Ch7 Model assessment and selection: part I

STAT 6312
Department of mathematics, UTA

ESL 7.1, 7.2, 7.4, 7.5, 7.7, 7.10

Optional reading: ESL 7.10.2

Test error

- Test error (also called generalized error) measures how well our training on τ has generalized to data that we have not seen before.

$$\mathrm{Err}_{\mathcal{T}} = \mathrm{E}[L(Y, \hat{f}(X))|\mathcal{T}]$$

τ is fixed -> the method is fixed

- Both X and Y are drawn randomly from their joint distribution (population).

- Expected prediction (test/generalized) error is an average test error over all training data (everything is random)

$$\mathrm{Err} = \mathrm{E}[L(Y, \hat{f}(X))] = \mathrm{E}[\mathrm{Err}_{\mathcal{T}}].$$

- These errors are used to assess the performance of a model, and we can do …
  - Model selection: estimating the performance of different models in order to choose the best one.
  - Model assessment: having chosen a final model, estimating its prediction error (generalization error) on new data.

Training error

- Training error is based on data points that have been used for training

$$\overline{\text{err}} = \frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{f}(x_i)).$$

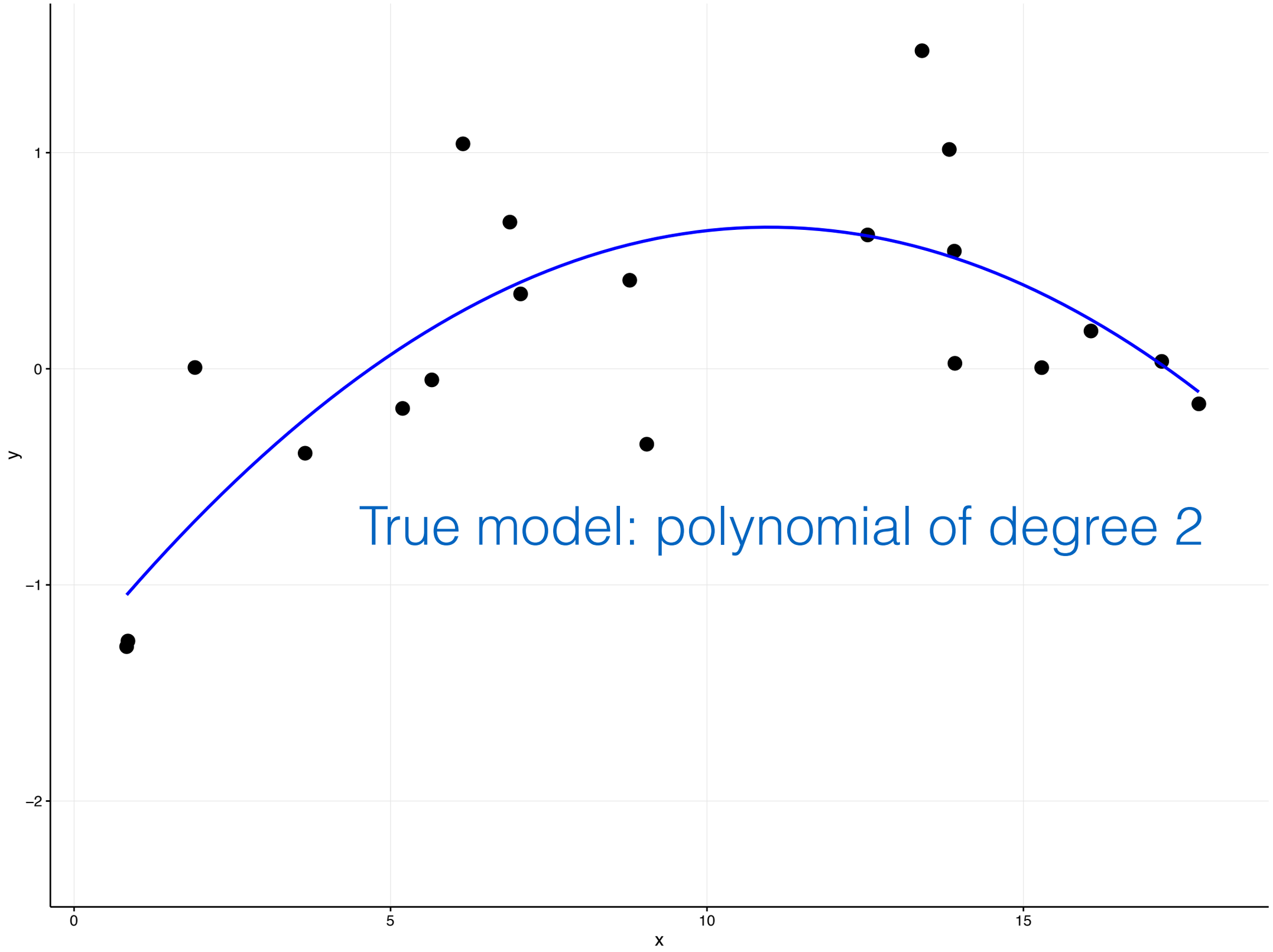
- Regression: $L(Y, \hat{f}(X)) = \begin{cases} (Y - \hat{f}(X))^2 & \text{squared error} \\ |Y - \hat{f}(X)| & \text{absolute error.} \end{cases}$

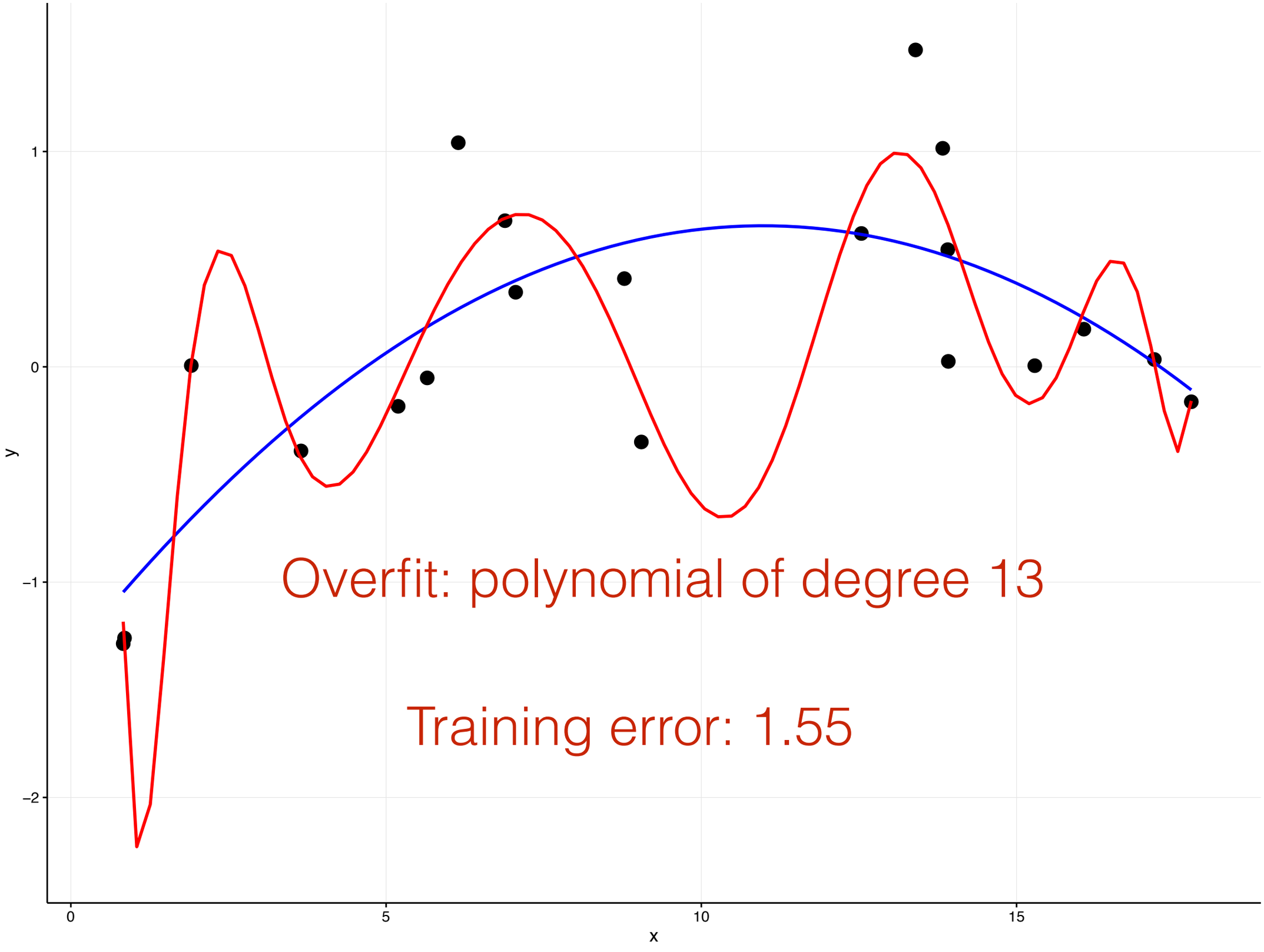
- Classification:

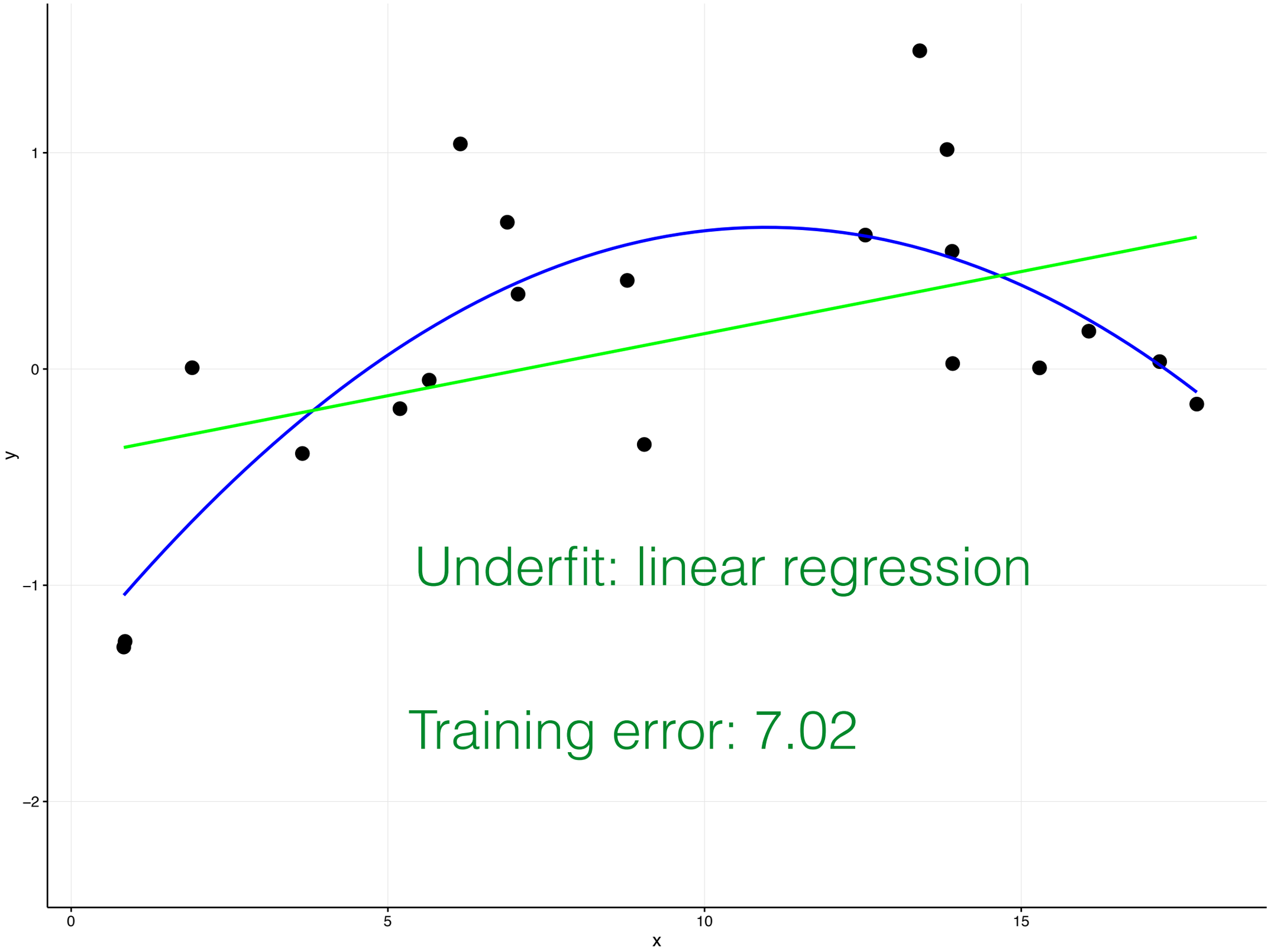
$$\begin{aligned} L(G, \hat{G}(X)) &= I(G \neq \hat{G}(X)) \quad \text{(0–1 loss)}, \\ L(G, \hat{p}(X)) &= -2\sum_{k=1}^{K} I(G = k) \log \hat{p}_k(X) \\ &= -2 \log \hat{p}_G(X) \quad (-2 \times \text{log-likelihood}). \end{aligned}$$

Test vs training errors

- Is the training error an good estimate for the test error?
  - Before the final exam, a teacher hands out some practice problems and solutions to the class. They can be considered as a training set.
  - If the teacher gives out the exam problems from the training data, students may perform very well in the exam.
  - However, students know the exam problems ahead of time, so students' performance on them cannot measure how well students have learned course materials.


1
0
−1
−2
y
0
5
10
15
x
True model: polynomial of degree 2

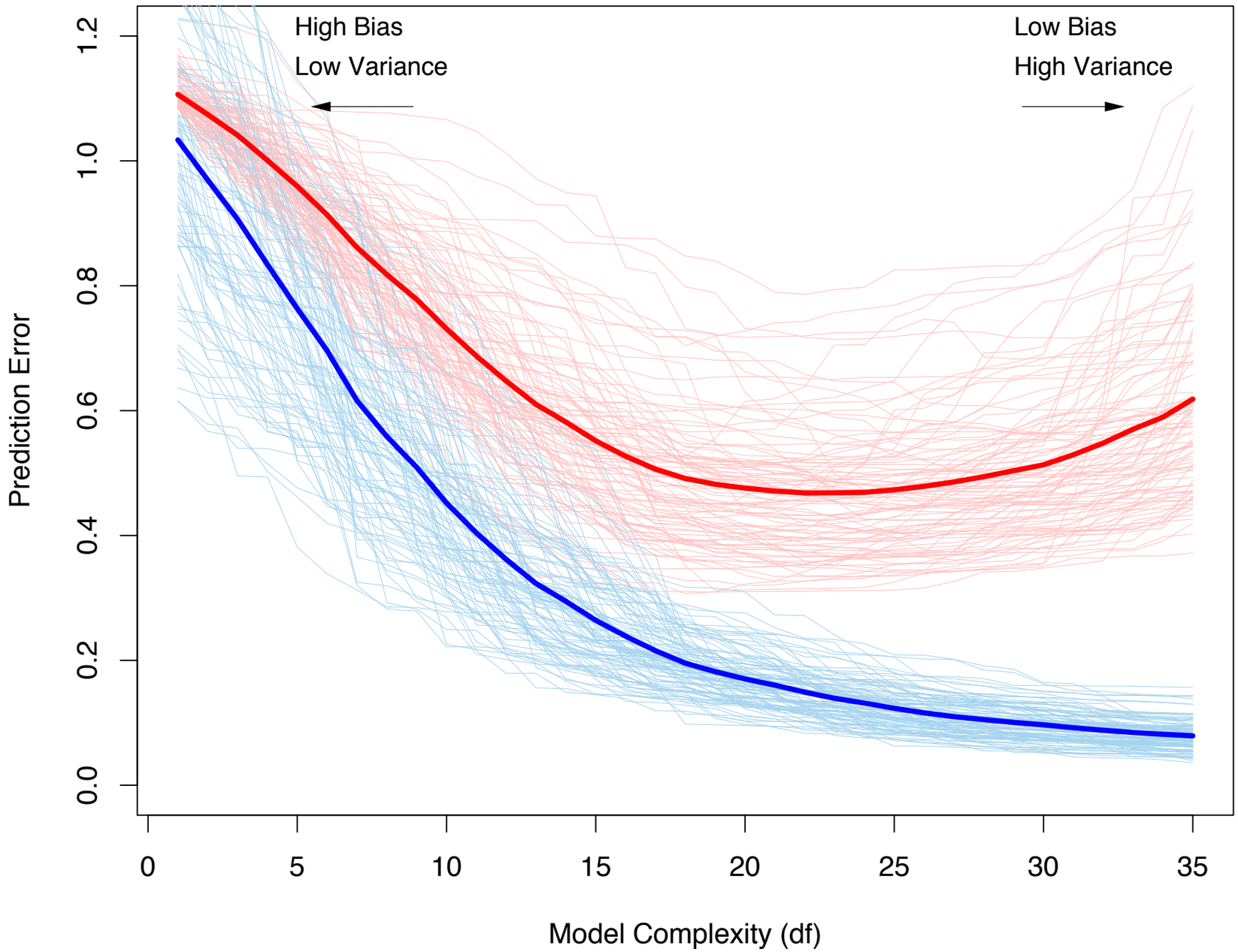

1
0
−1
−2
y
0
5
10
15
x
Overfit: polynomial of degree 13
Training error: 1.55



Underfit: linear regression

Training error: 7.02

Expected training error vs expected test error



Optimism of training error

- The training error is an overly optimistic estimate of the test error.

- In the test error, the test input vectors does not coincide with the training input vectors x_i's.

- In-sample error: Y^0 is new responses at fixed training points x_i -> easier to see the optimism.

$$\mathrm{Err}_{\mathrm{in}} = \frac{1}{N}\sum_{i=1}^{N} \mathrm{E}_{Y^0}[L(Y_i^0, \hat{f}(x_i))|\mathcal{T}]$$

- If one uses the training error as an estimate of test error, it is usually biased downward.

$$\text{op} \equiv \text{Err}_{\text{in}} - \overline{\text{err}}.$$

$$\text{Err}_{\text{in}} = \frac{1}{N}\sum_{i=1}^{N} \text{E}_{Y^0}[L(Y_i^0, \hat{f}(x_i))|\mathcal{T}]$$

$$\overline{\text{err}} = \frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{f}(x_i)).$$

- The average optimism ω is the expectation of the optimism over training set outcome variable y (training input vectors x_i's are fixed).

$$\omega \equiv \text{E}_{\mathbf{y}}(\text{op}). \quad \omega = \frac{2}{N}\sum_{i=1}^{N} \text{Cov}(\hat{y}_i, y_i), \quad \text{E}_{\mathbf{y}}(\text{Err}_{\text{in}}) = \text{E}_{\mathbf{y}}(\overline{\text{err}}) + \frac{2}{N}\sum_{i=1}^{N} \text{Cov}(\hat{y}_i, y_i).$$

- Fitting the data harder will increase the optimism, and therefore it will increase the discrepancy b/w training and test errors.

In-sample error for model selection

- For the model selection, one needs to estimate the performance of different models.

- Usually, in-sample error is not of interest since future values of the input vectors are not likely to coincide with their training input vectors.

- For the model selection, however, the relative (rather than absolute) size of the error is what matters, and the in-sample error often leads to effective model selection.

Estimate of in-sample error

- General form of the in-sample estimate:

$$\widehat{\mathrm{Err}}_{\mathrm{in}} = \overline{\mathrm{err}} + \hat{\omega}$$

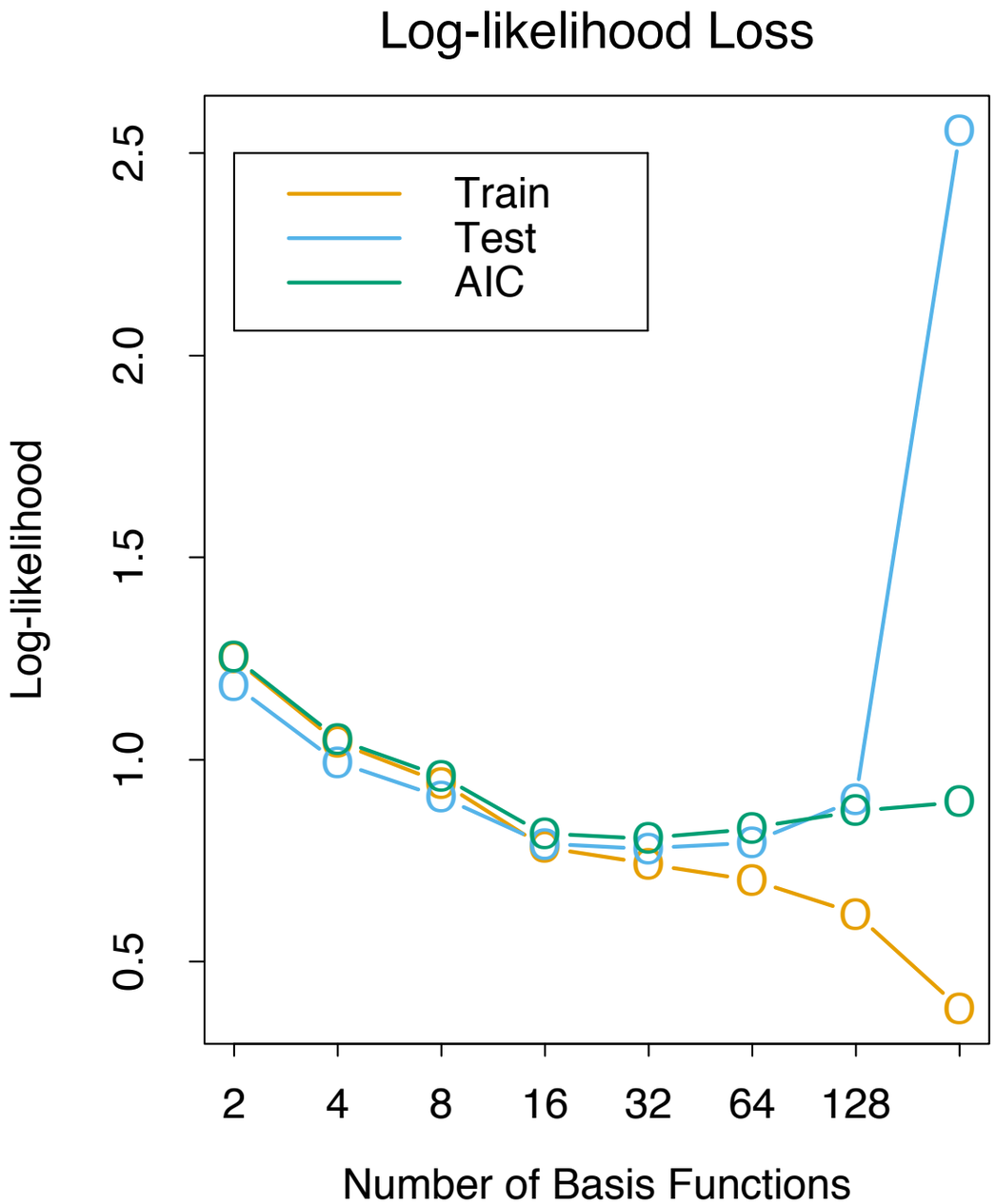
- Mallow Cp (d = # of parameters):

$$C_p = \overline{\mathrm{err}} + 2 \cdot \frac{d}{N} \hat{\sigma_\varepsilon}^2.$$

- Akaike information criterion (AIC):

$$\mathrm{AIC} = -2 \cdot \mathrm{loglik} + 2 \cdot d$$

-



Bayesian information criterion (BIC)

- BIC is motivated by the Bayesian approach to model selection.

$$\mathrm{BIC} = -2 \cdot \mathrm{loglik} + (\log N) \cdot d$$

- Consider M different models ($\mathcal{M}_m$, m=1,…,M). If we assume that the prior over models is uniform (i.e., all M models are equally likely to be true), then

$$\Pr(\mathbf{M}_m \mid \text{training data}) \approx \frac{1}{C} \exp[-\tfrac{1}{2}\mathrm{BIC}_m]$$

- Lower BIC implies higher posterior probability of the model.
-

$$\frac{e^{-\frac{1}{2}\cdot \mathrm{BIC}_m}}{\sum_{\ell=1}^{M} e^{-\frac{1}{2}\cdot \mathrm{BIC}_\ell}}$$

AIC vs BIC

	M1	**M2**	**M3**
AIC	301	302	303

- AIC selects M1. However, it does not tell how much more M1 is likely to be true than other models.

	M1	**M2**	**M3**
BIC	301	302	303
P(M|data)	0.5	0.31	0.19

- BIC selects M1, and M1 is 1.66 times more likely to be true than M2.

AIC vs BIC

$$\text{AIC} = -2 \cdot \text{loglik} + 2 \cdot d \qquad \text{BIC} = -2 \cdot \text{loglik} + (\log N) \cdot d$$

- BIC is asymptotically consistent as a selection criterion. That means, given a family of models including the true model, the probability that BIC will select the correct one approaches one as the sample size becomes large.

- AIC does not have the above property. Instead, it tends to choose more complex models for very large samples.

- AIC is asymptotically efficient as a selection criterion. The probability that AIC will select the model with the smallest prediction error approaches one as the sample size becomes large. BIC does not have the above property.

- For small or moderate samples, BIC often chooses models that are too simple, because of its heavy penalty on complexity.

- Both AIC and BIC are popular, and they can be used to compare non-nested models.

In a data-rich situation…

- One can randomly divide a dataset into three parts
  - 1) Training set (50%): used to fit the models
  - 2) Validation set (25%): used to estimate prediction error for model selection.
  - 3) Test set (25%): used to assess the generalized error for the final chosen model.
- Unfortunately, data is not enough in the most of cases.

Cross-validation

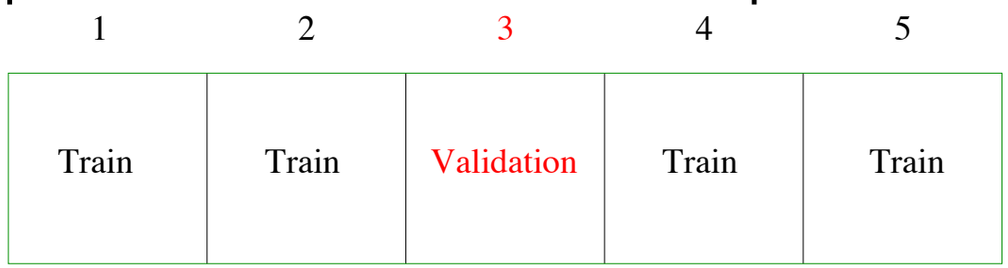
- The simplest and most widely used method for estimating prediction error is cross-validation (CV).

- This method directly estimates the expected prediction error (average generalized error when the method is applied to an independent test sample),

$$\mathrm{Err} = \mathrm{E}[L(Y, \hat{f}(X))] = \mathrm{E}[\mathrm{Err}_{\mathcal{T}}].$$

K-fold cross-validation

- 1. Randomly split the data into K equal-sized parts.



- 2. Leave k*th* part out. Use the other K-1 parts as a training set to fit a model. Then, use the k*th* part as the test set (calculate the prediction error).

- 3. Repeat (2) for each k = 1,2,…,K

average prediction error across all K trials

$$\mathrm{CV}(\hat{f}) = \frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{f}^{-\kappa(i)}(x_i)).$$

- $\hat{f}^{-k(i)}$ is the model fitted without k-th part that includes i-th observation

Cross-validation for model selection

- Let α (tuning parameter) be a value that determines the model complexity.
  - eg) degree of polynomial regression, $X(X'X+\alpha I)^{-1}X'y$, etc.

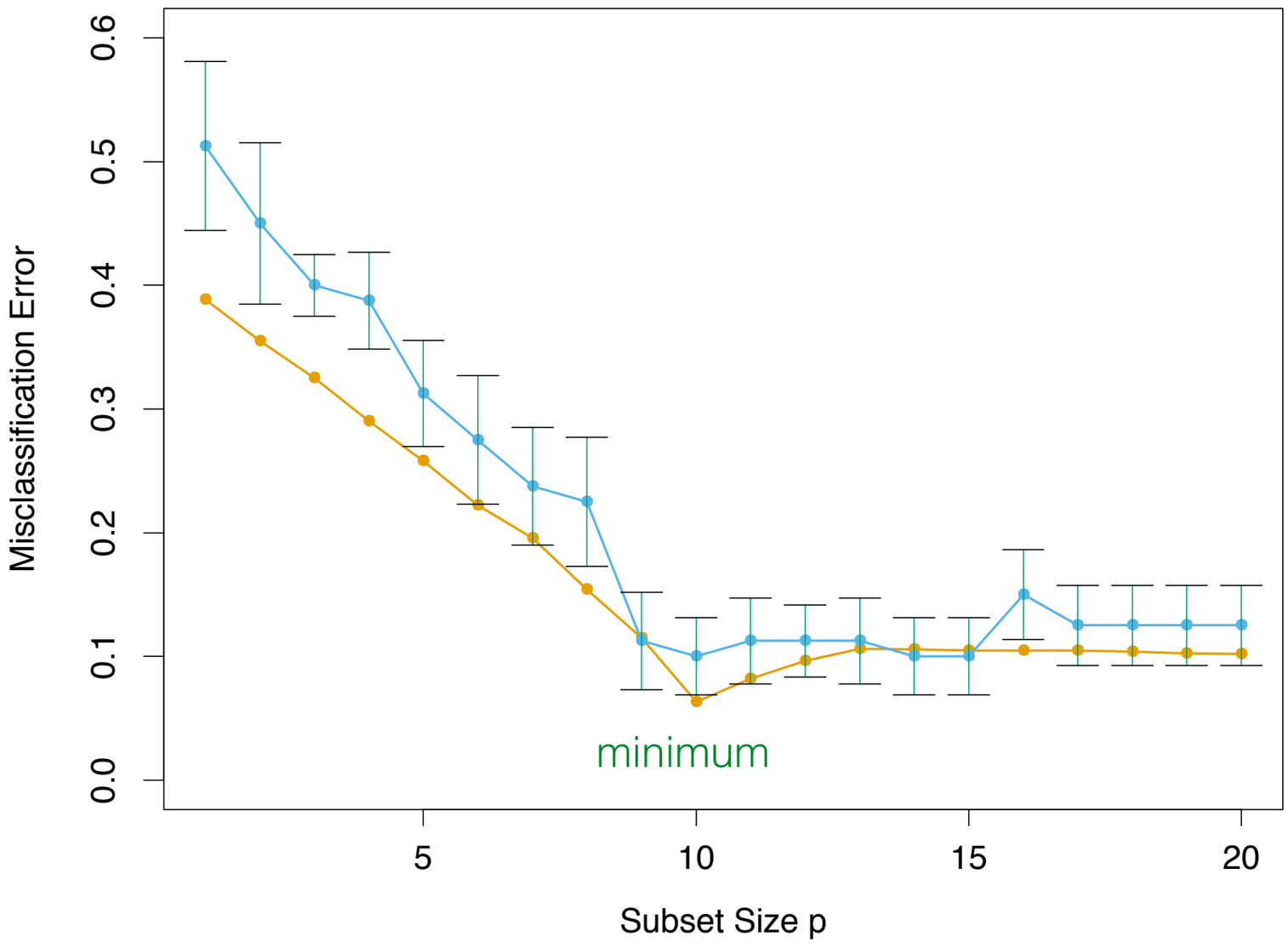
$$\mathrm{CV}(\hat{f}, \alpha) = \frac{1}{N} \sum_{i=1}^{N} L(y_i, \hat{f}^{-\kappa(i)}(x_i, \alpha)).$$

- The above estimates the test error at each α (cross validation error curve), so we can find α such that minimizes the error.

“When there are two competing explanations for an event, the simpler one is more likely.”

–William of Ockham

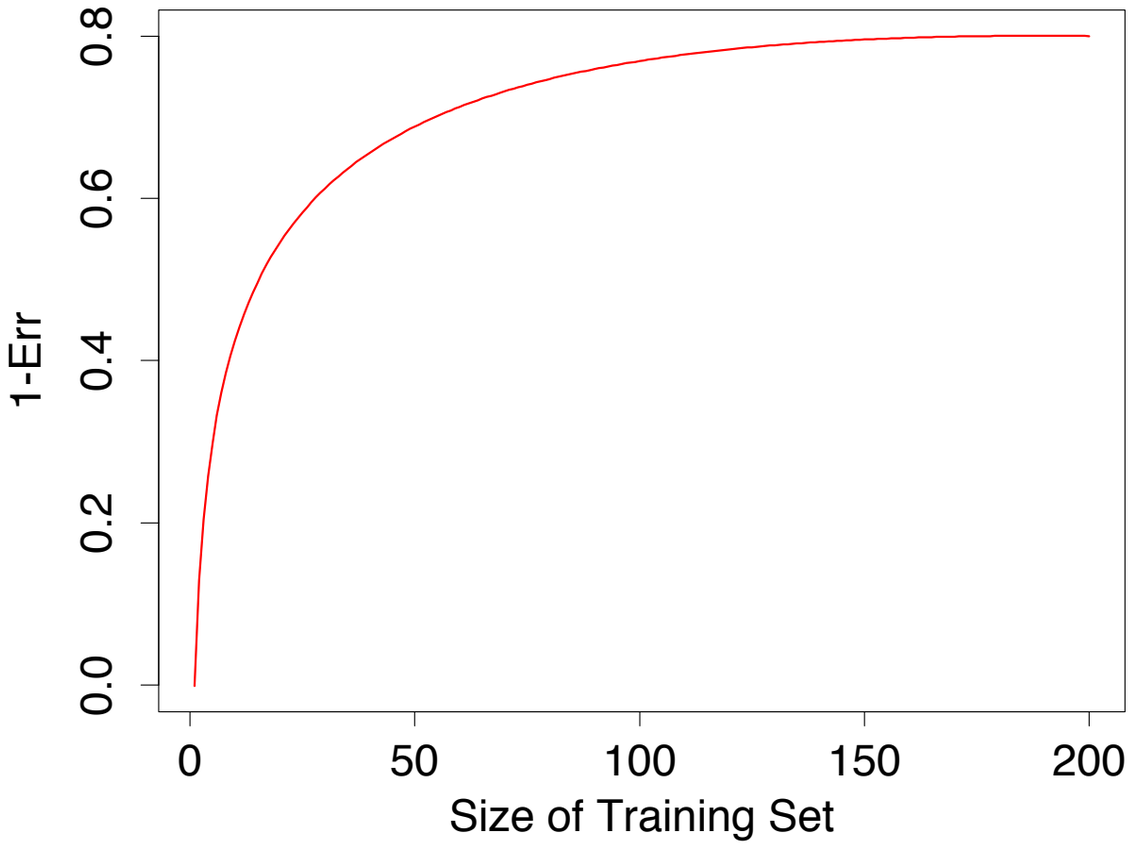
Prediction error and tenfold CV error curve



One standard error rule: we choose the most parsimonious model whose error is no more than one standard error above the error of the best model (the least complex model that performs as well as the best one).

Which K should we use?

- K=5 or 10 CV is recommended.
- K=N: leave-one-out CV is approximately unbiased for the prediction error, but it can have high variance (training sets are so similar).
- K=2: high bias (biased upwards because we are only training on half the data each time), but low variance (training sets are independent).
- K=5 or 10 often gives lower variance and lower bias -> bias-variance tradeoff.



Model assessment and selection: part II

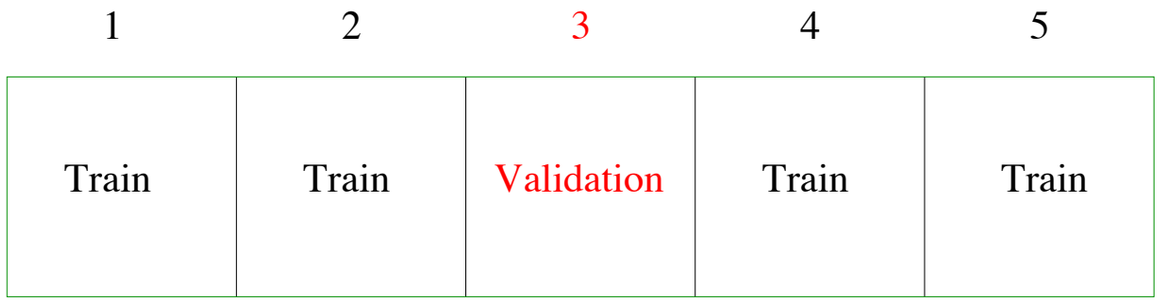
STAT 6312
Department of mathematics, UTA

ESL 7.3, 3.3, 3.3.1-2, 3.3.4, 3.4, 3.4.1-3

Optional reading: ESL 7.3.1, 3.3.3, 3.4.4, 3.6, 3.8.4-5

Review: Cross-validation

- The cross-validation (CV) is widely used for estimating prediction error.



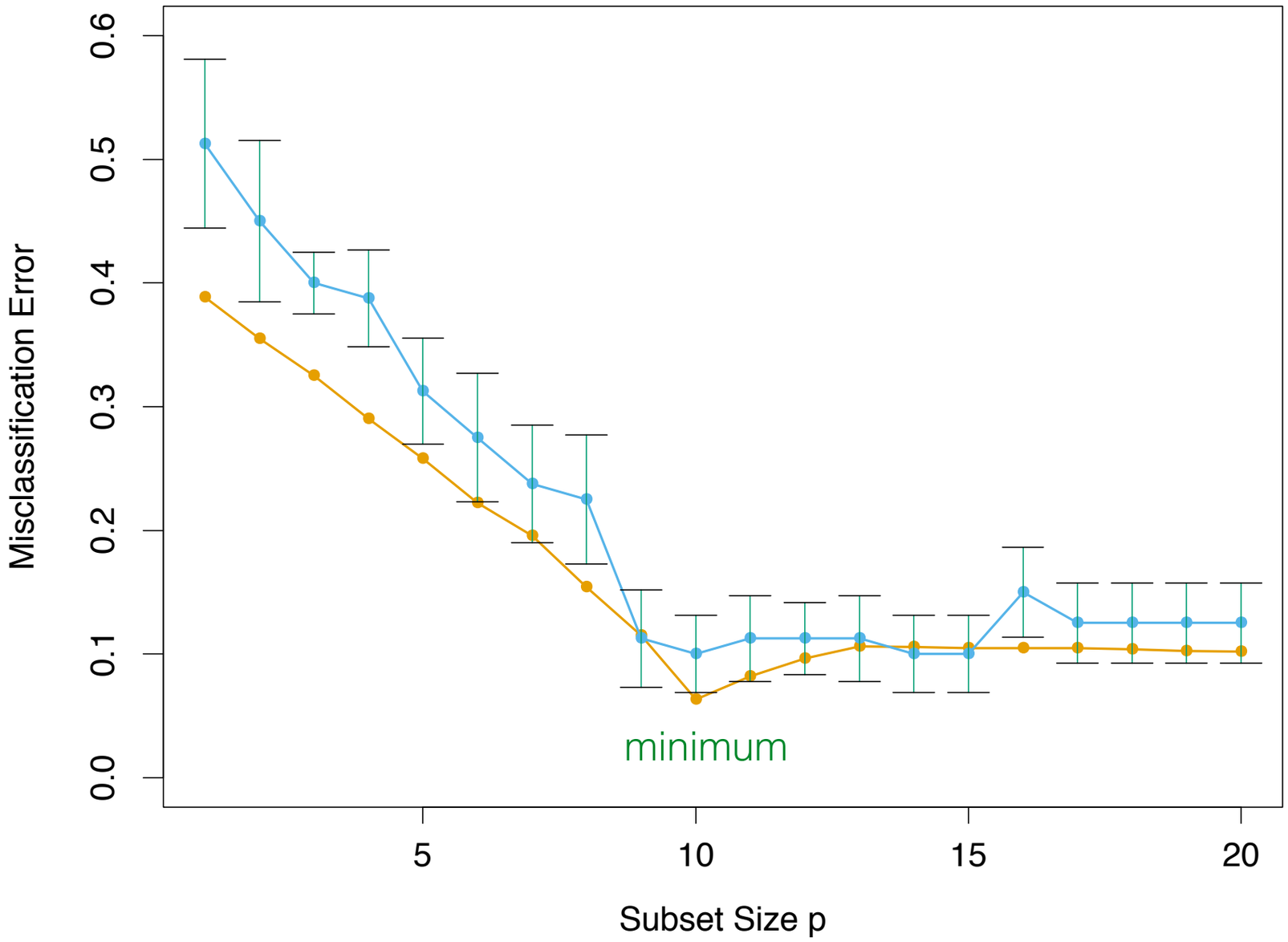
$$\mathrm{CV}(\hat{f}) = \frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{f}^{-\kappa(i)}(x_i)).$$

$\hat{f}^{-k(i)}$ is the model fitted without k-th part that includes i-th observation

- For a tuning parameter α of the model, one can use the CV error curve to find the tuning parameter minimizing the error.

- $$\mathrm{CV}(\hat{f}, \alpha) = \frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{f}^{-\kappa(i)}(x_i, \alpha)).$$

Prediction error and tenfold CV curve



One standard error rule: we choose the least complex model that performs as well as the best on.

Bias-variance decomposition

- Recall: expected prediction error of a regression fit at $X=x_0$:

$$\begin{aligned}
\mathrm{Err}(x_0) &= E[(Y-\hat{f}(x_0))^2|X=x_0] \\
&= \sigma_\varepsilon^2 + [\mathrm{E}\hat{f}(x_0) - f(x_0)]^2 + E[\hat{f}(x_0) - \mathrm{E}\hat{f}(x_0)]^2 \\
&= \sigma_\varepsilon^2 + \mathrm{Bias}^2(\hat{f}(x_0)) + \mathrm{Var}(\hat{f}(x_0)) \\
&= \text{Irreducible Error} + \mathrm{Bias}^2 + \text{Variance}.
\end{aligned}$$

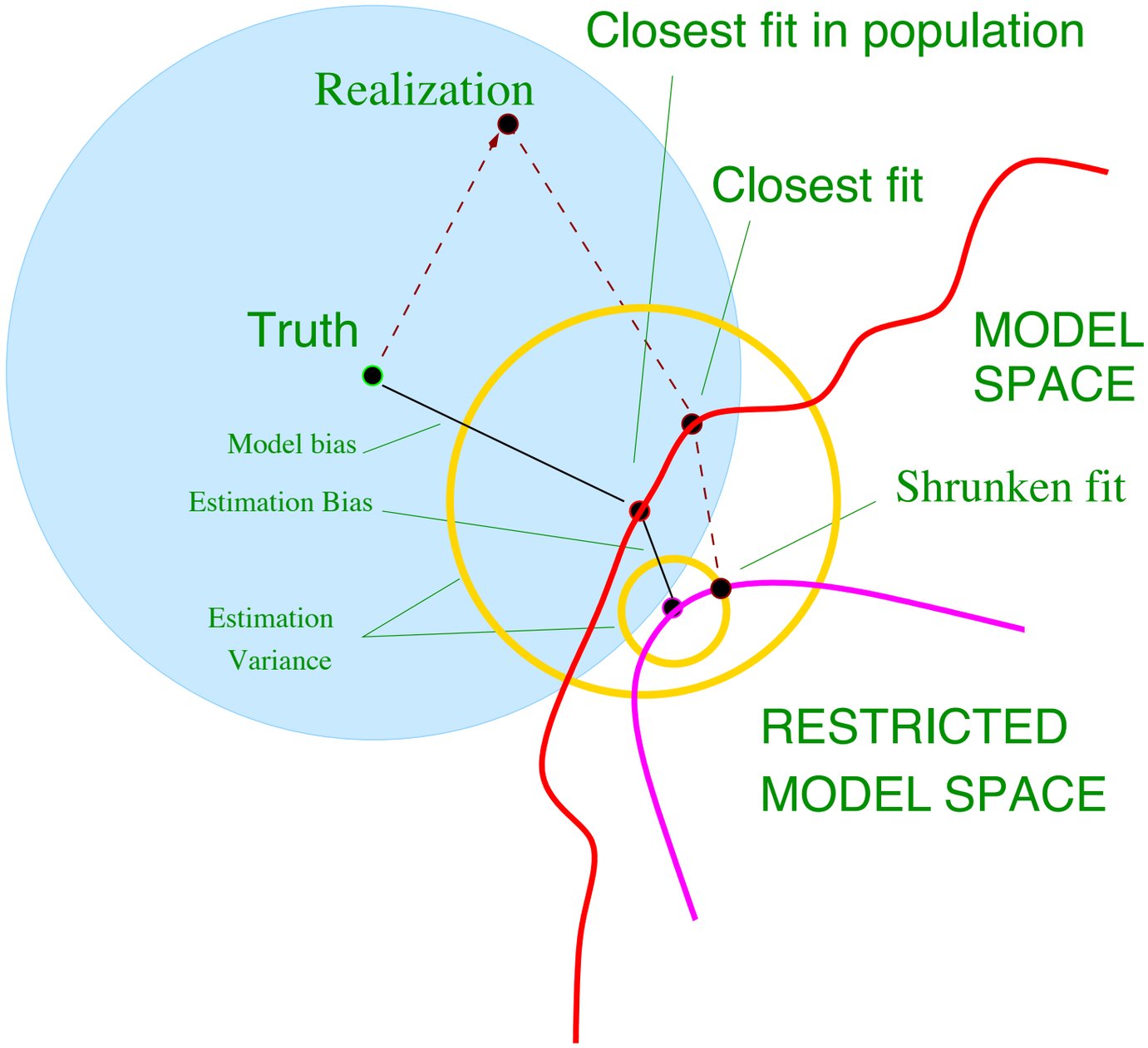
- First term is the variance of $E(Y|X=x_0)$, and it cannot be avoided no matter how well we estimate the regression function.

- Second term is the squared bias: The squared distance of the mean of estimates from from the target.

- Third term is the variance of the estimate: The average squared distance of the estimate from its mean.

- bias-variance tradeoff: typically, the complex model has lower (squared) bias, but higher variance.

- Often, considering the restricted model space (e.g. estimated coefficients are shrunken towards zero) lead us to have lower expected prediction error.



Model selection

- Best subset selection
- Forward stepwise selection
- Backward stepwise selection
- Forward stagewise regression

Best subset selection

1. For k =0,1,…p:

    1. Fit all ${}_pC_k$ models that contain exactly k predictors (M_0 denote the null model, which contains no predictors).

    2. Pick the best among these ${}_pC_k$ models, and call it M_k. The best model is defined as having the largest R^2.

2. Select a single best model from among $M_0,\ldots,M_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2

Stepwise selection

- What is the number of models we consider in the best subset selection? Can apply the best subset selection with very large p?

- An enormous search space can lead to overfitting and high variance of the coefficient estimates.

- For these reasons, stepwise methods, which explore a far more restricted set of models, are attractive alternatives to best subset selection.

- It is not guaranteed to find the best possible model out of all 2^p models containing subsets of the p predictors.

Forward stepwise selection

- Forward stepwise selection begins with the null model, and then adds predictors to the model, one-at-a-time, until all of the predictors are in the model.
- In particular, at each step the variable that gives the greatest additional improvement to the fit is added to the model.

1. M_0 denote the null model.
2. For k=0,…,p−1,
    1. Consider all p − k models that augment the predictors in M_k with one additional predictor.
    2. Choose the best among these p − k models, and call it M_{k+1}. Here best is defined as having highest R^2.
3. Select a single best model from among M_0,…,M_p using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2.

Backward stepwise selection

- Backward stepwise selection begins with the full model containing all p predictors, and then iteratively removes the least useful predictor, one-at-a-time.

- Backward stepwise selection requires that the number of samples n is larger than the number of variables p (so that the full model can be fit).

- In contrast, forward stepwise can be used even when $n < p$, and so is the only viable subset method when p is very large.

- Because of their discrete process (variables are either retained or discarded), these three model selection methods it often exhibits high variance, and so doesn't reduce the prediction error of the full model.

Revisit: shortcoming of linear regression

- Predictive ability
  - The linear regression fit often has low bias but high variance.
- Interpretative ability
  - Linear regression “freely” assigns a coefficient to each predictor variable (i.e. no constraints).
  - Hence we want to encourage our fitting procedure to make only a subset of the coefficients large, and others small or even zero

- Suppose two predictors (x_1, x_2) are highly correlated.
  - min $||\mathbf{y}-\mathbf{X}\beta||_2$: $2x_1 - 1x_2 \approx 3x_1 - 2x_2 \approx 4x_1 - 3x_2$.
  - Coefficients may be “freely” assigned to each predictor variable.
- min $||\mathbf{y}-\mathbf{X}\beta||_2$ subject to $\beta_1^2 + \beta_2^2 \leq t$
  - $t = 5$: $\beta_1=2$, $\beta_2=1$
  - Now, coefficients can be determined well with the size constraint. The smaller t is, the more shrinkage towards zero.

Ridge regression

- Ridge regression shrinks the regression coefficients by imposing a penalty on their size (L_2 norm).

- No penalty for β_0: β_0 is estimated by the mean of **y**, and the design matrix **X** is centered.

- Predictors are not in the same scale: to get fair penalty on each predictor, **X** is standardized (ie. each column has a mean zero and a unit variance).

$$\hat{\beta}^{\text{ridge}} = \text{argmin}_{\beta}(\mathbf{y} - \mathbf{X}\beta)'(\mathbf{y} - \mathbf{X}\beta) \text{ subject to } \sum_{j=1}^{p} \beta_j^2 \leq t$$

$$\Leftrightarrow \hat{\beta}^{\text{ridge}} = \text{argmin}_{\beta}(\mathbf{y} - \mathbf{X}\beta)'(\mathbf{y} - \mathbf{X}\beta) + \lambda \| \beta \|_2^2$$

$$\Leftrightarrow \hat{\beta}^{\text{ridge}} = \mathbf{X}(\mathbf{X}'\mathbf{X} + \lambda \mathbf{I})^{-1}\mathbf{X}'\mathbf{y}$$

$$\hat{\beta}^{\text{ridge}} = \text{argmin}_{\beta}(\mathbf{y} - \mathbf{X}\beta)'(\mathbf{y} - \mathbf{X}\beta) \text{ subject to } \sum_{j=1}^{p} \beta_j^2 \leq t$$

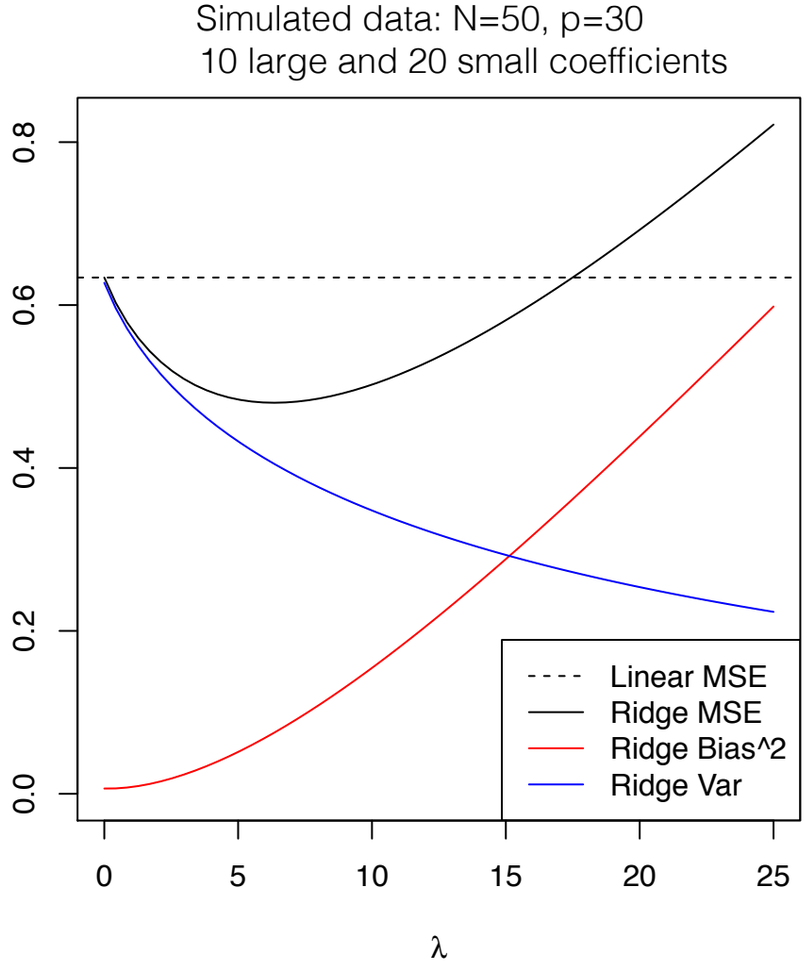
$$\Leftrightarrow \hat{\beta}^{\text{ridge}} = \text{argmin}_{\beta}(\mathbf{y} - \mathbf{X}\beta)'(\mathbf{y} - \mathbf{X}\beta) + \lambda \| \beta \|_2^2$$

$$\Leftrightarrow \hat{\beta}^{\text{ridge}} = \mathbf{X}(\mathbf{X}'\mathbf{X} + \lambda \mathbf{I})^{-1}\mathbf{X}'\mathbf{y}$$

- The larger $\lambda \geq 0$ is, the more penalty on the size of β.
  - For $\lambda > 0$, the ridge estimate of β is shrunk towards zero (ridge estimate is shrinkage estimate).
    - If $\lambda \approx \infty$, all coefficients (except β_0) are zero.
- No penalty on β if $\lambda = 0$. In the case, the ridge estimate = the least square estimate.
- If N < p, $\mathbf{X}'\mathbf{X}$ is singular, but $\mathbf{X}'\mathbf{X}+\lambda\mathbf{I}$ is nonsingular.

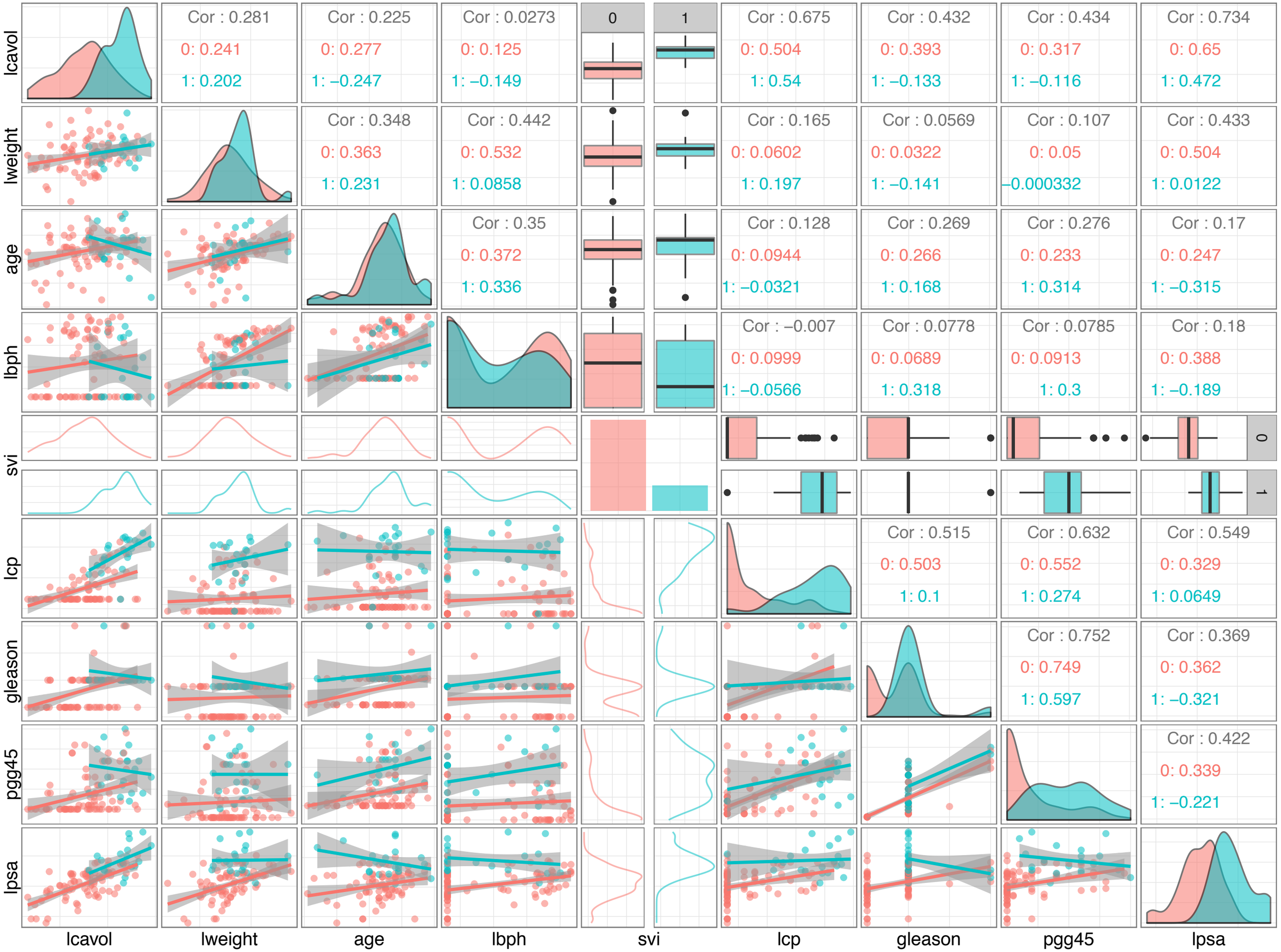
Tuning parameter λ

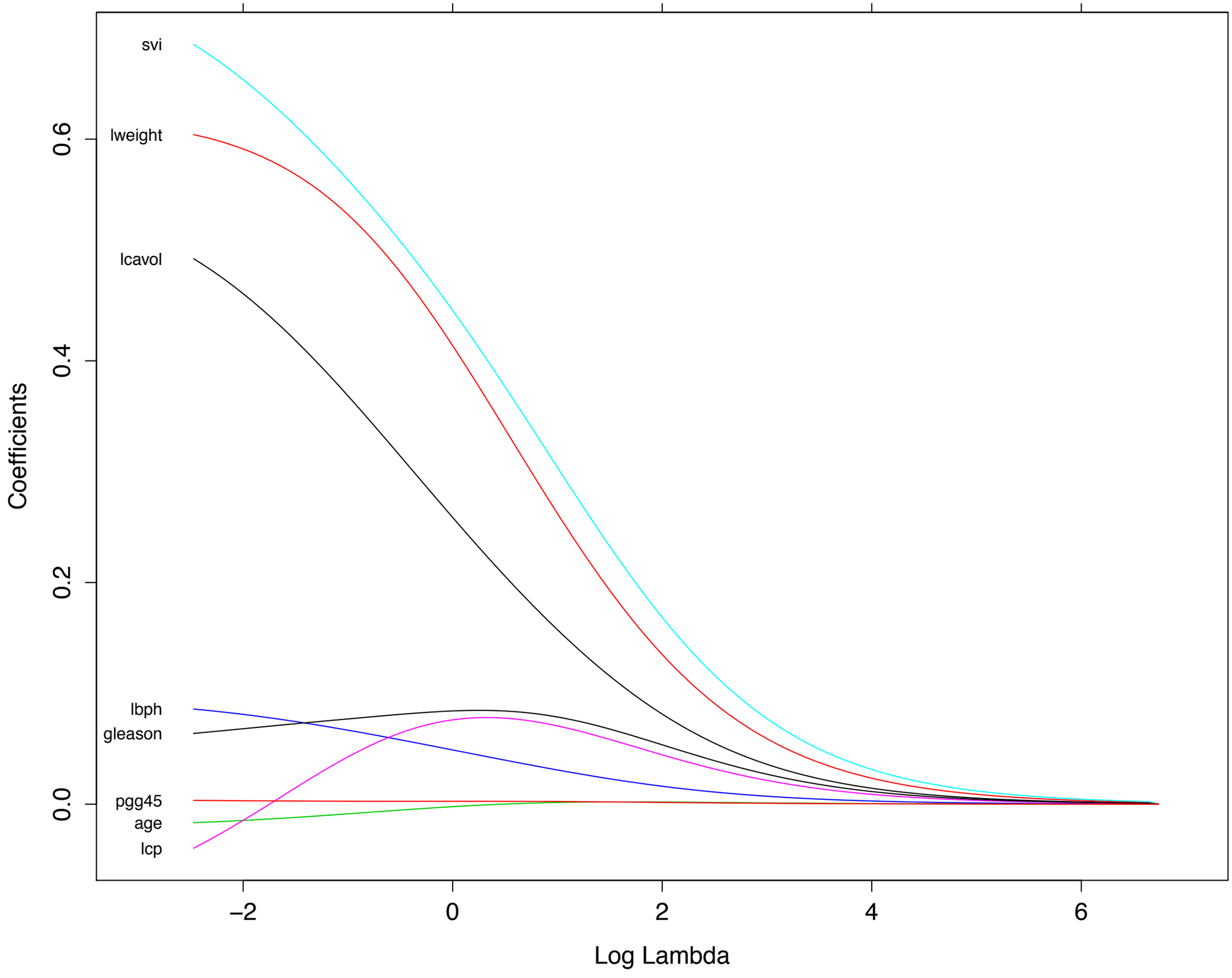
- Bias-variance tradeoff: ridge regression
  - The bias increases as λ (amount of shrinkage) increases. The variance decreases as λ increases.
- λ is a tuning parameter that has to be chosen by users.
- One can use a validation set to choose λ give us the best performance.
- In practice, we cannot afford the validation set, and we use the cross-validation.



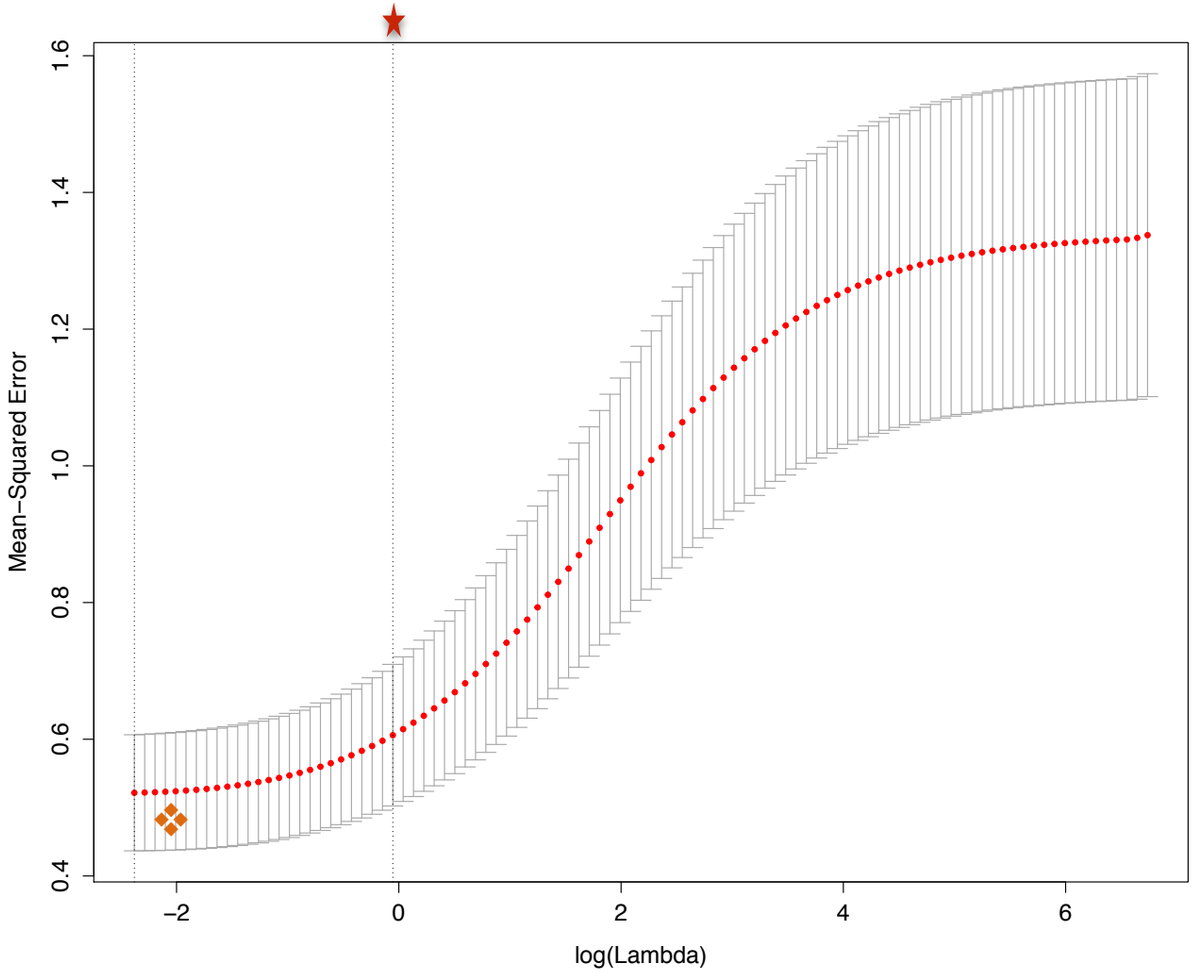
Revisit: Prostate cancer

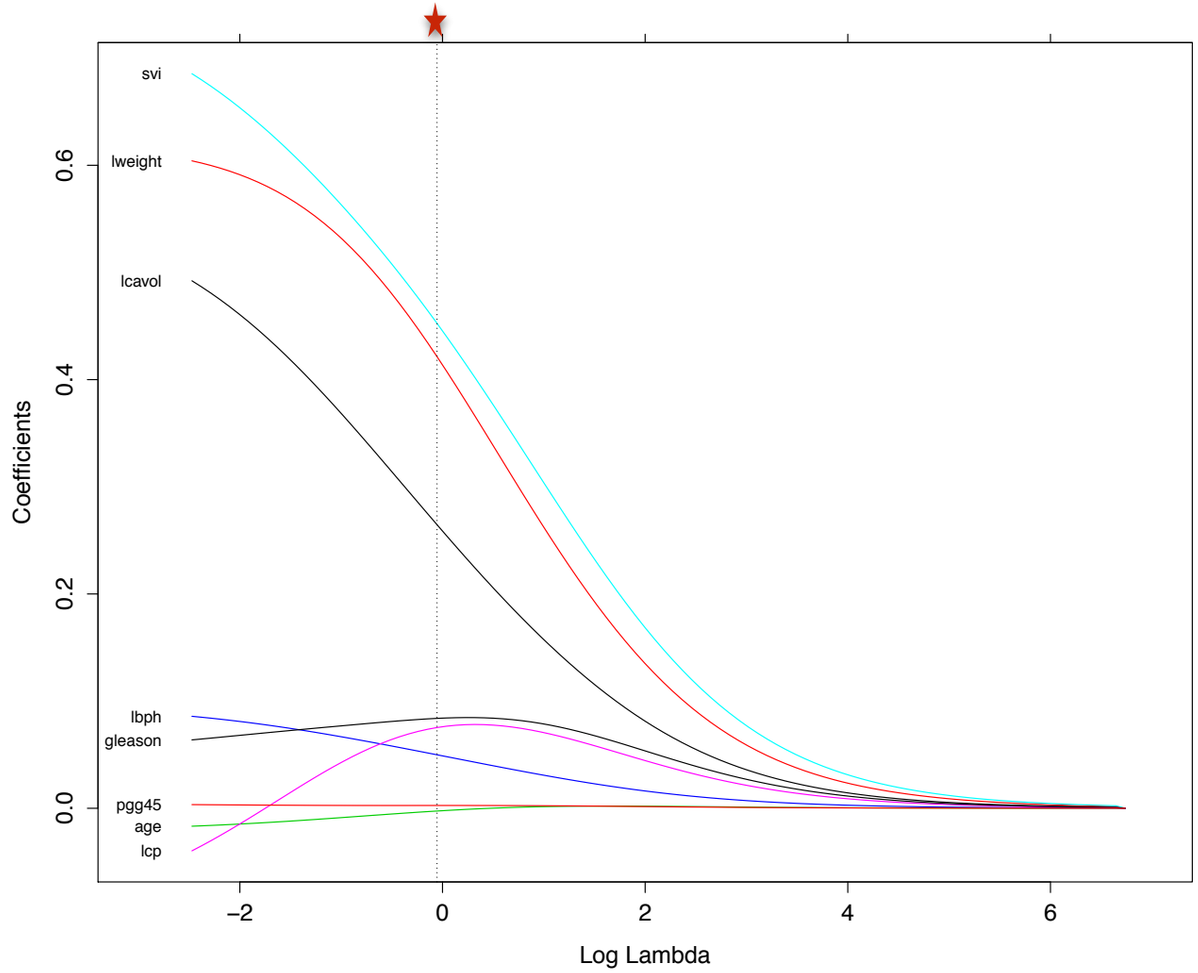
- The goal is to predict the log of PSA (lpsa)
- Predictors
  - log cancer volume (lcavol)
  - log prostate weight (lweight)
  - age
  - log of benign prostatic hyperplasia amount (lbph)
  - seminal vesicle invasion (svi)
  - log of capsular penetration (lcp)
  - Gleason score (gleason)
  - percent of Gleason scores 4 or 5 (pgg45).


lcavol
lweight
age
lbph
svi
lcp
gleason
pgg45
lpsa
Cor : 0.281
0: 0.241
1: 0.202
Cor : 0.225
0: 0.277
1: −0.247
Cor : 0.0273
0: 0.125
1: −0.149
0
1
Cor : 0.675
0: 0.504
1: 0.54
Cor : 0.432
0: 0.393
1: −0.133
Cor : 0.434
0: 0.317
1: −0.116
Cor : 0.734
0: 0.65
1: 0.472
Cor : 0.348
0: 0.363
1: 0.231
Cor : 0.442
0: 0.532
1: 0.0858
Cor : 0.165
0: 0.0602
1: 0.197
Cor : 0.0569
0: 0.0322
1: −0.141
Cor : 0.107
0: 0.05
−0.000332
Cor : 0.433
0: 0.504
1: 0.0122
Cor : 0.35
0: 0.372
1: 0.336
Cor : 0.128
0: 0.0944
1: −0.0321
Cor : 0.269
0: 0.266
1: 0.168
Cor : 0.276
0: 0.233
1: 0.314
Cor : 0.17
0: 0.247
1: −0.315
Cor : −0.007
0: 0.0999
1: −0.0566
Cor : 0.0778
0: 0.0689
1: 0.318
Cor : 0.0785
0: 0.0913
1: 0.3
Cor : 0.18
0: 0.388
1: −0.189
Cor : 0.515
0: 0.503
1: 0.1
Cor : 0.632
0: 0.552
1: 0.274
Cor : 0.549
0: 0.329
1: 0.0649
Cor : 0.752
0: 0.749
1: 0.597
Cor : 0.369
0: 0.362
1: −0.321
Cor : 0.422
0: 0.339
1: −0.221


svi
lweight
lcavol
lbph
gleason
pgg45
age
lcp
0.0
0.2
0.4
0.6
Coefficients
−2
0
2
4
6
Log Lambda

- 10-fold cross-validation is used for each λ to compute the estimate prediction error.
- The left vertical line is the value at which the minimal mean squared error is achieved
- The right vertical line (starred) is for the most regularized model whose mean squared error is within one standard error of the minimal (one standard error rule).



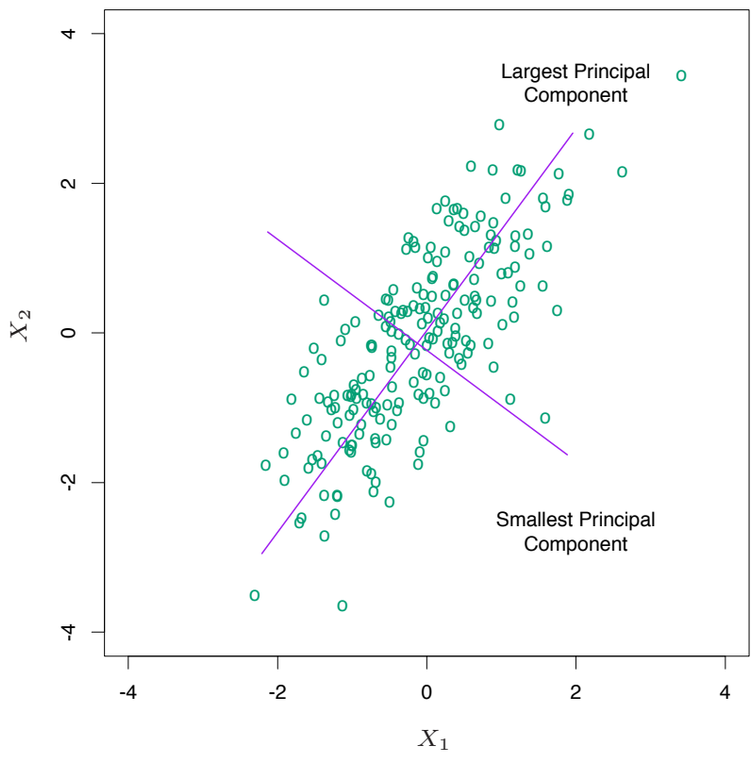


SVD and ridge regression

- The SVD of the feature matrix **X** has the form **X**=**UDV**′.
- $\mathbf{u}_j$ is j*th* column of **U** (normalized principal component), and d_j^2/N is variance of j*th* principal component variable.
- Ridge regression shrinks low-variance directions more.

$$\begin{aligned}\mathbf{X}\hat{\beta}^{\text{ls}} &= \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \\ &= \mathbf{U}\mathbf{U}^T\mathbf{y},\end{aligned}$$

$$\begin{aligned}\mathbf{X}\hat{\beta}^{\text{ridge}} &= \mathbf{X}(\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y} \\ &= \mathbf{U}\,\mathbf{D}(\mathbf{D}^2+\lambda\mathbf{I})^{-1}\mathbf{D}\,\mathbf{U}^T\mathbf{y} \\ &= \sum_{j=1}^{p}\mathbf{u}_j\frac{d_j^2}{d_j^2+\lambda}\mathbf{u}_j^T\mathbf{y},\end{aligned}$$



Summary: ridge regression

- The ridge regression shrinks regression coefficients towards zero. It increases bias, but reduces variance, so it can outperform the ordinary least square (OLS) regression.

- The turning parameter can be determined by the cross-validation.

- The ridge includes all p predictors in the final model (shrinkage estimates are not exactly zero), so it is not convenient for the variable selection.

- The ridge regression can be applied for a short-and-fat (N<p) feature matrix, but the OLS regression cannot.

Lasso

- The ridge regression uses the L_2 penalty on the size of coefficients.

- In the lasso (least absolute shrinkage and selection operator), the L_2 penalty is replaced by L_1 penalty.

$$\hat{\beta}^{\text{lasso}} = \text{argmin}_{\beta}(\mathbf{y} - \mathbf{X}\beta)'(\mathbf{y} - \mathbf{X}\beta) \text{ subject to } \sum_{j=1}^{p} |\beta_j| \leq t$$

$$\Leftrightarrow \hat{\beta}^{\text{lasso}} = \text{argmin}_{\beta}(\mathbf{y} - \mathbf{X}\beta)'(\mathbf{y} - \mathbf{X}\beta) + \lambda \| \beta \|_1$$

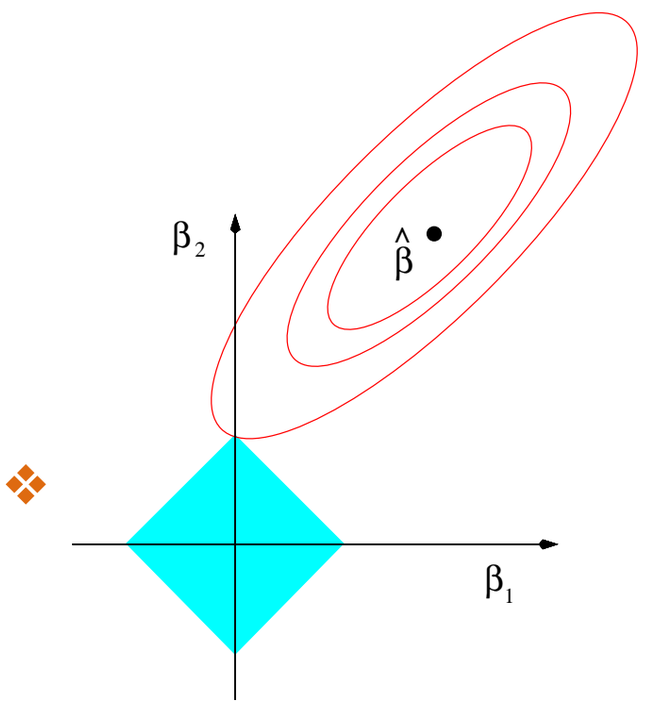
- As with ridge regression, the lasso shrinks the coefficient estimates towards zero.

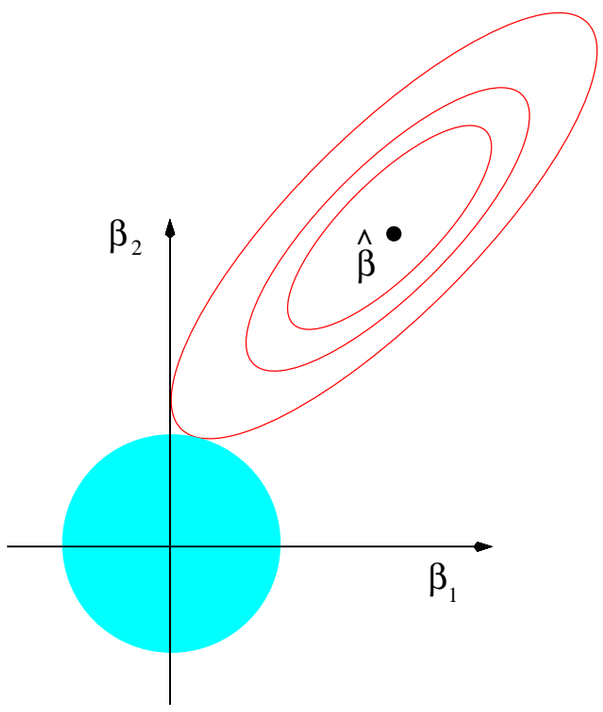
- However, in the case of the lasso, the L_1 penalty has the effect of forcing some of the coefficient estimates to be exactly zero when the tuning parameter λ is sufficiently large.

- Lasso has a big advantage with respect to interpretation. It performs variable selection in the linear model. (one can say, the lasso yields sparse models).

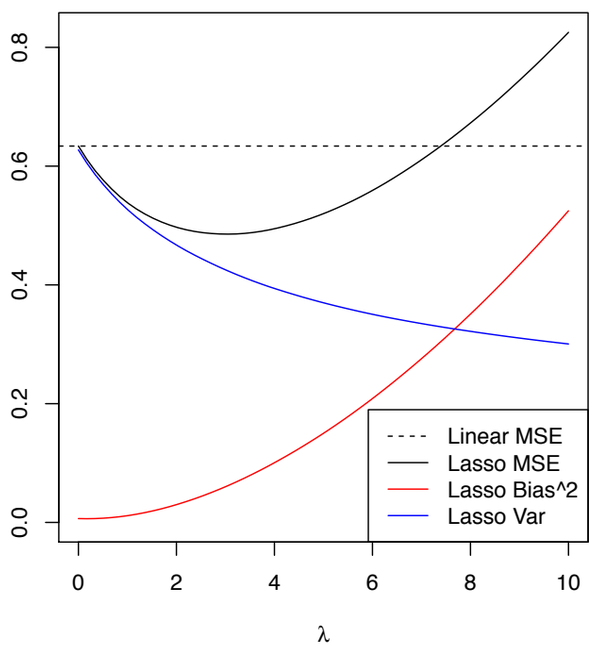
- Bias-variance tradeoff on the choice of the tuning parameter λ.

- As in ridge regression, selecting a good value of λ for the lasso is critical; cross-validation is again the method of choice.





Simulated data: N=50, p=30
10 large and 20 small coefficients

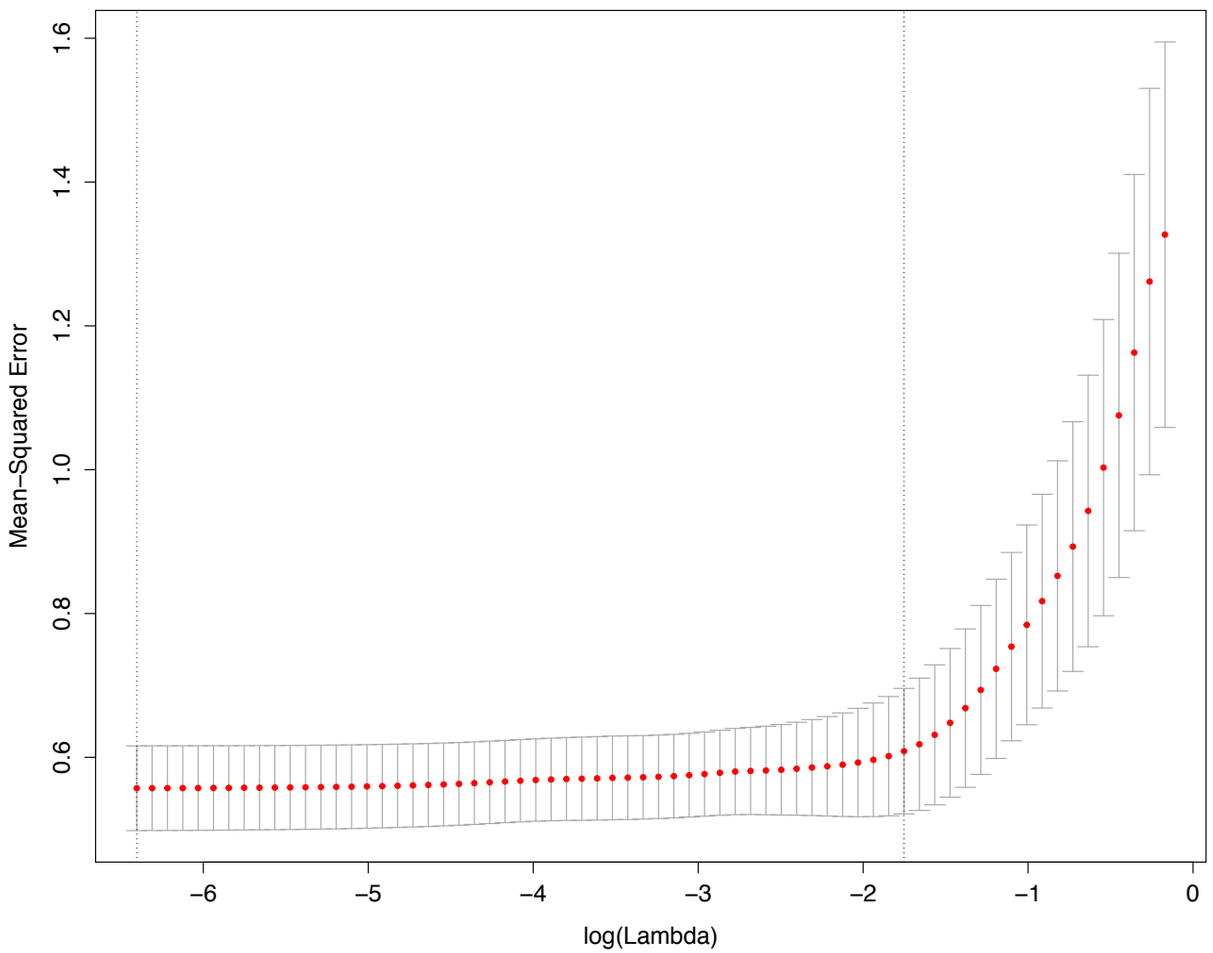


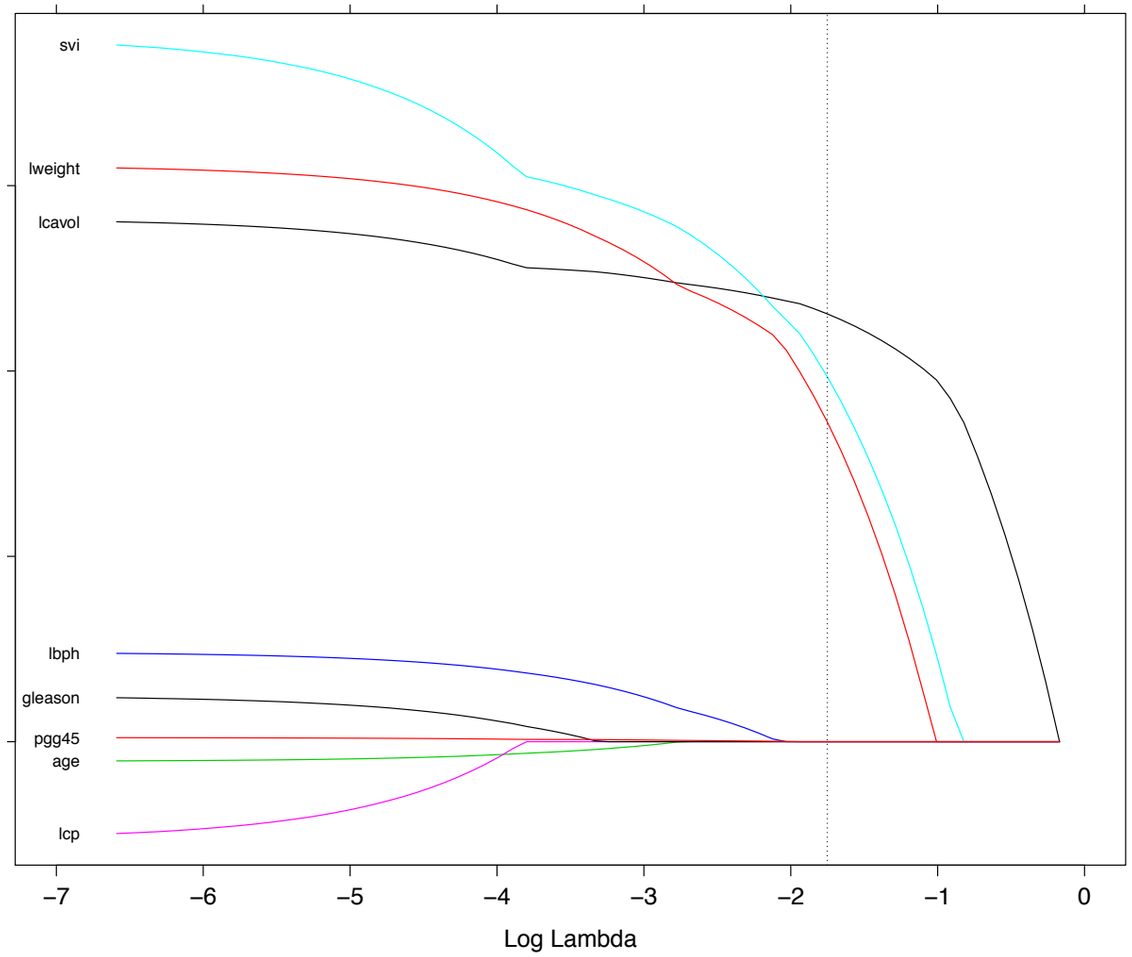
Ridge vs lasso

- Neither the ridge regression nor the lasso will universally dominate the other in terms of the prediction error.

- In general, one might expect the lasso to perform better when the response is a function of only a relatively small number of predictors.

- Example: Prostate cancer data
- Generalized gradient decent algorithm can be used to find the lasso solution.
- We observe that coefficients are shrunk to exactly zero as λ gets larger.
- 10-fold CV is used. One standard error rule to choose λ.





Elastic net = lasso + ridge

- The lasso will occasionally achieve poor results when there is a high degree of collinearity in the features and ridge regression will perform better.

- Also, if there is a group of highly correlated variables, then the lasso tends to select one variable from a group and ignore the others.

- L1 norm is underdetermined when N<p, while ridge regression can handle this.

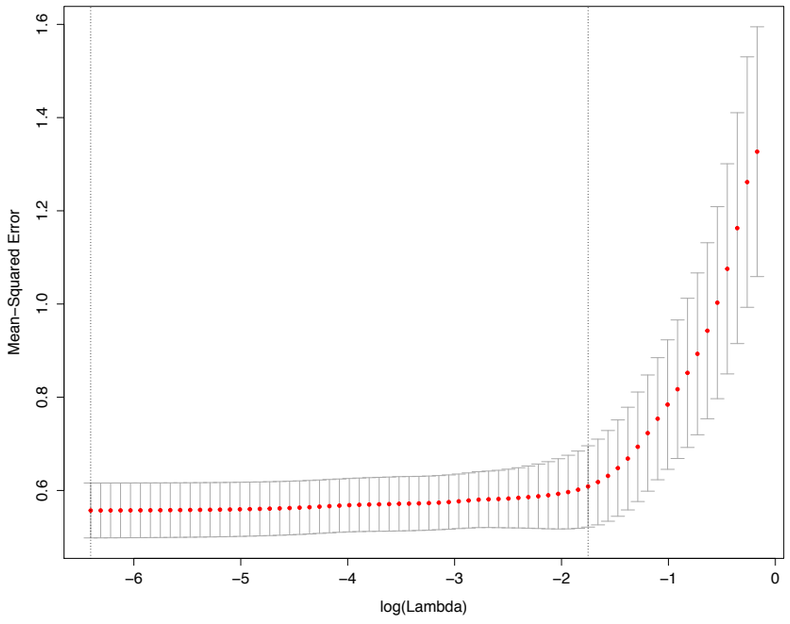
- Elastic net regression includes lasso (L1) and ridge (L2) penalties.

$$\hat{\beta}^{\text{e-net}} = \text{argmin}_{\beta}(\mathbf{y} - \mathbf{X}\beta)'(\mathbf{y} - \mathbf{X}\beta) + \lambda[(1-\alpha)\,\|\beta\|_1 + \alpha\,\|\beta\|_2^2] \text{ where } \lambda \geq 0, 0 \leq \alpha \leq 1$$

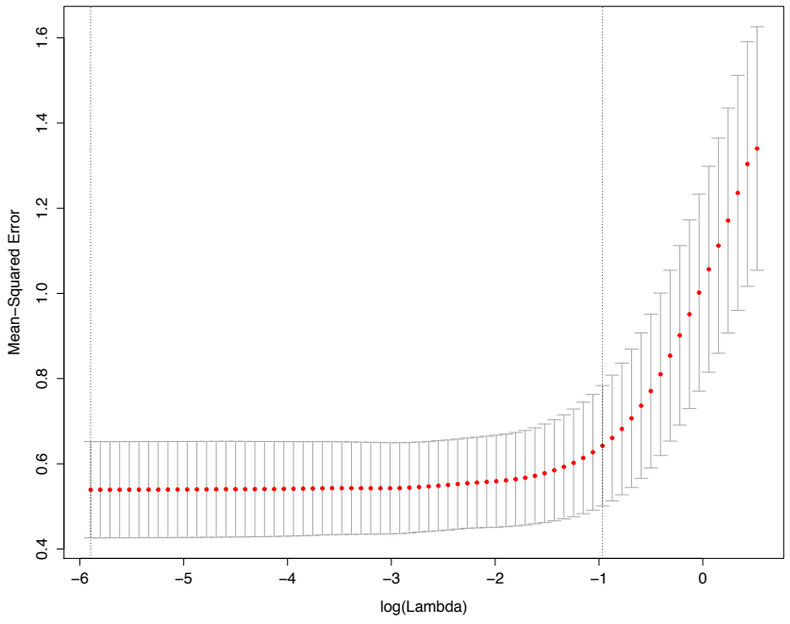
- Now, we have two tuning parameters. What happen if α=0 or α=1?

- Typically, we compute CV error curves over a grid of α, say (0,0.05,0.1,…,1).
- Choose α which include the smallest CV error. Then, we find λ with the smallest CV error (one standard error rule).
- It is recommended to control the folds used in the CV.

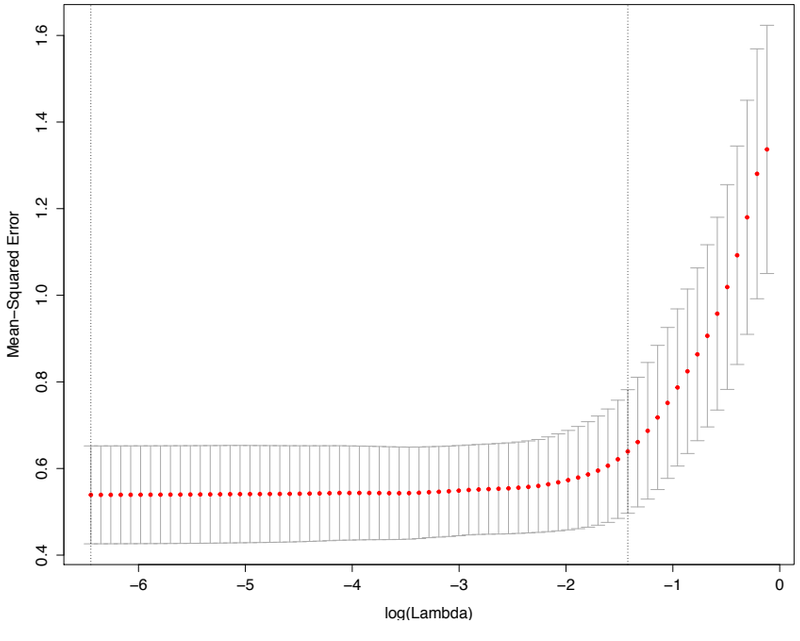
α=0.05



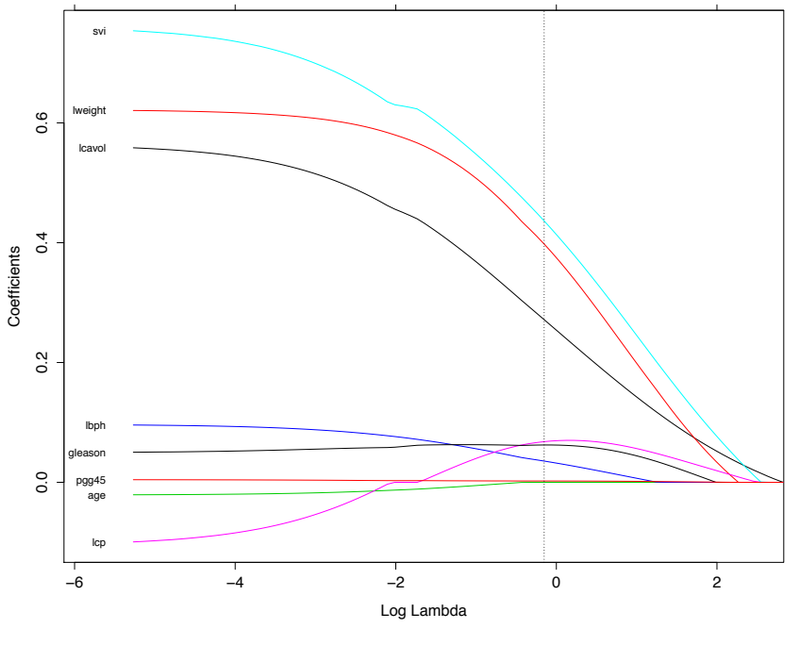
α=0.5



α=0.95



α=0.05



Effective degree of freedom

- The linear regression with p variables usually has p degree of freedom (df), which is equal to tr(**H**).
- For the ridge regression, we can calculate the df in the similar fashion.
  - Hat matrix: $\mathbf{H}_\lambda = \mathbf{X}(\mathbf{X}'\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}'$. Recall that

$$\begin{aligned}
\mathbf{X}\hat{\beta}^{\text{ridge}} &= \mathbf{X}(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y} \\
&= \mathbf{U}\,\mathbf{D}(\mathbf{D}^2 + \lambda\mathbf{I})^{-1}\mathbf{D}\,\mathbf{U}^T\mathbf{y} \\
&= \sum_{j=1}^{p} \mathbf{u}_j \frac{d_j^2}{d_j^2 + \lambda}\mathbf{u}_j^T\mathbf{y},
\end{aligned}
\qquad
\begin{aligned}
\text{df}(\lambda) &= \text{tr}[\mathbf{X}(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T], \\
&= \text{tr}(\mathbf{H}_\lambda) \\
&= \sum_{j=1}^{p} \frac{d_j^2}{d_j^2 + \lambda}.
\end{aligned}$$

  - df(λ) = p when λ = 0 (no regularization) and df(λ) → 0 as λ → ∞.
- The general definition of the effective degree of freedom
- $$\text{df}(\hat{\mathbf{y}}) = \frac{1}{\sigma^2}\sum_{i=1}^{N}\text{Cov}(\hat{y}_i, y_i).$$

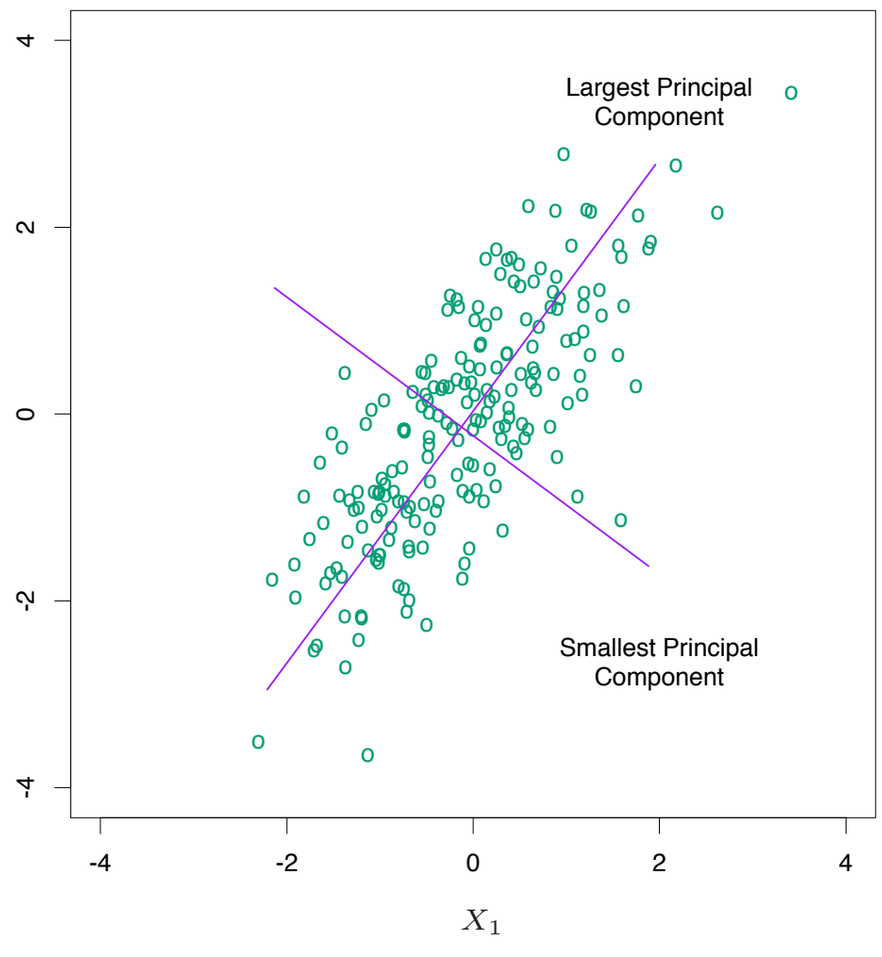
Summary

- The model selection methods are introduced, which are an essential tool for data analysis, especially for big datasets involving many predictors.

- These methods are can be applied to the logistic regression (generalized linear models).
  - min -log L(β) + penalty on β
  - max log L(β) - penalty on β

Some dimension reduction methods

- Principal component regression
- Partial least squares
- SVD on **X**=**UDV**′.
- $\mathbf{v}_1$ is first column of **V** (principal component direction), $\mathbf{u}_1$ is first column of **U** (normalized principal component), and $\mathbf{z}_1=\mathbf{X}\mathbf{v}_1$ is first principal component.



Principal components regression (PCR)

- Recall the SVD on $\mathbf{X}=\mathbf{UDV}'$.
- $\mathbf{v}_m$ is m*th* column of **V** (principal component direction), $\mathbf{u}_m$ is m*th* column of **U** (normalized principal component), and $\mathbf{z}_m=\mathbf{Xv}_m$ is m*th* principal component.
- Principal component is the linear combination of the variables in **X**, and the 1st principal component has largest variance.
- The second principal component has 2nd largest variance, and it is uncorrelated with the 1st, and so on… -> principal components are uncorrelated.
- The first few principal components can be thought as the low-dimensional approximation of the feature matrix.
- We choose the first few principal components, and fit the OLS regression, which gives us shrinkage estimate with respect to the original feature space.

Partial least squares (PLS)

- PCR identifies linear combinations, or directions, that best represent the predictors.

- These directions are identified in an unsupervised way, since the response is not used to help determine the principal component directions — that is, the response does not supervise the identification of the principal components.

- Consequently, PCR suffers from a potential drawback: there is no guarantee that the directions that best explain the predictors will also be the best directions to use for predicting the response.

- Unlike PCR, PLS identifies latent features ($z_1,\ldots z_M$) in a supervised way – i.e., it makes use of the response y in order to identify new features.

- PLS seeks directions that have high variance and have high correlation with the response.

The m*th* PLS direction $\hat{\varphi}_m$ solves:

$$\max_\alpha \mathrm{Corr}^2(\mathbf{y}, \mathbf{X}\alpha)\mathrm{Var}(\mathbf{X}\alpha)$$
$$\text{subject to } ||\alpha|| = 1,\ \alpha^T \mathbf{S}\hat{\varphi}_\ell = 0,\ \ell = 1,\ldots,m-1.$$

S : sample covariance matrix

$z_l = X\hat{\varphi}_l$ (features are standardized).

Algorithm 3.3 *Partial Least Squares.*

1. Standardize each $\mathbf{x}_j$ to have mean zero and variance one. Set $\hat{\mathbf{y}}^{(0)} = \bar{y}\mathbf{1}$, and $\mathbf{x}_j^{(0)} = \mathbf{x}_j,\ j = 1, \ldots, p$.

2. For $m = 1, 2, \ldots, p$

   (a) $\mathbf{z}_m = \sum_{j=1}^{p} \hat{\varphi}_{mj}\mathbf{x}_j^{(m-1)}$, where $\hat{\varphi}_{mj} = \langle \mathbf{x}_j^{(m-1)}, \mathbf{y} \rangle$.

   (b) $\hat{\theta}_m = \langle \mathbf{z}_m, \mathbf{y} \rangle / \langle \mathbf{z}_m, \mathbf{z}_m \rangle$.

   (c) $\hat{\mathbf{y}}^{(m)} = \hat{\mathbf{y}}^{(m-1)} + \hat{\theta}_m \mathbf{z}_m$.

   (d) Orthogonalize each $\mathbf{x}_j^{(m-1)}$ with respect to $\mathbf{z}_m$: $\mathbf{x}_j^{(m)} = \mathbf{x}_j^{(m-1)} - [\langle \mathbf{z}_m, \mathbf{x}_j^{(m-1)} \rangle / \langle \mathbf{z}_m, \mathbf{z}_m \rangle]\mathbf{z}_m,\ j = 1, 2, \ldots, p$.

3. Output the sequence of fitted vectors $\{\hat{\mathbf{y}}^{(m)}\}_1^p$. Since the $\{\mathbf{z}_\ell\}_1^m$ are linear in the original $\mathbf{x}_j$, so is $\hat{\mathbf{y}}^{(m)} = \mathbf{X}\hat{\beta}^{\text{pls}}(m)$. These linear coefficients can be recovered from the sequence of PLS transformations.

PCR & PLS: CV and number of components