Prediction, Estimation, and Attribution

Bradley Efron

brad@stat.stanford.edu

Department of Statistics
Stanford University

Regression

Gauss (1809), Galton (1877)

- Prediction

  random forests, boosting, support vector machines, neural nets, deep learning

- Estimation

  OLS, logistic regression, GLM: MLE

- Attribution (significance)

  ANOVA, lasso, Neyman–Pearson

Estimation

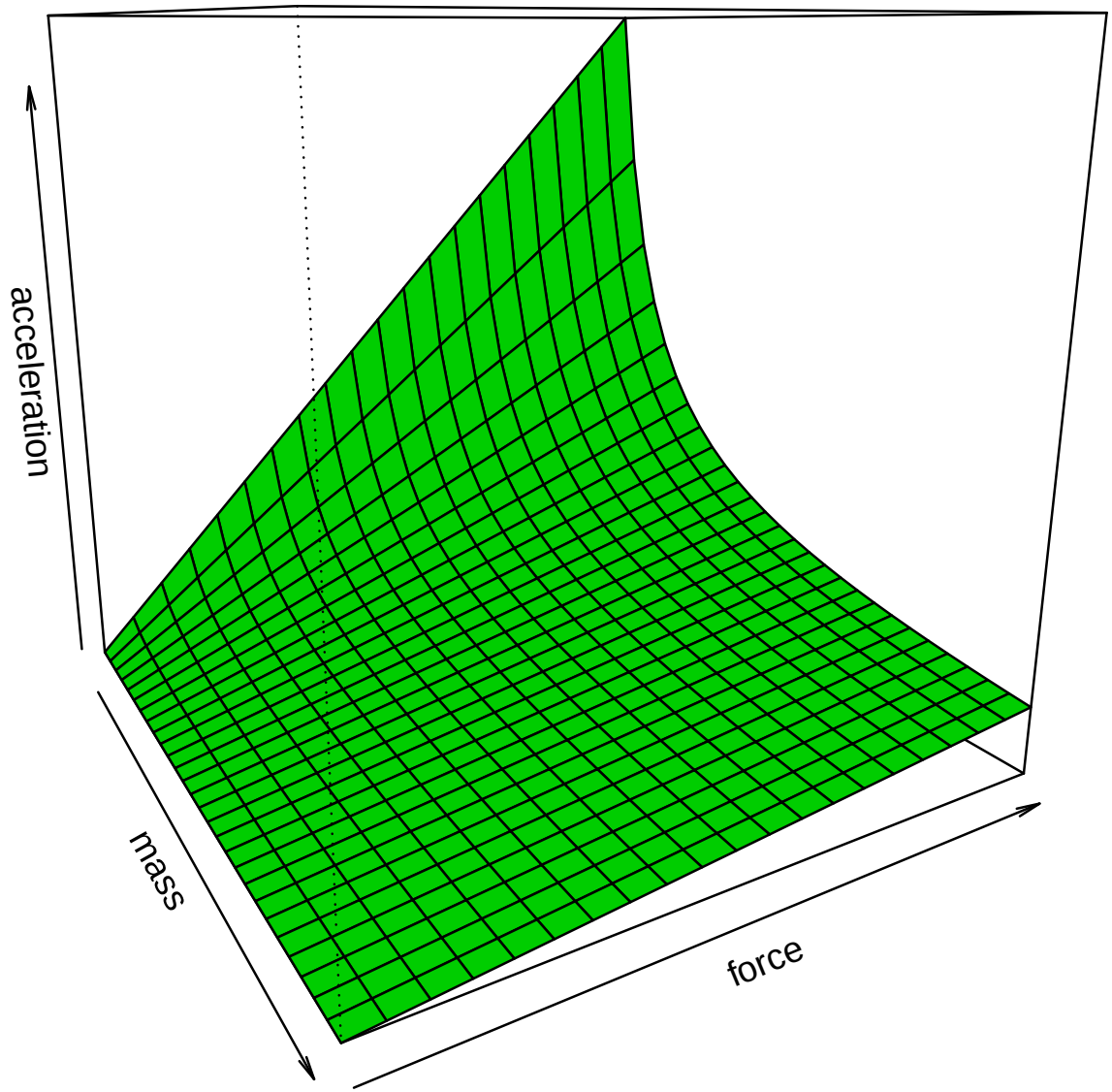
Normal Linear Regression

- Observe
  - $y_i = \mu_i + \epsilon_i$ for $i = 1, \ldots, n$
  - $\mu_i = x_i^t \beta$
  - x_i a p-dimensional covariate
  - $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$
  - β unknown

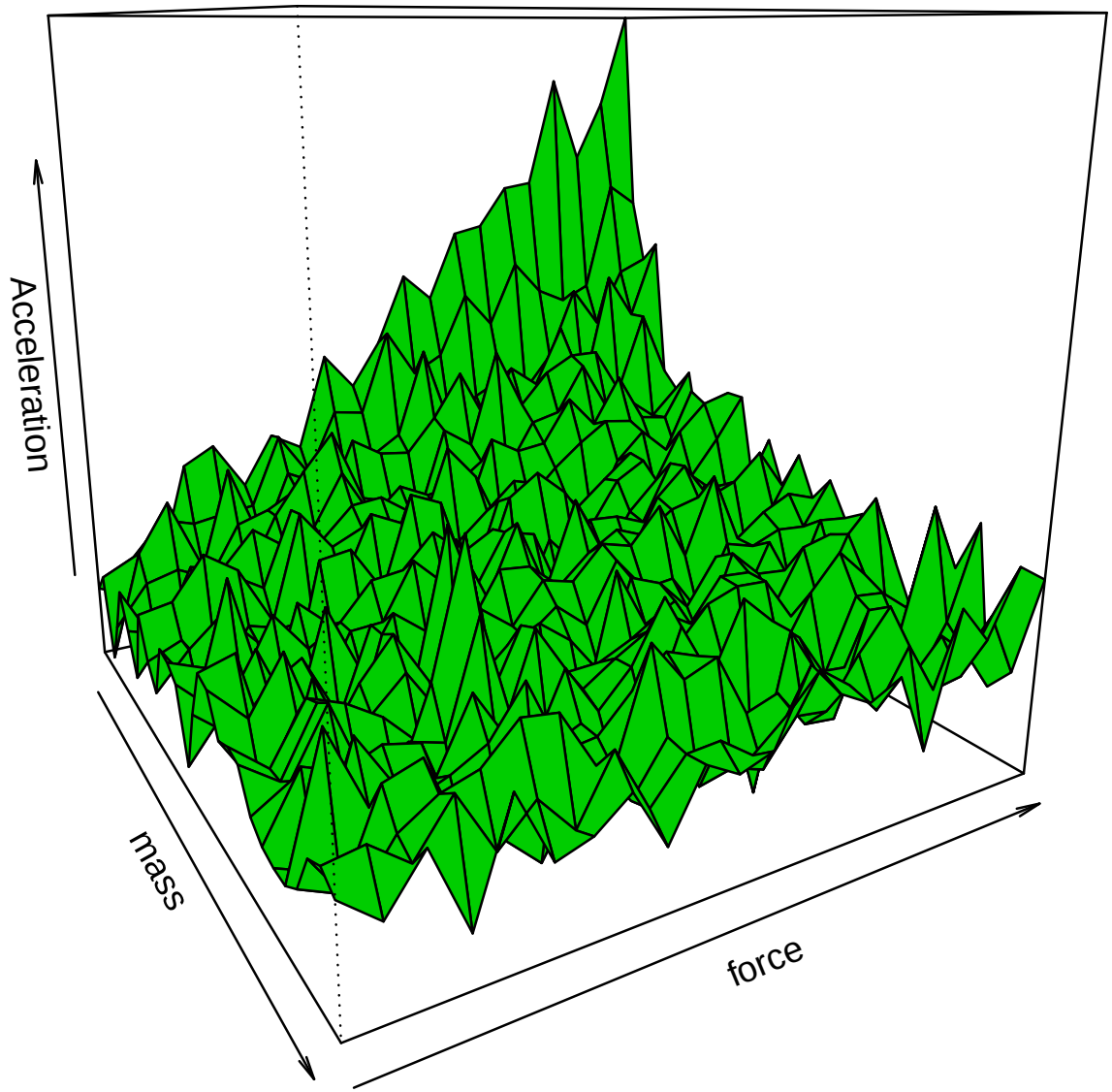
$$\underset{n}{y} = \underset{n \times p}{\boldsymbol{X}} \; \underset{p}{\beta} + \underset{n}{\epsilon}$$

- *Surface plus noise* $\quad y = \mu(x) + \epsilon$
- Surface $\{\mu(x), x \in \mathcal{X}\}$: codes scientific truth (hidden by noise)
- Newton's second law $\quad$ acceleration = force / mass

Newton's 2nd law: acceleration=force/mass



If Newton had done the experiment



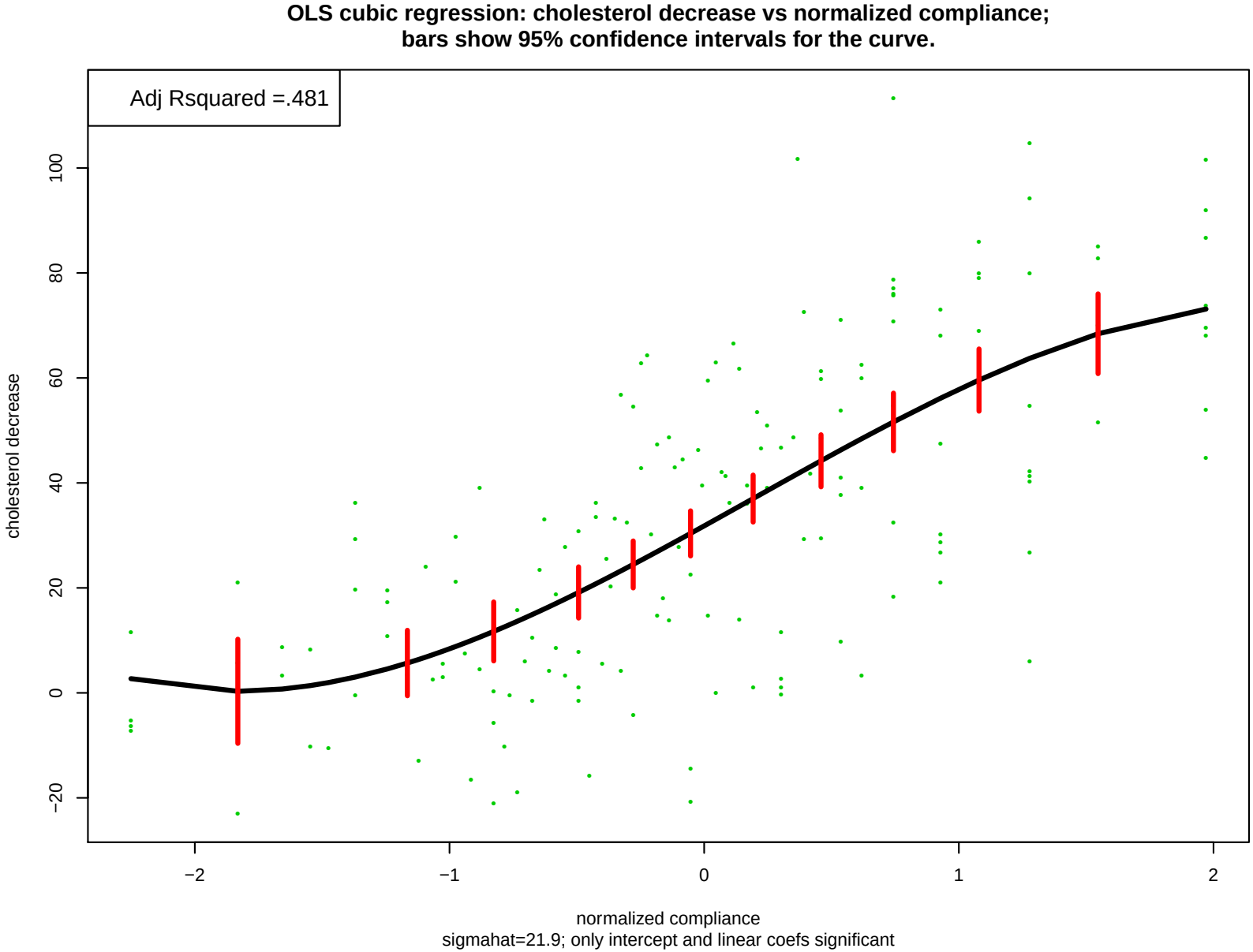
Example

The Cholesterol Data

- $n = 164$ men took cholostyramine
- Observe (c_i, y_i)
  - c_i = normalized compliance (how much taken)
  - y_i = reduction in cholesterol
- Model $\quad y_i = x_i^t \beta + \epsilon_i$

$$x_i^t = (1, c_i, c_i^2, c_i^3) \qquad \epsilon_i \sim \mathcal{N}(0, \sigma^2)$$

- $n = 164$, $p = 4$


OLS cubic regression: cholesterol decrease vs normalized compliance;
bars show 95% confidence intervals for the curve.
Adj Rsquared =.481
cholesterol decrease
100
80
60
40
20
0
−20
−2
−1
0
1
2
normalized compliance
sigmahat=21.9; only intercept and linear coefs significant

Neonate Example

- $n = 800$ babies in an African facility
- 600 lived, 200 died
- 11 covariates: apgar score, body weight, . . .
- Logistic regression $n = 800$, $p = 11$

$$\texttt{glm}(\underset{800}{\boldsymbol{y}} \sim \underset{800\times 11}{\boldsymbol{X}}, \text{binomial})$$

  - $y_i = 1$ or 0 as baby dies or lives
  - $x_i = i$th row of $\boldsymbol{X}$ (vector of 11 covariates)
- Linear logistic surface, Bernoulli noise

Output of logistic regression program

predictive error 15%

	estimate	st.error	z-value	p-**value**
gest	−.474	.163	−2.91	**.004****
ap	−.583	.110	−5.27	**.000*****
bwei	−.488	.163	−2.99	**.003****
resp	.784	.140	5.60	**.000*****
cpap	.271	.122	2.21	**.027***
ment	1.105	.271	4.07	**.000*****
rate	−.089	.176	−.507	**.612**
hr	.013	.108	.120	**.905**
head	.103	.111	.926	**.355**
gen	−.001	.109	−.008	**.994**
temp	.015	.124	.120	**.905**

Prediction Algorithms

Random Forests, Boosting, Deep Learning, . . .

- Data $d = \{(x_i, y_i), i = 1, 2, \ldots, n\}$
  - y_i = response
  - x_i = vector of p predictors

  (Neonate: $n = 800$, $p = 11$, $y = 0$ or 1)

- Prediction rule $f(x, d)$

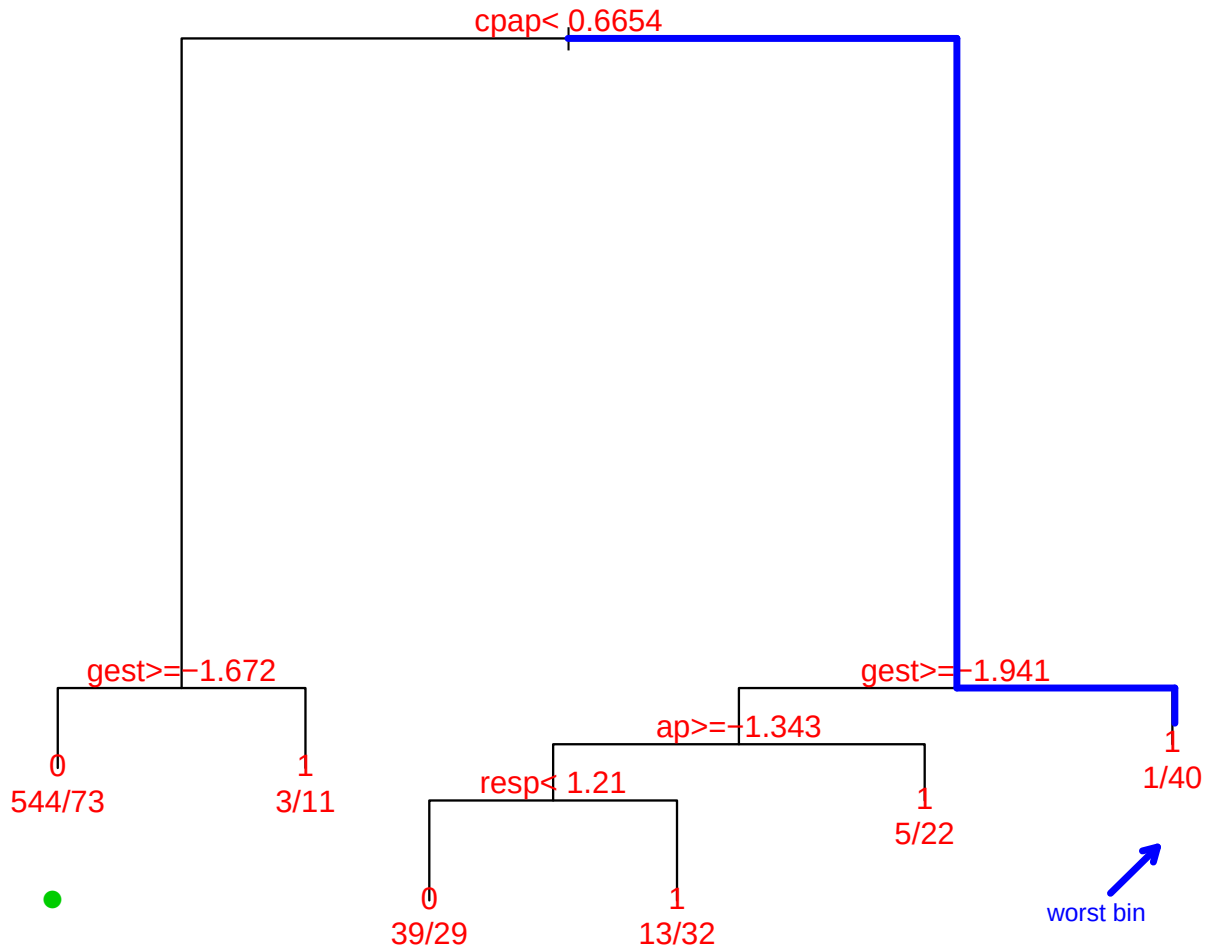
  New $(x, ?)$ gives $\hat{y} = f(x, d)$

- *Strategy* Go directly for high predictive accuracy; forget (mostly) about surface + noise
- Machine learning

Classification Using Regression Trees

- n cases: $n_0 =$ "0" and $n_1 =$ "1"
- p predictors (features)
  (Neonate: $n = 800, n_0 = 600, n_1 = 200, p = 11$)
- *Split into two groups* with predictor and split value chosen to maximize difference in rates
- Then split the splits, etc. . . (some stopping rule)

Classification Tree: 800 neonates, 200 died
(<<−− lived died −−>>)



Random Forests

Breiman (2001)

1. Draw a bootstrap sample of original n cases
2. Make a classification tree from the bootstrap data set *except* at each split use only a random subset of the p predictors
3. Do all this lots of times (≈ 1000)
4. Prediction rule For any new x predict $\hat{y}$ = majority of the 1000 predictions

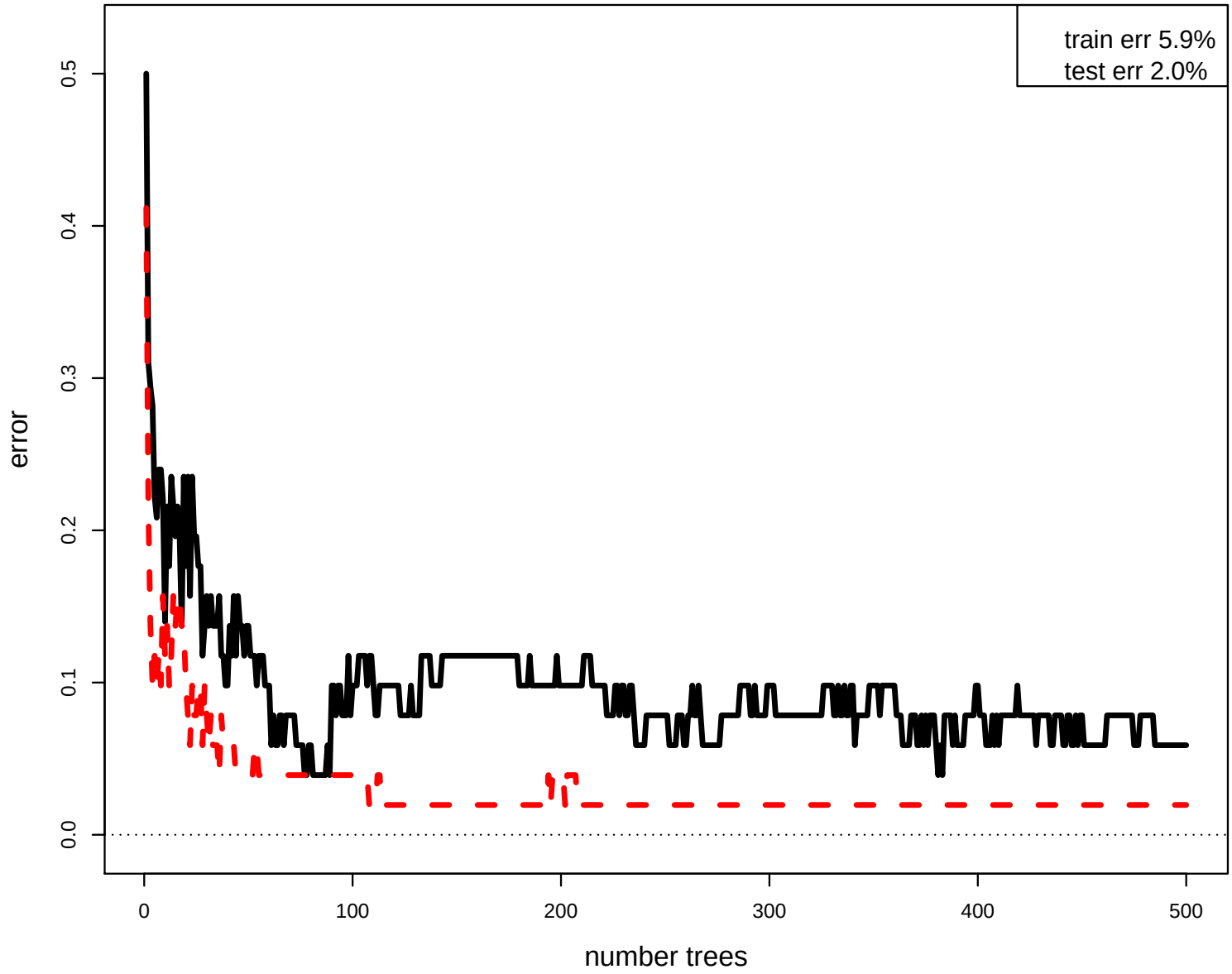
The Prostate Cancer Microarray Study

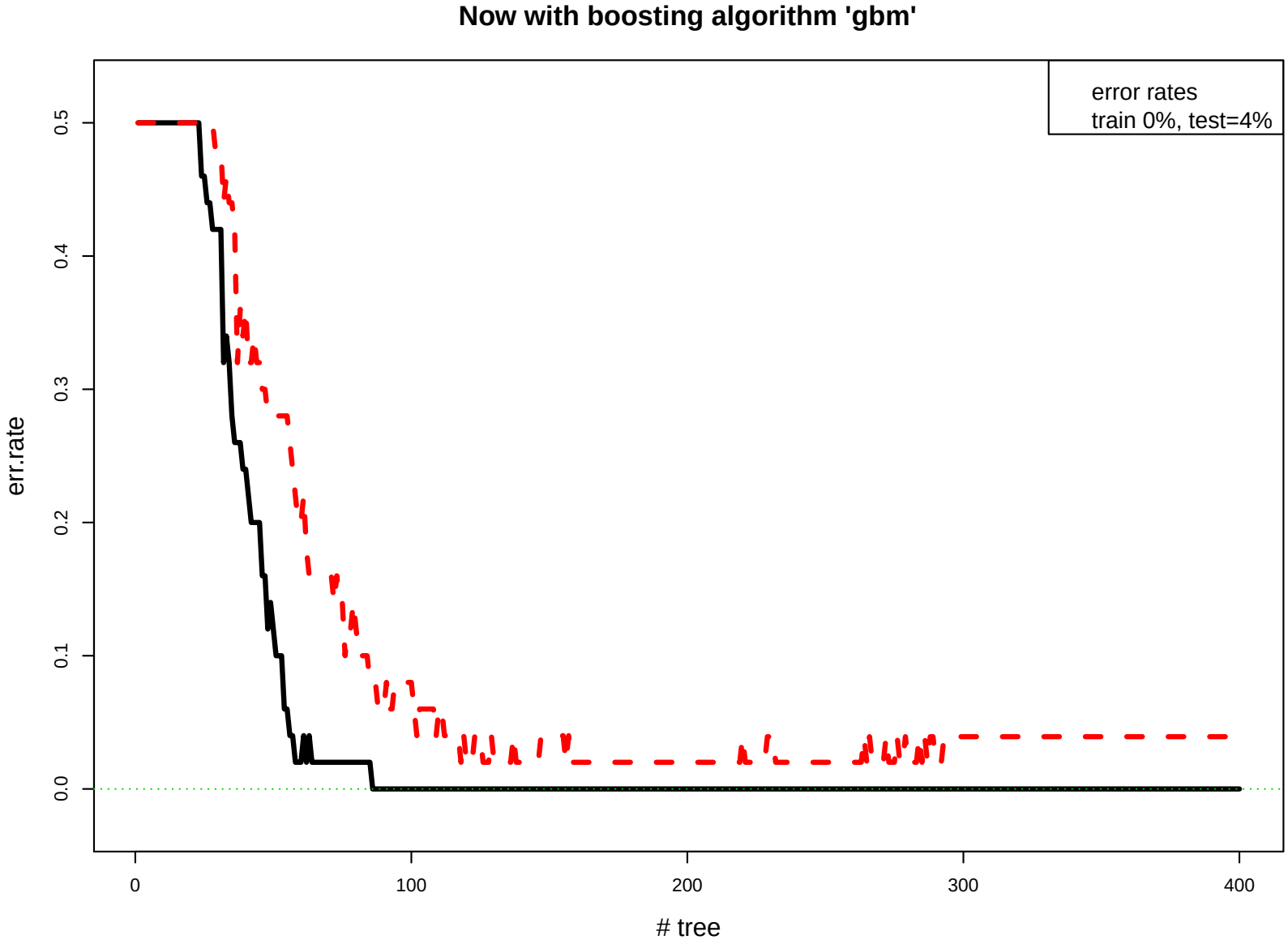
- $n = 100$ men: 50 prostate cancer, 50 normal controls
- For each man measure activity of $p = 6033$ genes
- Data set $\boldsymbol{d}$ is 100 × 6033 matrix ("wide")
- Wanted: Prediction rule $f(x, \boldsymbol{d})$ that inputs new 6033-vector x and outputs $\hat{y}$ correctly predicting cancer/normal

Random Forests

for Prostate Cancer Prediction

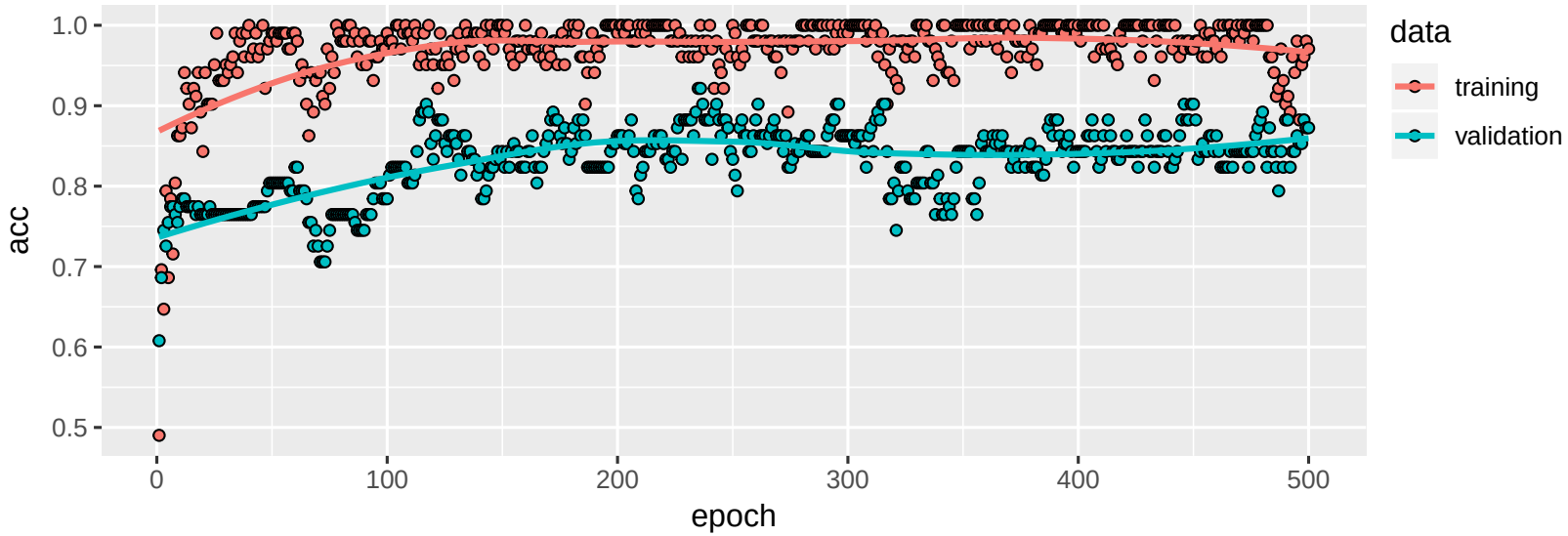
- Randomly divide the 100 subjects into
  - "training set" of 50 subjects $(25 + 25)$
  - "test set" of the other 50 $(25 + 25)$
- Run R program `randomforest` on the training set
- Use its rule $f(x, d_{\text{train}})$ on the test set and see how many errors it makes


Prostate cancer prediction using random forests
Black is cross−validated training error, Red is test error rate
train err 5.9%
test err 2.0%
error
0.0
0.1
0.2
0.3
0.4
0.5
0
100
200
300
400
500
number trees


Now with boosting algorithm 'gbm'
error rates
train 0%, test=4%
err.rate
0.0
0.1
0.2
0.3
0.4
0.5
0
100
200
300
400
tree

Now using deep learning ("Keras")

parameters = 780,738



Prediction is Easier than Estimation

- Observe
  - $x_1, x_2, x_3, \ldots, x_{25} \overset{\text{ind}}{\sim} \mathcal{N}(\mu, 1)$
  - $\bar{x} =$ mean, $\overset{\perp}{x} =$ median

- Estimation

$$E\left\{(\mu - \overset{\perp}{x})^2\right\} \Big/ E\left\{(\mu - \bar{x})^2\right\} = 1.57$$

- Wish to predict new $X_0 \sim \mathcal{N}(\mu, 1)$

- Prediction

$$E\left\{(X_0 - \overset{\perp}{x})^2\right\} \Big/ E\left\{(X_0 - \bar{x})^2\right\} = 1.02$$

Prediction is Easier than Attribution

- *Microarray study* N genes: $z_j \overset{\text{ind}}{\sim} \mathcal{N}(\delta_j, 1),\ j = 1, 2, \ldots, N$
  - $N_0 : \delta_j = 0$ (null genes)
  - $N_1 : \delta_j > 0$ (non-null)
- New subject's microarray: $x_j \sim \mathcal{N}(\pm\delta_j, 1)$ $\begin{cases} + & \text{sick} \\ - & \text{healthy} \end{cases}$

- Prediction

$$\text{Possible if } N_1 = O\left(N_0^{1/2}\right)$$

- Attribution

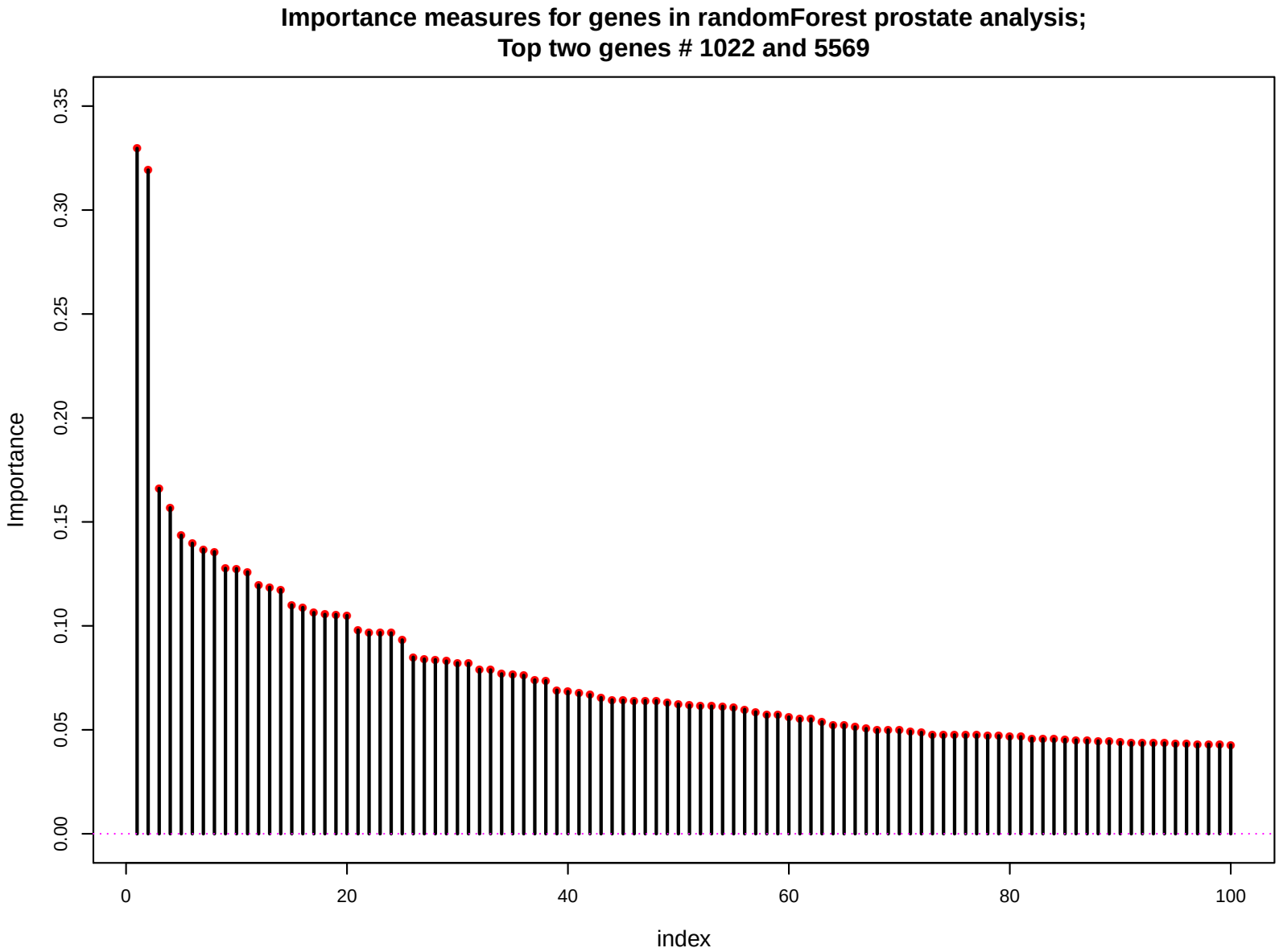
$$\text{Requires } N_1 = O(N_0)$$

- Prediction allows accrual of "weak learners"

Prediction and Medical Science

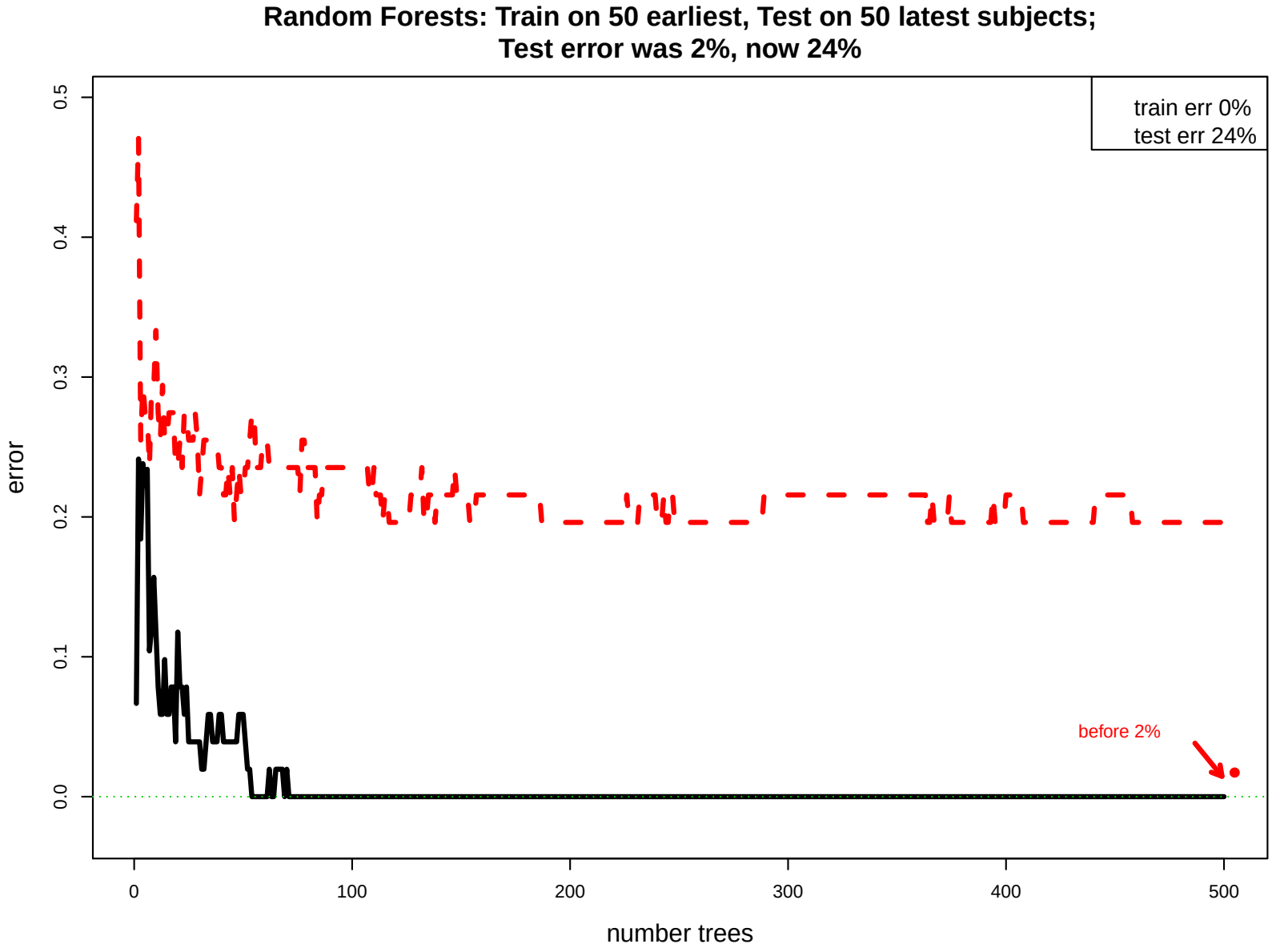
- Random forest test set predictions made only 1 error out of 50!
- Promising for diagnosis
- Not so much for scientific understanding
- Next

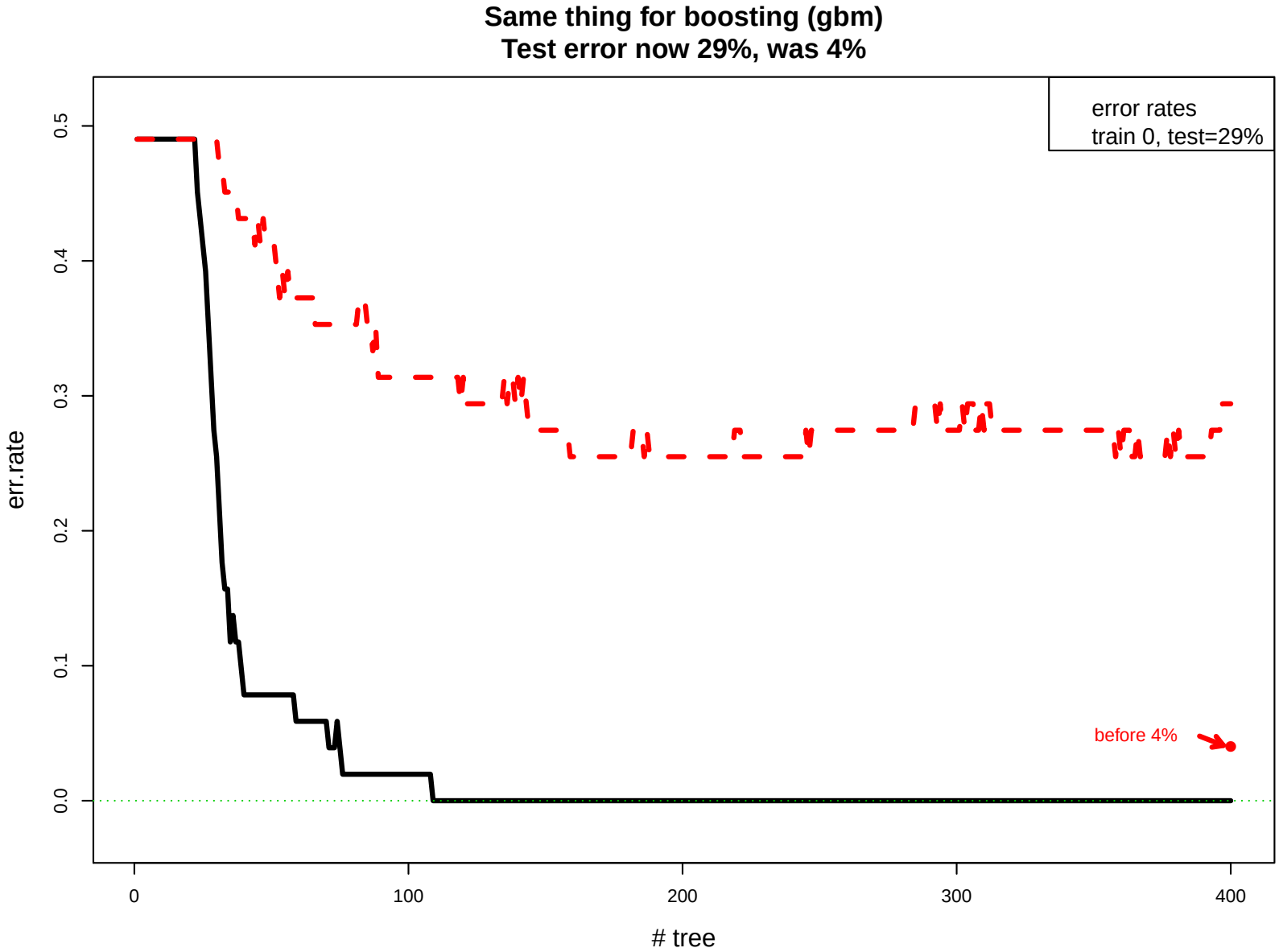
  "Importance measures" for the predictor genes


Importance measures for genes in randomForest prostate analysis;
Top two genes # 1022 and 5569
Importance
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0
20
40
60
80
100
index

Were the Test Sets Really a Good Test?

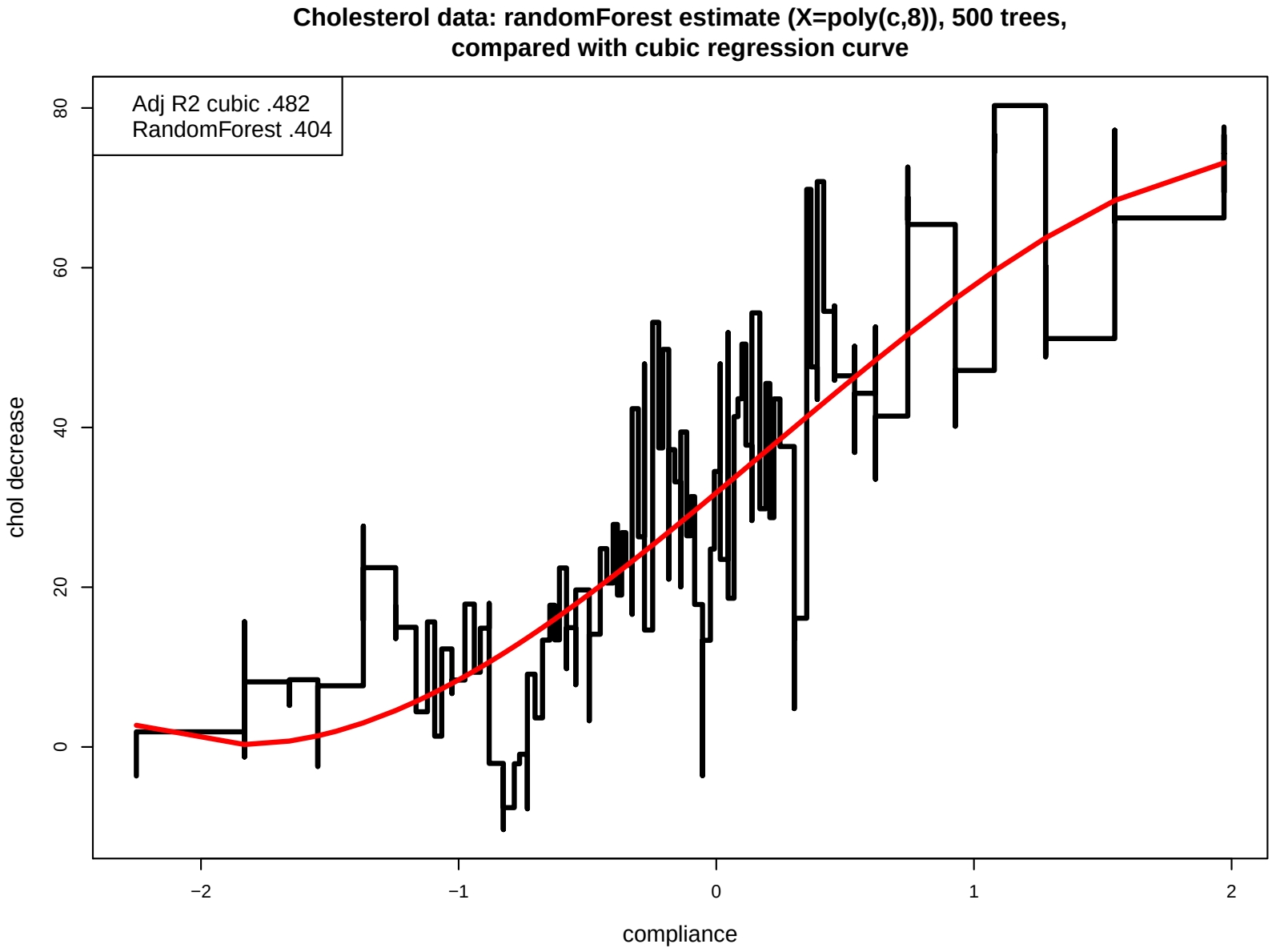
- Prediction can be highly context-dependent and fragile
- *Before* Randomly divided subjects into "training" and "test"
- Next
  - 50 earliest subjects for training
  - 50 latest for test
  - both $25 + 25$

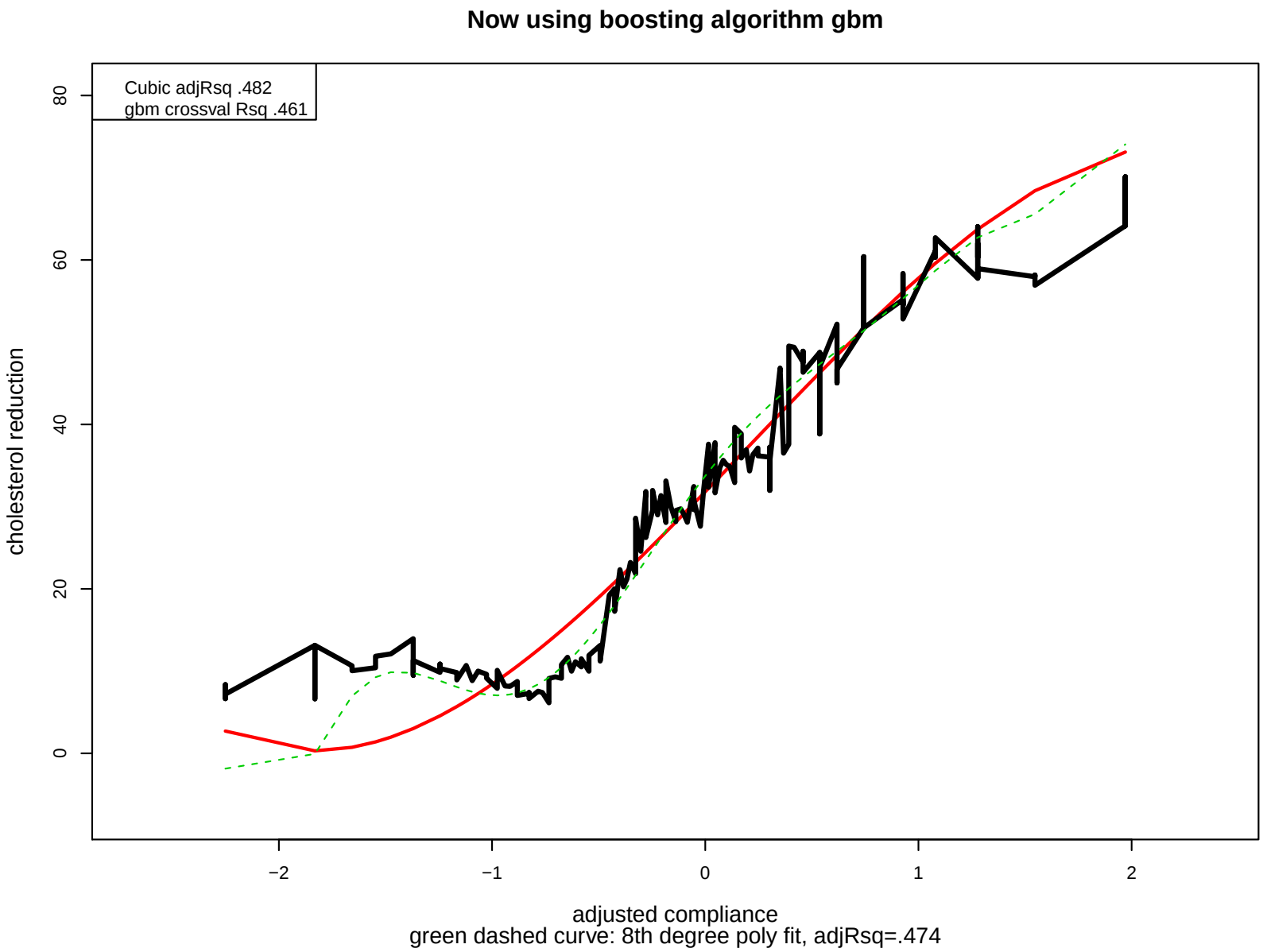

Random Forests: Train on 50 earliest, Test on 50 latest subjects;
Test error was 2%, now 24%
train err 0%
test err 24%
before 2%
error
0.0
0.1
0.2
0.3
0.4
0.5
0
100
200
300
400
500
number trees


Same thing for boosting (gbm)
Test error now 29%, was 4%
error rates
train 0, test=29%
before 4%
err.rate
0.0
0.1
0.2
0.3
0.4
0.5
0
100
200
300
400
tree

Truth, Accuracy, and Smoothness

- Estimation and Attribution: seek long-lasting scientific truths
  - physics
  - astronomy
  - medicine
  - economics?
- Prediction algorithms: truths and ephemeral relationships
  - credit scores
  - movie recommendations
  - image recognition
- Estimation and Attribution: theoretical optimality (MLE, Neyman–Pearson)
- Prediction training-test performance
- *Nature:* rough or smooth?


Cholesterol data: randomForest estimate (X=poly(c,8)), 500 trees,
compared with cubic regression curve
Adj R2 cubic .482
RandomForest .404
chol decrease
0
20
40
60
80
−2
−1
0
1
2
compliance


Now using boosting algorithm gbm
Cubic adjRsq .482
gbm crossval Rsq .461
cholesterol reduction
0
20
40
60
80
−2
−1
0
1
2
adjusted compliance
green dashed curve: 8th degree poly fit, adjRsq=.474

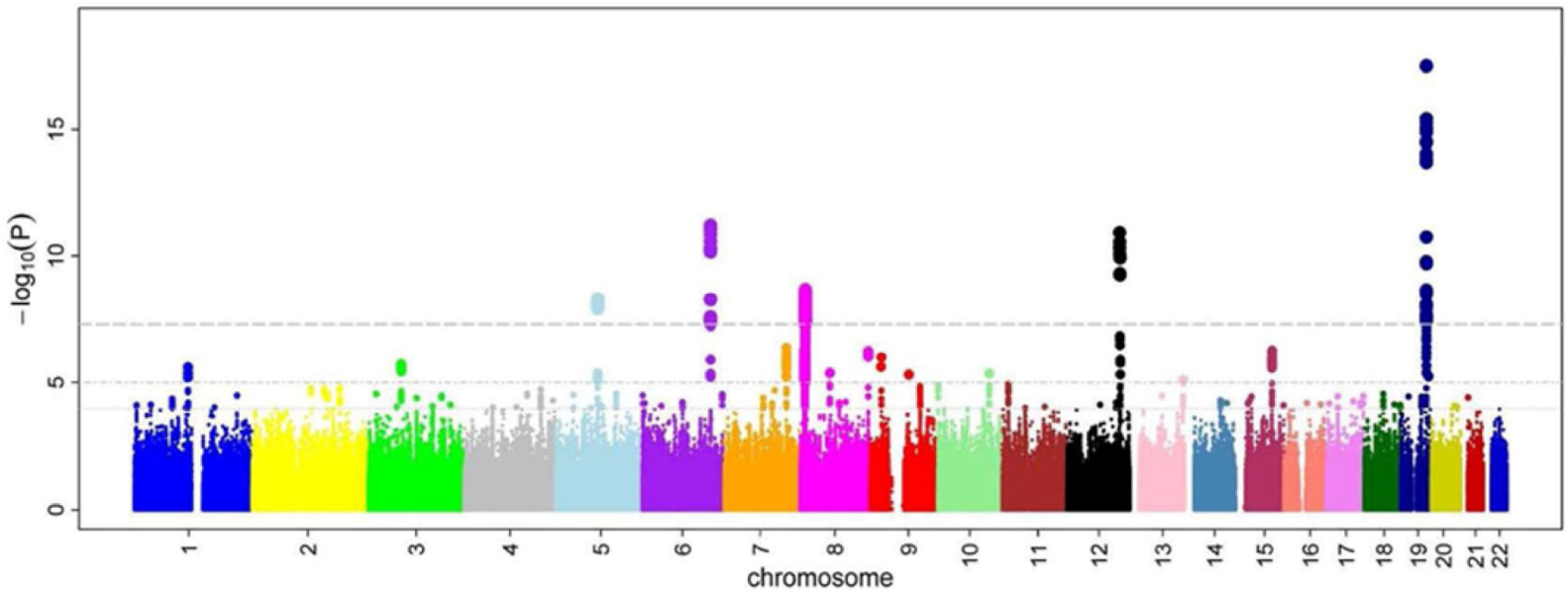
Estimation v. Prediction Algorithms

1	Surface plus noise	Direct prediction
2	Scientific truth (eternal or at least long-lasting)	Empirical prediction efficiency (could be ephemeral, e.g., commerce)
3	$\underset{n\times p}{\boldsymbol{X}}$: $p < n$ (p moderate)	$p > n$ (both possibly huge, "$n =$ all")
4	$\boldsymbol{X}$ chosen parsimoniously (main effects $\gg$ interactions)	Anti-parsimony (algorithms expand $\boldsymbol{X}$)
5	Parametric modeling (condition on x's; smoothness)	Mostly nonparametric ((x, y) pairs iid)
6	Homogeneous data (RCT)	Very large heterogeneous data sets
7	Theory of optimal estimation (MLE)	Training and test sets (CTF, asymptotics)

Estimation and Attribution

in the Wide-Data Era

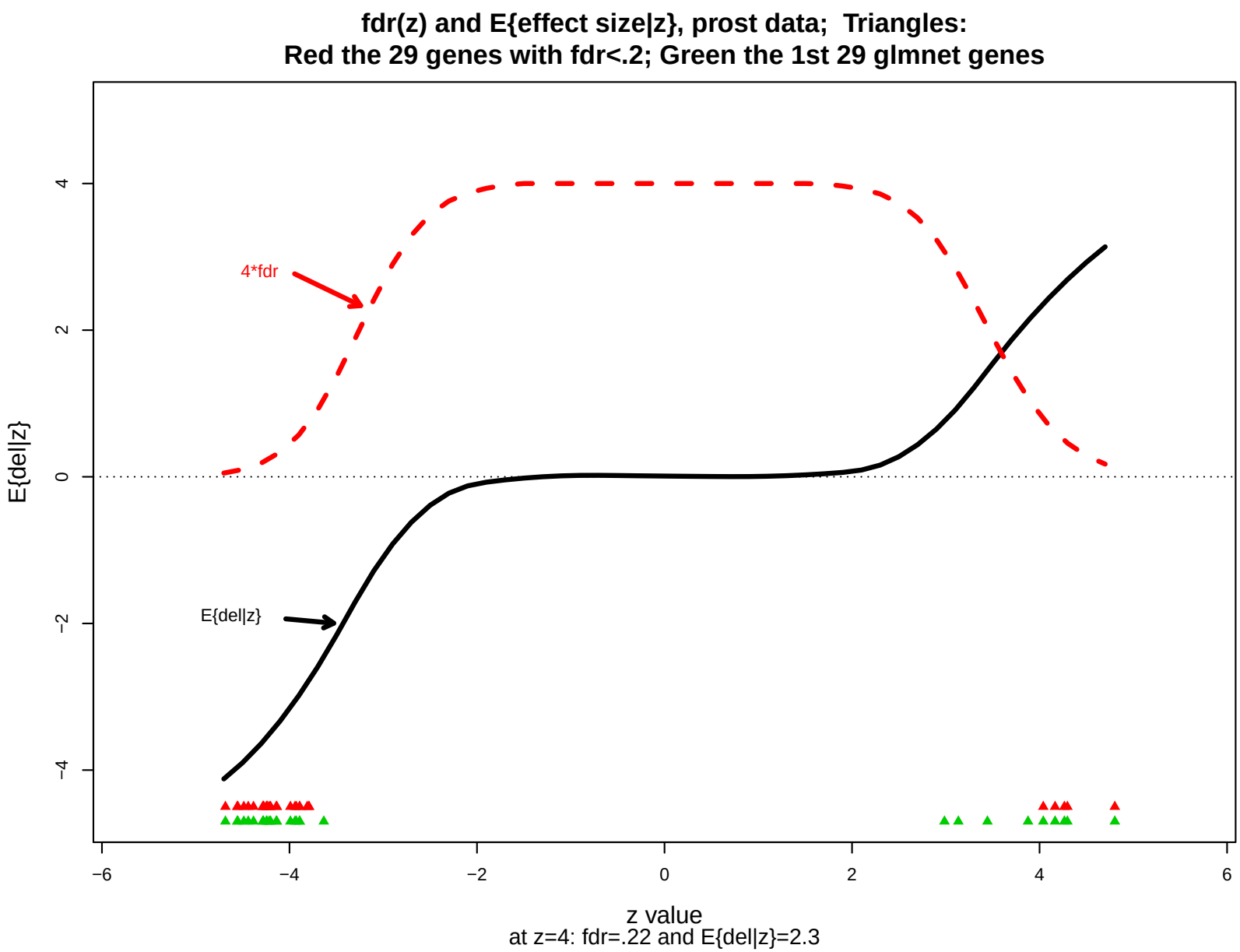
- Large p (the number of features) affects Estimation
  - MLS can be badly biased for individual parameters
  - "surface" if, say, $p = 6033$?
- Attribution still of interest
- *GWAS* $n = 4000, p = 500,000$
- Two-sample p-values for each SNP
- Plotted: $-\log_{10}(p)$


$-\log_{10}(P)$
0
5
10
15
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
chromosome

Attribution and Estimation

for the Prostate Cancer Study

- $\underset{n\times p}{\boldsymbol{X}}$: $n = 100$ men $(50 + 50)$, $p = 6033$ genes
  - gene$_i$ gives $z_i \sim \mathcal{N}(\delta_i, 1)$
  - δ_i = effect size
- *Local false discovery rate* $\text{fdr}(z_i) = \Pr\{\delta_i = 0 \mid z_i\}$
- Effect size estimate $E(z_i) = E\{\delta_i \mid z_i\}$
  - Bayes and empirical Bayes
  - `locfdr`


fdr(z) and E{effect size|z}, prost data; Triangles:
Red the 29 genes with fdr<.2; Green the 1st 29 glmnet genes
4*fdr
E{del|z}
E{del|z}
z value
at z=4: fdr=.22 and E{del|z}=2.3

Sparse Models and the Lasso

- We want to use OLS — $\min \|y - X\beta\|^2$ — but p is too big
- *Instead* minimize $\|y - X\beta\|^2 + \lambda \sum_1^p \left|\hat{\beta}_j\right|$
  - Large λ gives sparse $\hat{\beta}$
  - `glmnet` does this for logistic regression
- In between classical OLS and boosting algorithms
- Have it both ways?

Two Trends

- Making prediction algorithms better for scientific use
  - smoother
  - more interpretable
  - less brittle
- Making traditional estimation/attribution methods better for large-scale (n, p) problems
  - less fussy
  - more flexible
  - better scaled

References

Algorithms Hastie, Tibshirani and Friedman (2009). *The Elements of Statistical Learning* 2nd ed.

Random Forests Breiman (2001). "Random forests."

Data Science Donoho (2015). "50 years of data science."

CART Breiman, Friedman, Olshen and Stone (1984). *Classification and Regression Trees*.

locfdr Efron (2010). *Large-Scale Inference*.

Lasso & glmnet Friedman, Hastie and Tibshirani (2010). "Regularization paths for generalized linear models via coordinate descent."