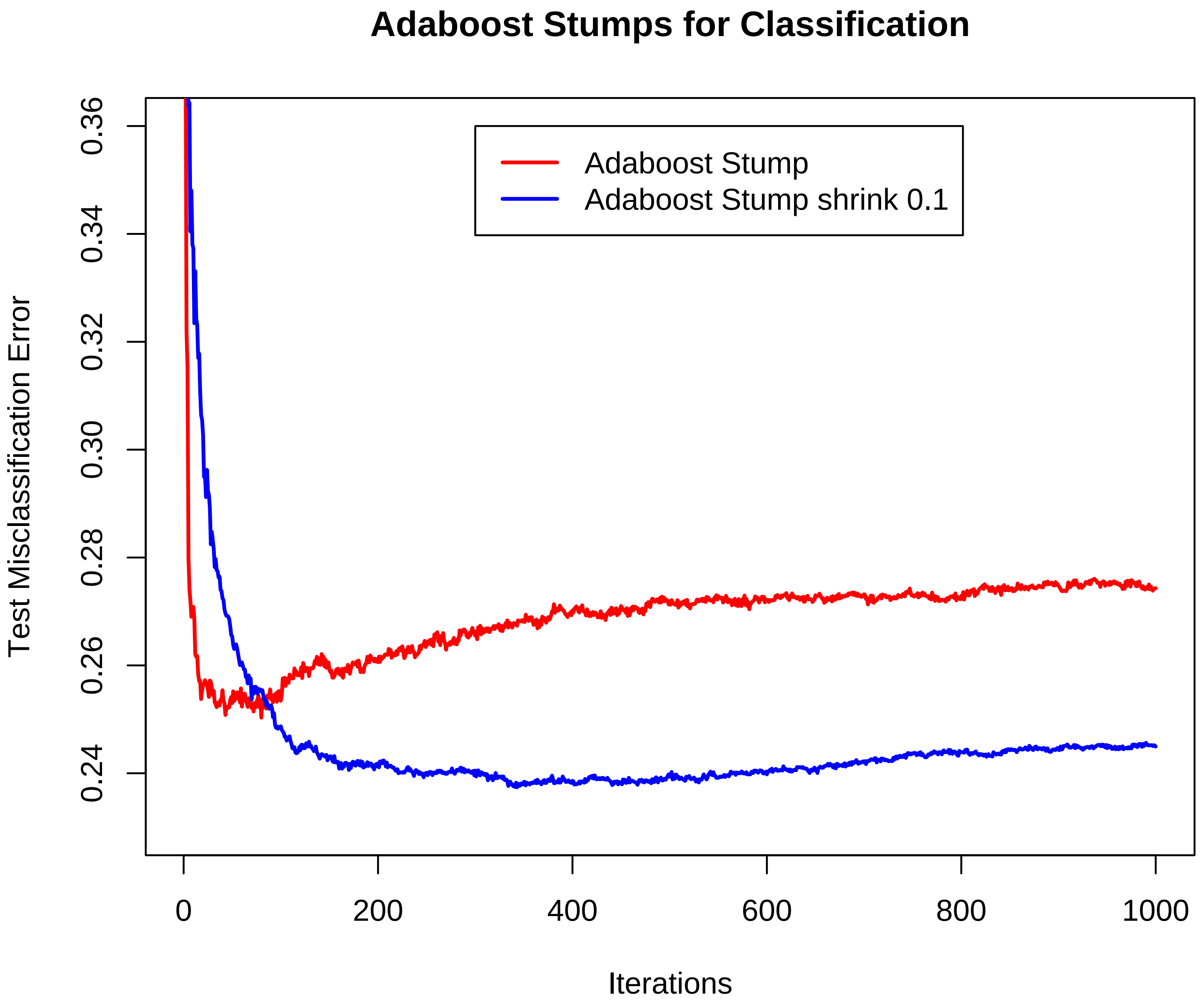
Regularization Paths

Trevor Hastie
Stanford University

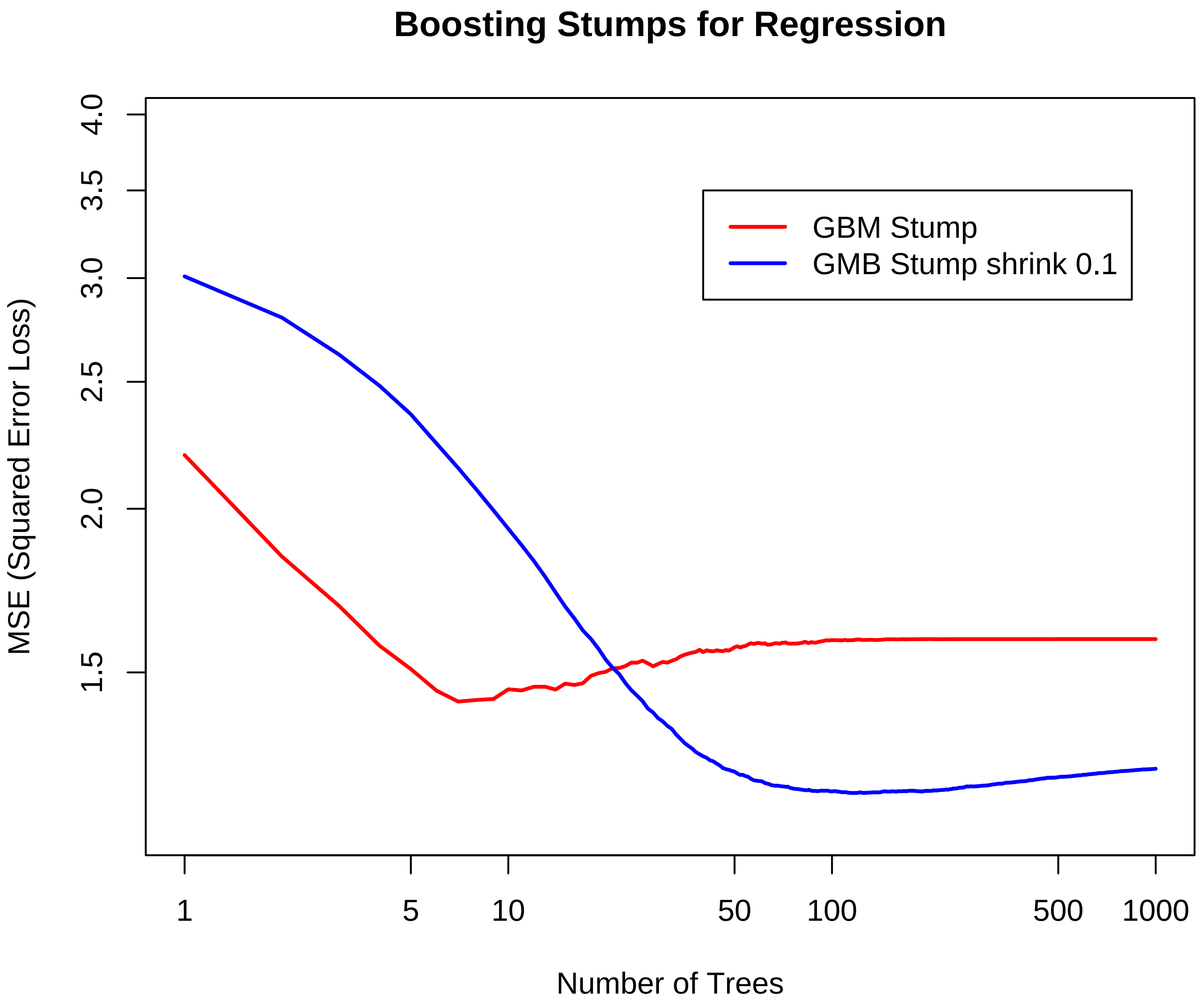
drawing on collaborations with Brad Efron, Saharon Rosset, Ji Zhu, Hui Zhou, Rob Tibshirani and Mee-Young Park

Theme

- Boosting fits a regularization path towards a max-margin classifier. Svmpath does as well.
- In neither case is this endpoint always of interest — somewhere along the path is often better.
- Having efficient algorithms for computing entire paths facilitates this selection.



Adaboost Stumps for Classification
Adaboost Stump
Adaboost Stump shrink 0.1
Test Misclassification Error
0.24
0.26
0.28
0.30
0.32
0.34
0.36
0
200
400
600
800
1000
Iterations


Boosting Stumps for Regression
MSE (Squared Error Loss)
4.0
3.5
3.0
2.5
2.0
1.5
GBM Stump
GMB Stump shrink 0.1
1
5
10
50
100
500
1000
Number of Trees

Least Squares Boosting

Friedman, Hastie & Tibshirani — see *Elements of Statistical Learning (chapter 10)*

Supervised learning: 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

Linear Regression

Here is a version of least squares boosting for multiple linear regression: (assume predictors are standardized)

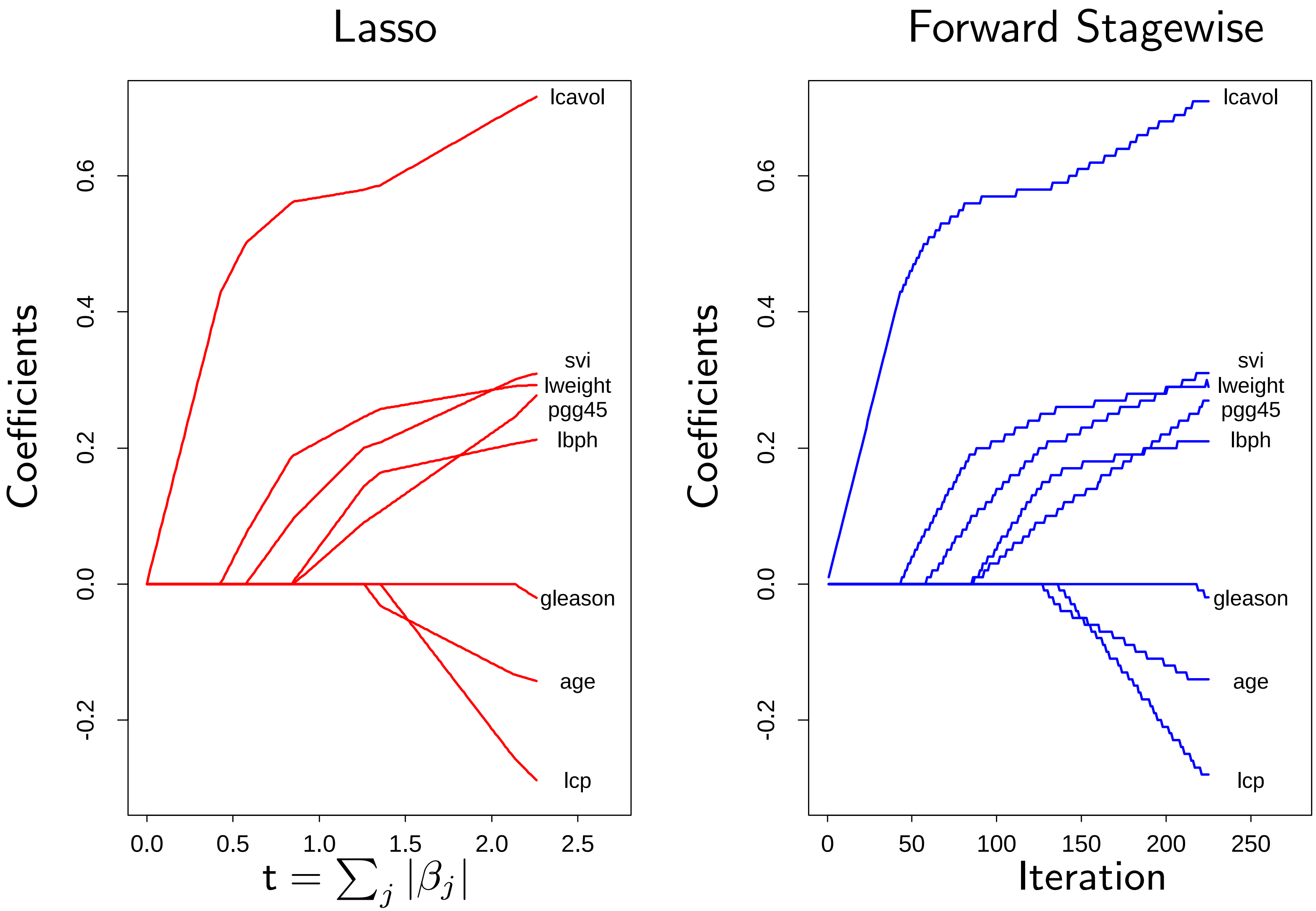
(Incremental) Forward Stagewise

1. Start with $r = y$, $\beta_1, \beta_2, \dots \beta_p = 0$.
2. Find the predictor x_j most correlated with r
3. Update $\beta_j \leftarrow \beta_j + \delta_j$, where $\delta_j = \epsilon \cdot \mathrm{sign}\langle r, x_j \rangle$
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

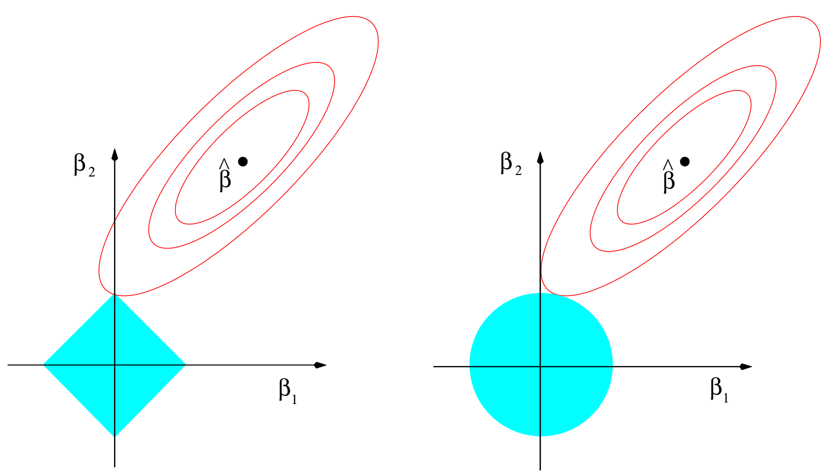
Analogous to least squares boosting, with *trees=predictors*

Example: Prostate Cancer Data

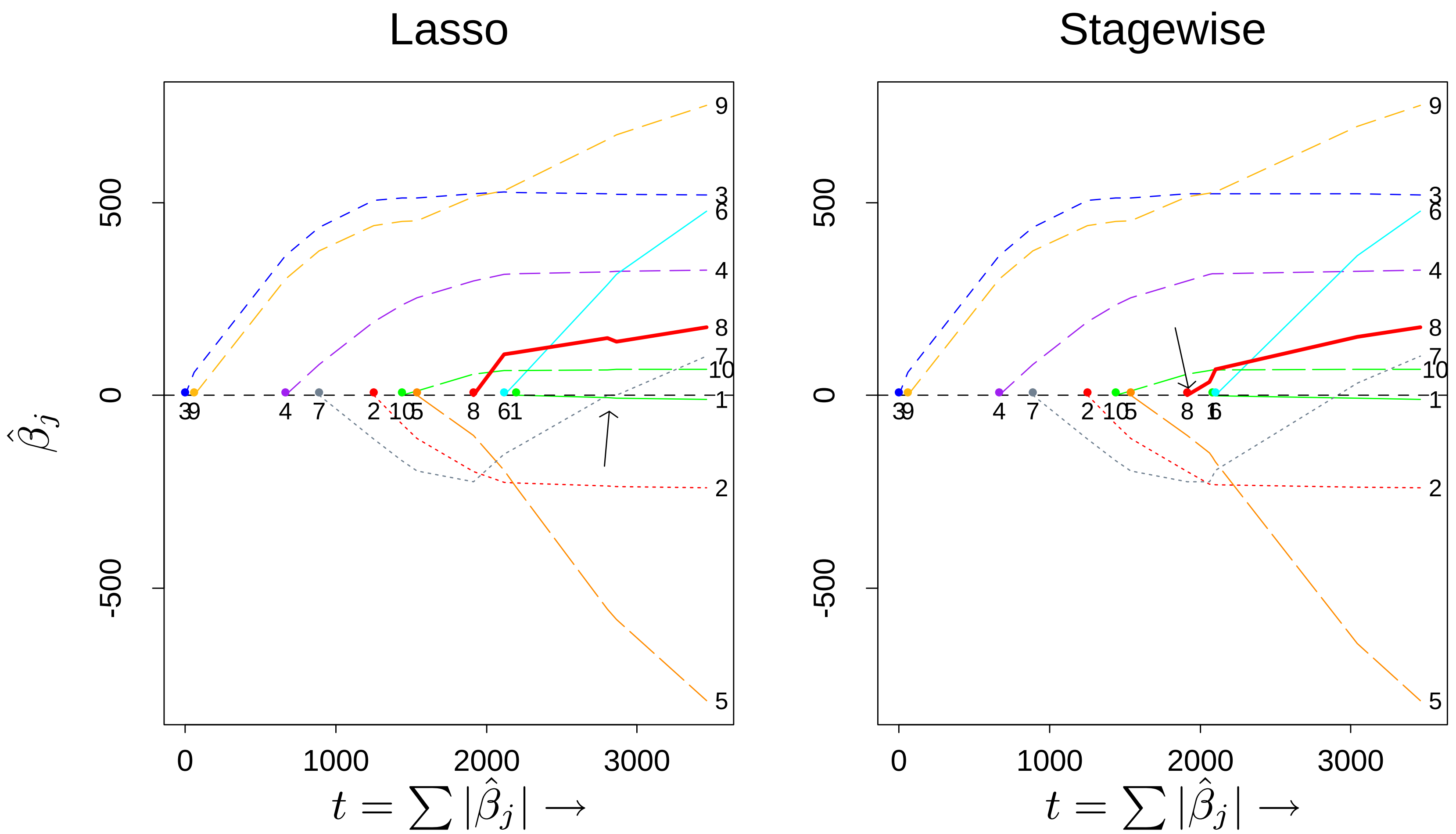


Linear regression via the Lasso (Tibshirani, 1995)

- Assume $\bar{y} = 0$, $\bar{x}_j = 0$, $\mathrm{Var}(x_j) = 1$ for all j.
- Minimize $\sum_i (y_i - \sum_j x_{ij}\beta_j)^2$ subject to $||\beta||_1 \leq t$
- Similar to *ridge regression*, which has constraint $||\beta||_2 \leq t$
- Lasso does variable selection and shrinkage, while ridge only shrinks.



Diabetes Data



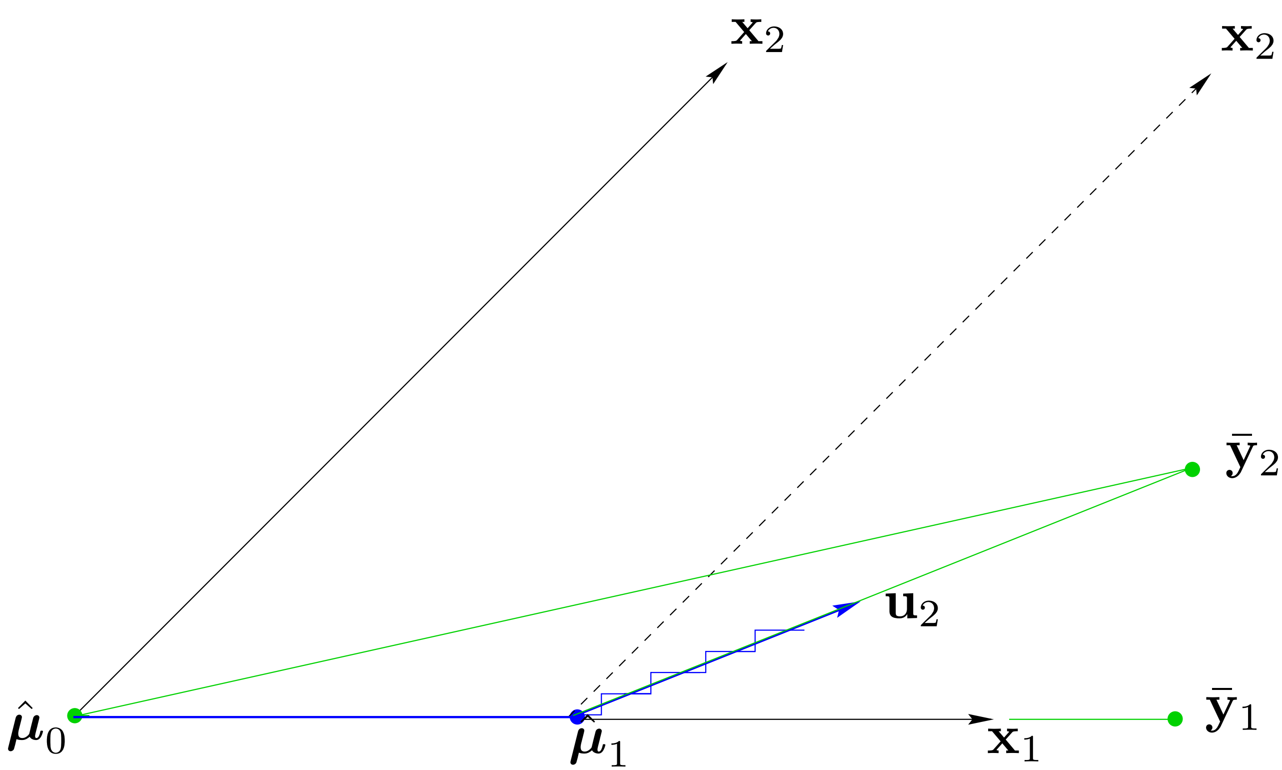
Why are Forward Stagewise and Lasso so similar?

- Are they identical?
- In orthogonal predictor case: *yes*
- In hard to verify case of *monotone* coefficient paths: *yes*
- In general, almost!
- Least angle regression (LAR) provides answers to these questions, and an efficient way to compute the complete Lasso sequence of solutions.

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 β_j in the direction of $\mathrm{sign}(\mathrm{corr}(r, x_j))$ 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_ℓ has as much correlation with the current residual
5. Continue in this way until all predictors have been entered. Stop when $\mathrm{corr}(r, x_j) = 0 \; \forall \; j$, i.e. OLS solution.



The LAR direction $\mathbf{u}_2$ at step 2 makes an equal angle with $\mathbf{x}_1$ and $\mathbf{x}_2$.

Relationship between the 3 algorithms

- Lasso and forward stagewise can be thought of as restricted versions of LAR
- *Lasso*: 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:

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

$$\Leftrightarrow$$

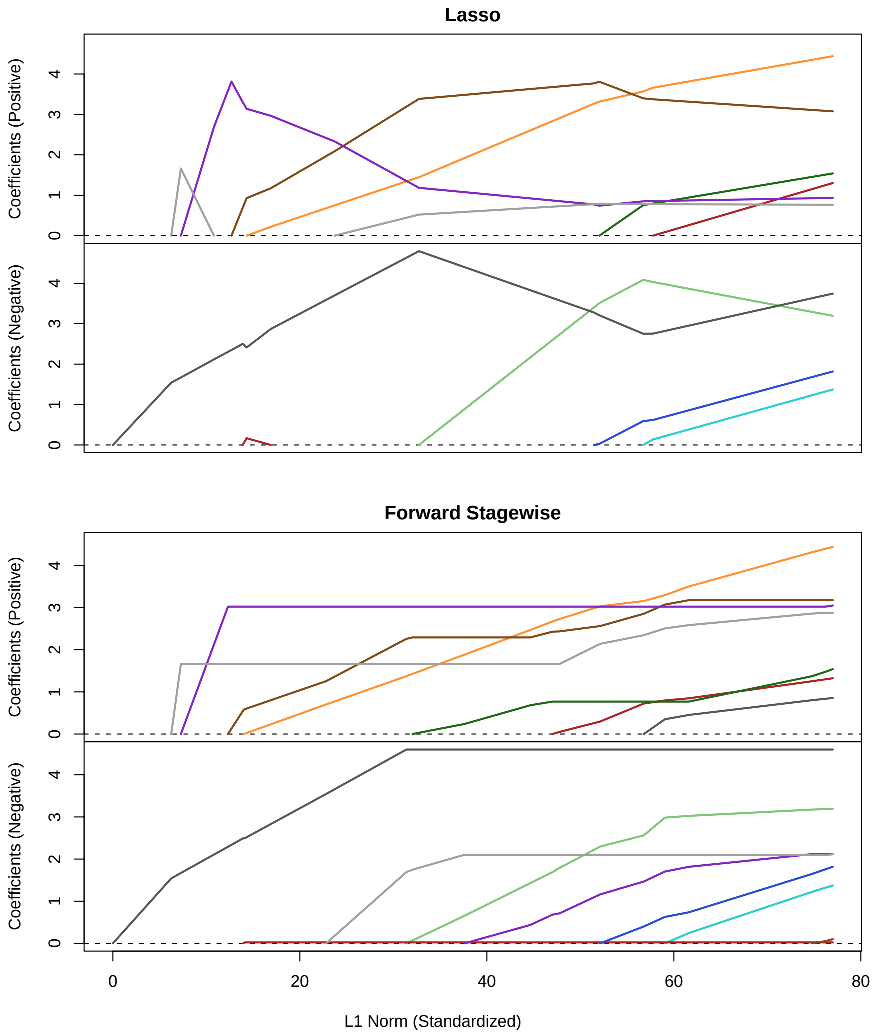
$$\langle \mathbf{x}_j, \mathbf{r} \rangle \quad = \quad \lambda \cdot \mathrm{sign}(\hat{\beta}_j) \quad \text{if } \hat{\beta}_j \neq 0 \text{ (active)}$$

- *Forward Stagewise:* Compute the LAR direction, but constrain the sign of the coefficients to match the correlations $\text{corr}(r, x_j)$.
- 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 LARS algorithm computes the entire Lasso/FS/LAR path in same order of computation as one full least squares fit. Splus/R Software on website:

`www-stat.stanford.edu/~hastie/Papers#LARS`

Forward Stagewise and the Monotone Lasso



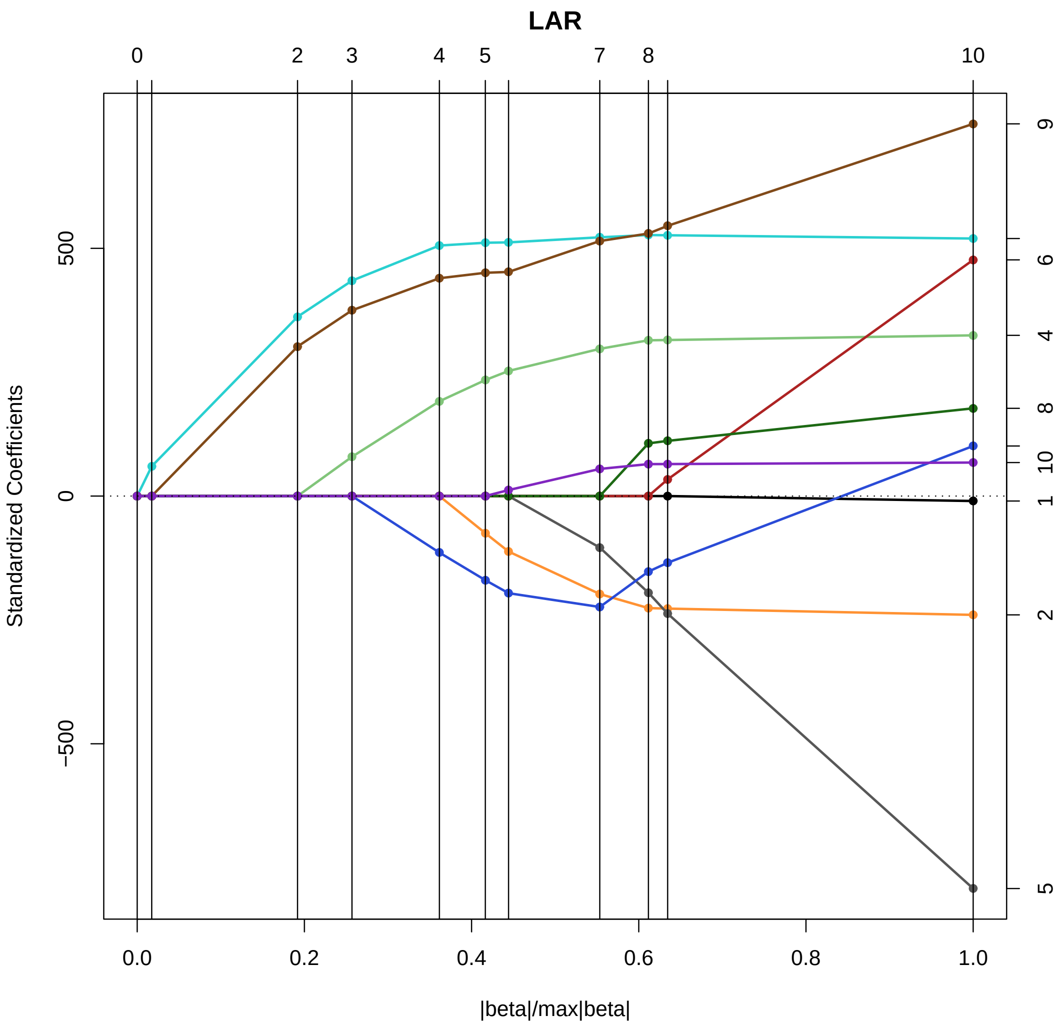
- Expand the variable set to include their negative versions $-x_j$.
- Original lasso corresponds to a *positive* lasso in this enlarged space.
- Forward stagewise corresponds to a *monotone lasso*. The L_1 norm $||\beta||_1$ in this enlarged space is *arc-length*.
- Forward stagewise produces the maximum decrease in loss per unit arc-length in coefficients.

Degrees of Freedom of Lasso

- The *df* or effective number of parameters give us an indication of how much fitting we have done.
- *Stein's Lemma:* If y_i are i.i.d. $N(\mu_i, \sigma^2)$,

$$df(\hat{\boldsymbol{\mu}}) \stackrel{\text{def}}{=} \sum_{i=1}^{n} \text{cov}(\hat{\mu}_i, y_i)/\sigma^2 = E\left[\sum_{i=1}^{n} \frac{\partial \hat{\mu}_i}{\partial y_i}\right]$$

- Degrees of freedom formula for LAR: After k steps, $df(\hat{\boldsymbol{\mu}}_k) = k$ exactly (amazing! with some regularity conditions)
- Degrees of freedom formula for lasso: Let $\hat{df}(\hat{\boldsymbol{\mu}}_\lambda)$ be the number of *non-zero* elements in $\hat{\beta}_\lambda$. Then $E\hat{df}(\hat{\boldsymbol{\mu}}_\lambda) = df(\hat{\boldsymbol{\mu}}_\lambda)$.



df for LAR

- *df* are labeled at the top of the figure
- At the point a competitor enters the active set, the *df* are incremented by 1.
- Not true, for example, for stepwise regression.

Back to Boosting

- Work with Rosset and Zhu (JMLR 2004) extends the connections between Forward Stagewise and L_1 penalized fitting to other loss functions. In particular the Exponential loss of Adaboost, and the Binomial loss of Logitboost.

- In the separable case, L_1 regularized fitting with these losses converges to a L_1 maximizing margin (defined by β^*), as the penalty disappears. i.e. if

$$\beta(t) = \arg\min L(y, f) \quad \text{s.t. } |\beta| \leq t,$$

then

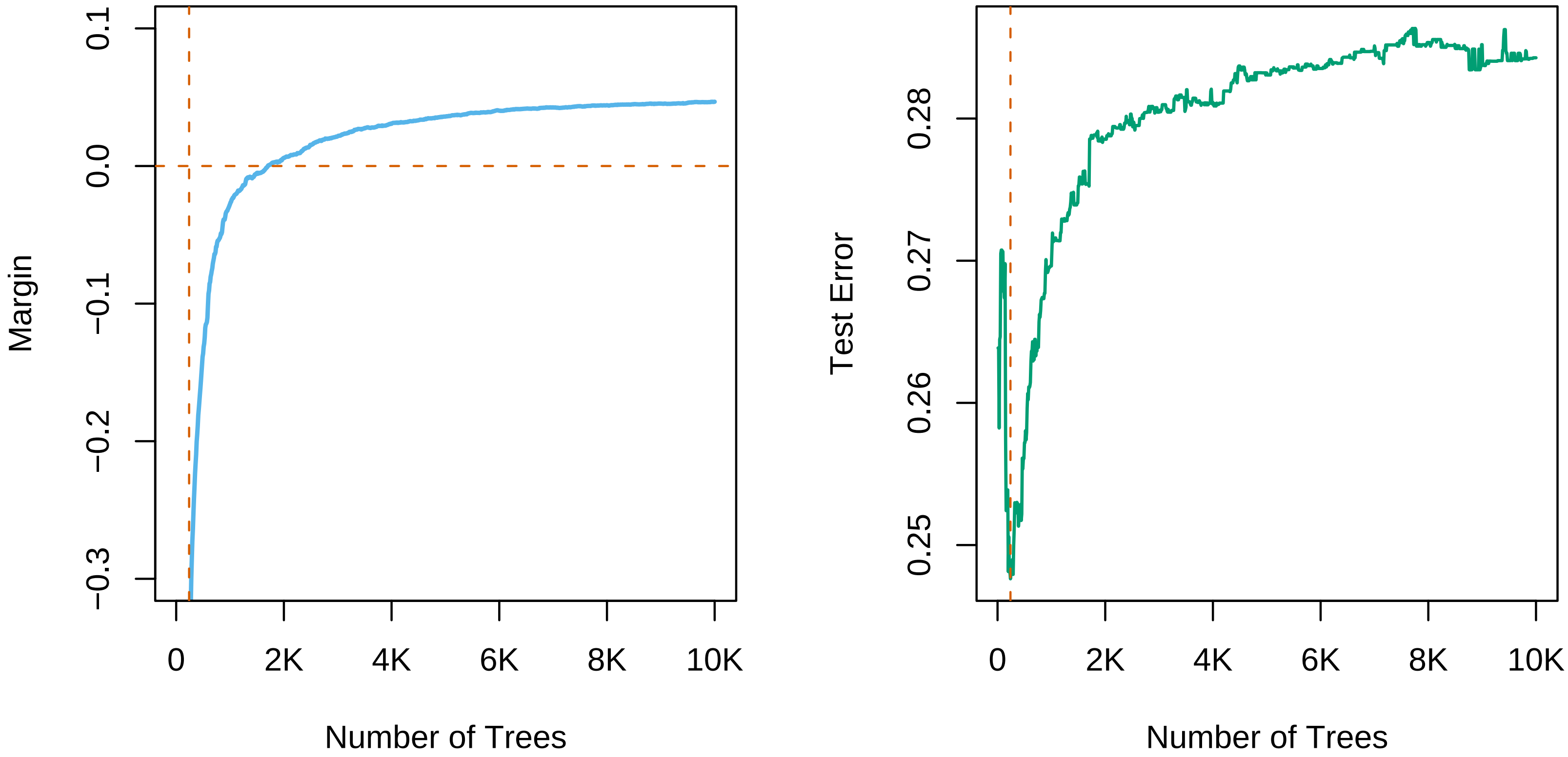
$$\lim_{t \uparrow \infty} \frac{\beta(t)}{|\beta(t)|} \rightarrow \beta^*$$

- Then $\min_i y_i F*(x_i) = \min_i y_i x_i^T \beta^*$, the L_1 margin, is maximized.

- When the monotone lasso is used in the expanded feature space, the connection with boosting (with shrinkage) is more precise.
- This ties in very nicely with the L_1 margin explanation of boosting (Schapire, Freund, Bartlett and Lee, 1998).
- makes connections between SVMs and Boosting, and makes explicit the margin maximizing properties of boosting.
- experience from statistics suggests that some $\beta(t)$ along the path might perform better—a.k.a stopping early.
- Zhao and Yu (2004) incorporate backward corrections with forward stagewise, and produce a boosting algorithm that mimics lasso.

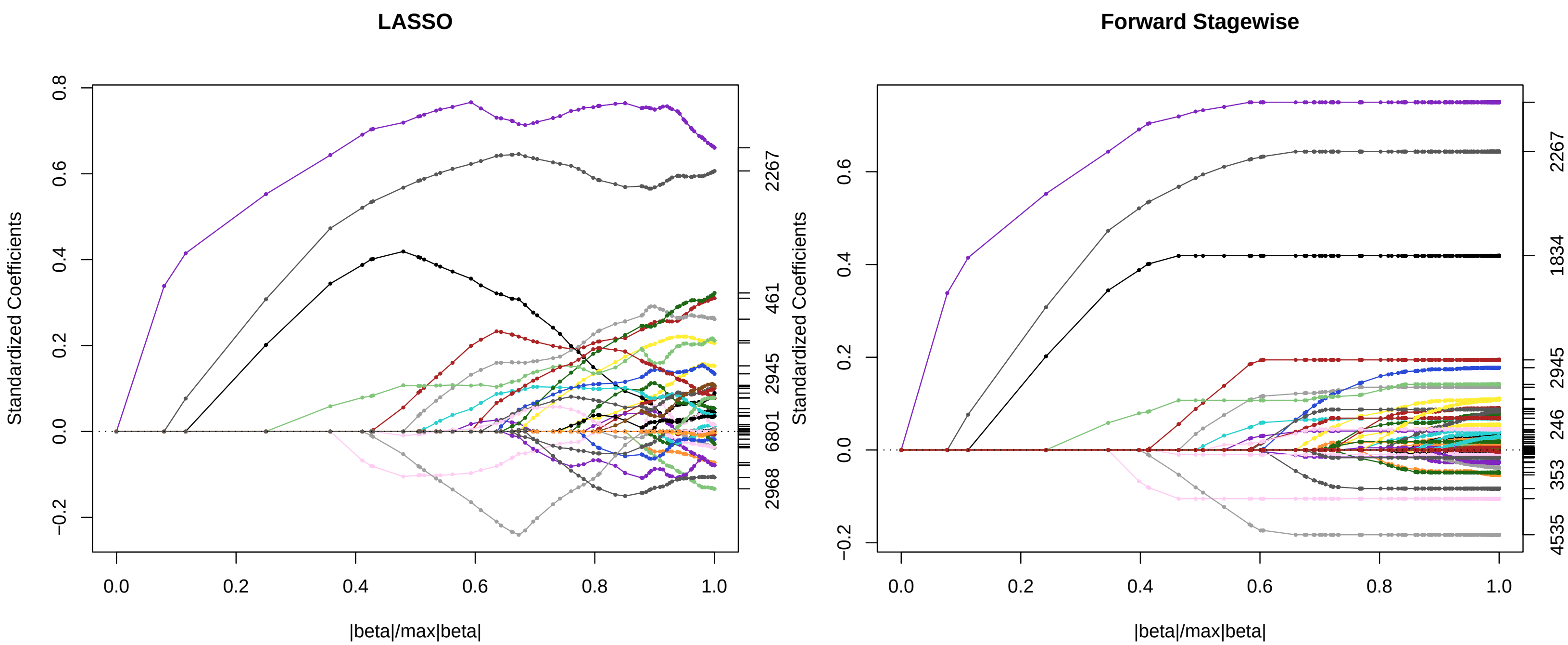
Maximum Margin and Overfitting

Mixture data from ESL. Boosting with 4-node trees, `gbm` package in R, shrinkage = 0.02, Adaboost loss.



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 and stable.

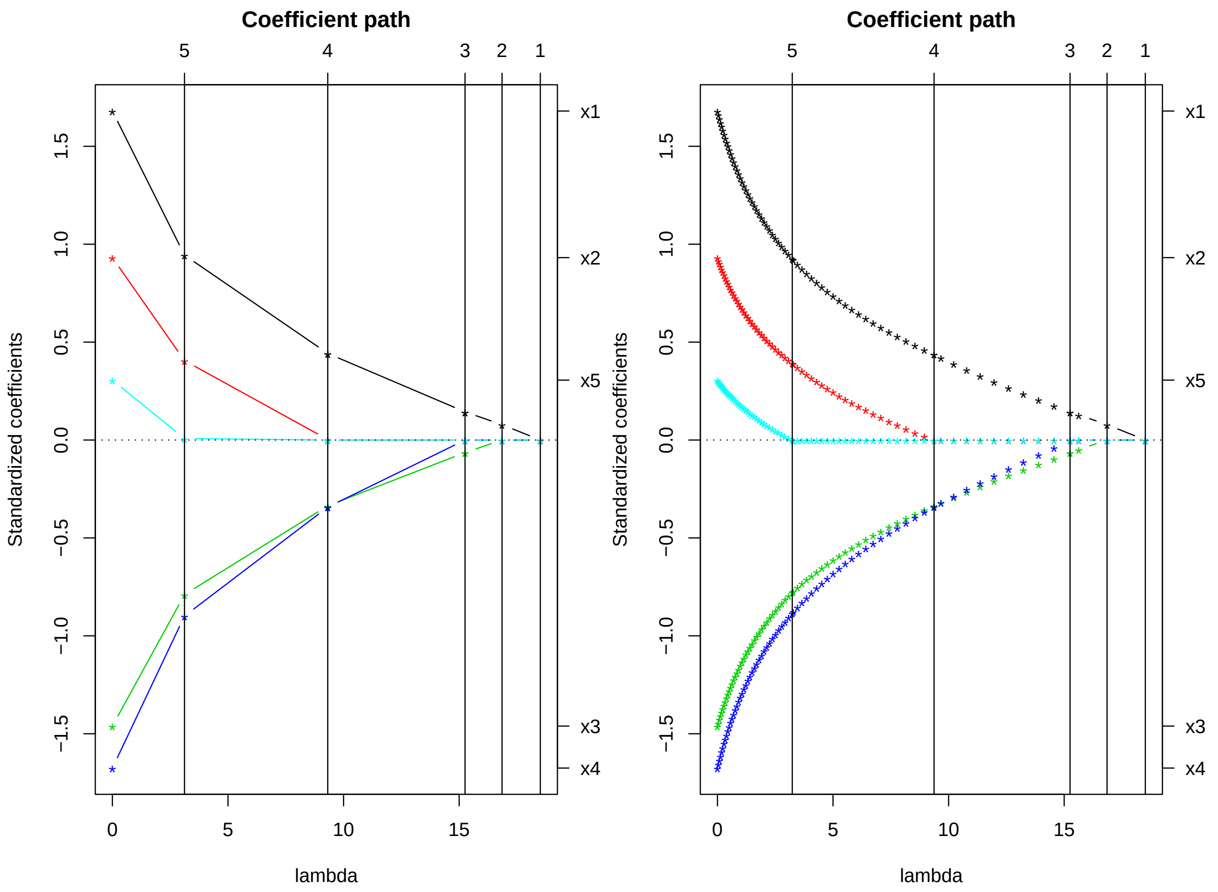


Other Path Algorithms

- *Elasticnet:* (Zhou 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.
- Friedman and Popescu (2004) created *Pathseeker*. It uses an efficient incremental forward-stagewise algorithm with a variety of loss functions. A generalization adjusts the leading k coefficients at each step; $k = 1$ corresponds to forward stagewise, $k = p$ to gradient descent.

- Bach and Jordan (2004) have path algorithms for Kernel estimation, and for efficient ROC curve estimation. The latter is a useful generalization of the Svmpath algorithm discussed later.
- 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.

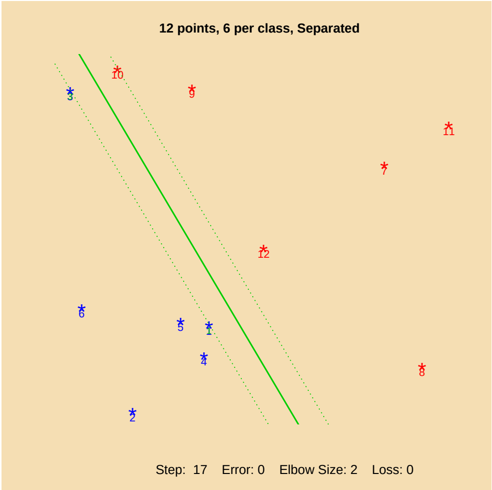
Glmpath

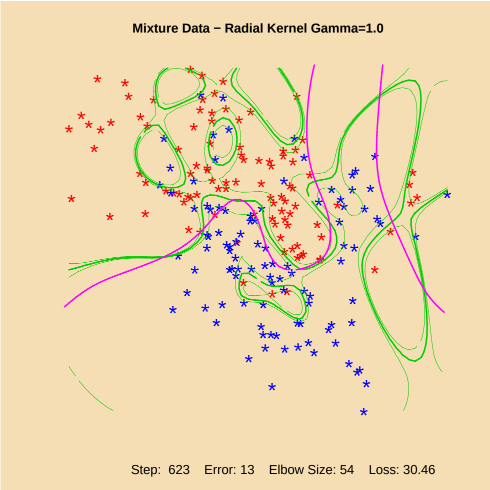


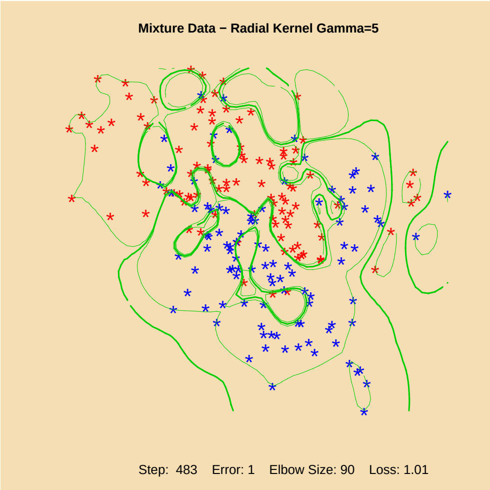
- Approximates the path at the junctions where the active set changes
- Uses predictor corrector methods in convex optimization
- `glmpath` package in R

Path algorithms for the SVM

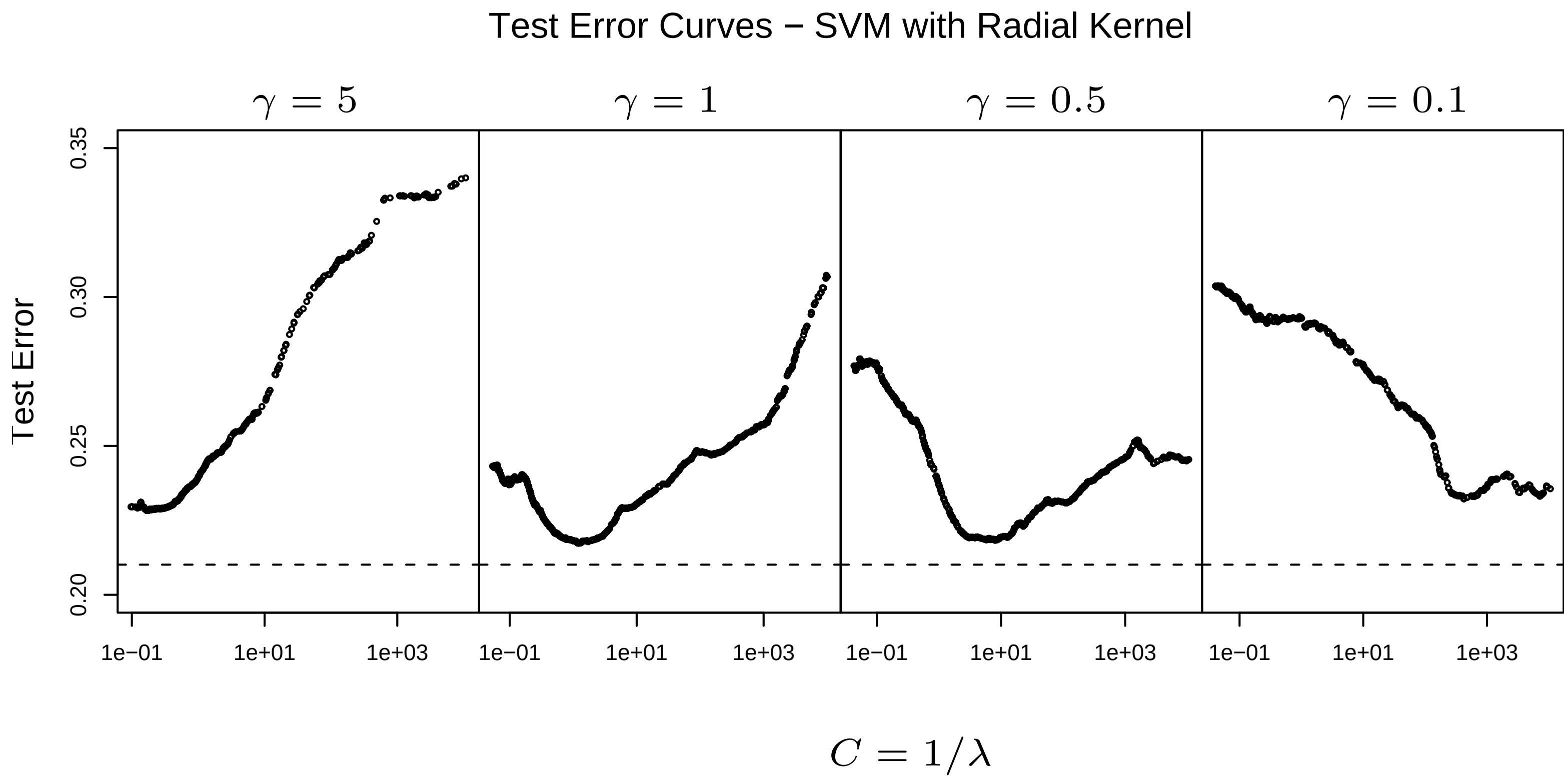
- The two-class SVM classifier $f(X) = \alpha_0 + \sum_{i=1}^{N} \alpha_i K(X, x_i) y_i$ can be seen to have a quadratic penalty and piecewise-linear loss. As the cost parameter C is varied, the *Lagrange multipliers* α_i change piecewise-linearly.
- This allows the entire regularization path to be traced exactly. The active set is determined by the points exactly on the margin.







The Need for Regularization



- γ is a kernel parameter: $K(x, z) = \exp(-\gamma||x - z||^2)$.
- λ (or C) are regularization parameters, which have to be determined using some means like cross-validation.

- Using logistic regression + binomial loss or Adaboost exponential loss, and same quadratic penalty as SVM, we get the same limiting margin as SVM (Rosset, Zhu and Hastie, JMLR 2004)
- Alternatively, using the "Hinge loss" of SVMs and an L_1 penalty (rather than quadratic), we get a *Lasso* version of SVMs (with at most N variables in the solution for any value of the penalty.

Concluding Comments

- *Boosting fits a monotone L_1 regularization path towards a maximum-margin classifier*
- Many modern function estimation techniques create a path of solutions via regularization.
- In many cases these paths can be computed efficiently and entirely.
- This facilitates the important step of model selection — selecting a desirable position along the path — using a test sample or by CV.