Summer School in Astroinformatics: Cross Validation

Le Bao

Department of Statistics
The Pennsylvania State University

Email: lebao@psu.edu

Prediction Accuracy

The previously introduced criterions have limitations, e.g. F-test, LRT requires nested models, AIC, BIC requires likelihood defined under a full probabilistic model.

Let's introduce a more generic model evaluation criterion today.

- We use the training data,

$$\mathcal{D}^{training} = \{(X_i, Y_i), i = 1, 2, ..., n\},$$

  to regress Y on X, and then predict a new Y-value, Y^{new}, by applying the fitted model to a brand-new X-value, X^{new}, from the test set $\mathcal{D}^{test}$.

- The resulting prediction is compared with the actual response value.

- The predictive ability of the regression model is assessed by its prediction error.

Prediction Accuracy

The previously introduced criterions have limitations, e.g. F-test, LRT requires nested models, AIC, BIC requires likelihood defined under a full probabilistic model.

Let's introduce a more generic model evaluation criterion today.

- We use the training data,

$$\mathcal{D}^{training} = \{(X_i, Y_i), i = 1, 2, ..., n\},$$

  to regress Y on X, and then predict a new Y-value, Y^{new}, by applying the fitted model to a brand-new X-value, X^{new}, from the test set $\mathcal{D}^{test}$.

- The resulting prediction is compared with the actual response value.

- The predictive ability of the regression model is assessed by its prediction error.

Prediction Error

- The prediction error, PE, is defined as the mean squared error in predicting Y^{new} using $\hat{f}(X^{new})$.

$$PE = E[(Y^{new} - \hat{f}(X^{new}))^2] = \sigma^2 + ME,$$

where the expectation is taken over (X^{new}, Y^{new}).

ME is the model error,

$$\begin{aligned} ME &= E[(f(X^{new}) - \hat{f}(X^{new}))^2] \\ &= E[(X^{new}\beta - X^{new}\hat{\beta})^2], \\ &= (\beta - \hat{\beta})'\Sigma_{XX}(\beta - \hat{\beta}), \end{aligned}$$

where Σ_{XX} is the covariance matrix of X.

Apparent Error Rate

- We can estimate PE by

$$\hat{PE}(\mathcal{D}^{training}, \hat{\beta}) = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{f}(X_i))^2 = \frac{RSS}{n},$$

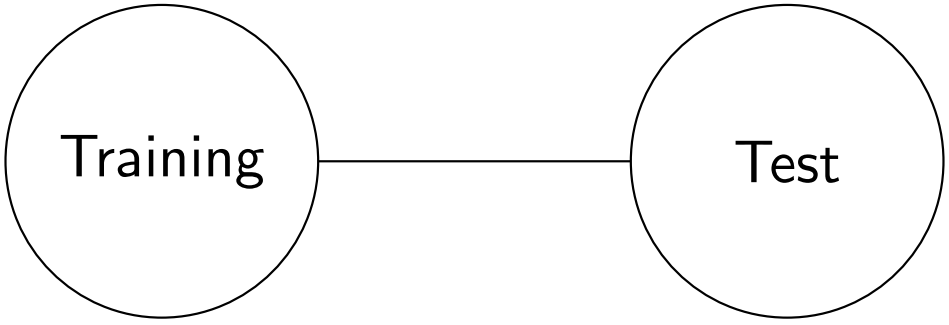
  which we call the apparent error rate for $\mathcal{D}^{training}$.

- The apparent error rate is a misleadingly optimistic value because it estimates the predictive ability of the fitted model from the same data that was used to fit that model.

$$\hat{PE}(\mathcal{D}^{training}, \hat{\beta}) < PE.$$

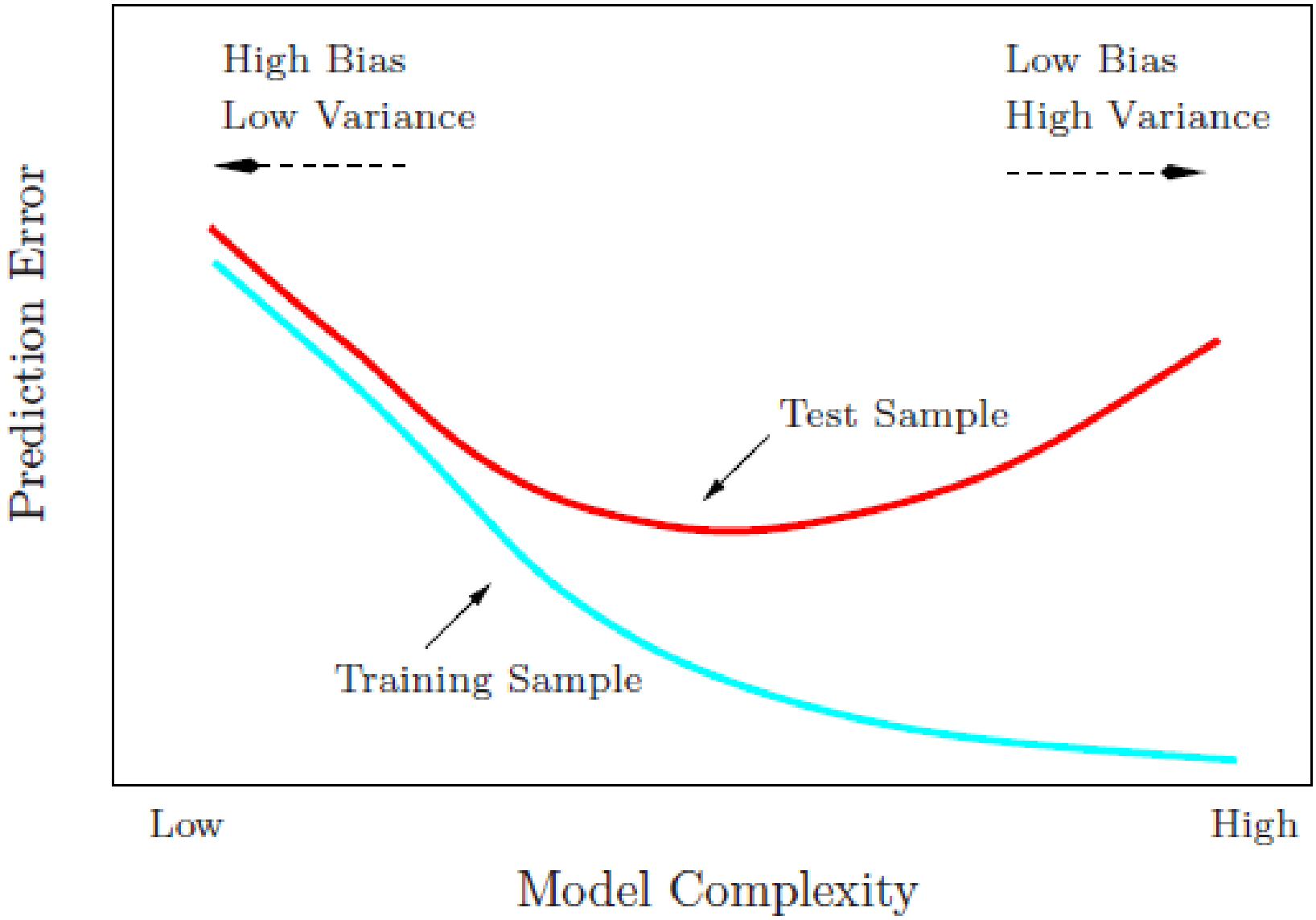
The 2-Way Split

Solution: Let us preserve some data just for estimating PE.



- Training Data (50%): model fit;
- Test Data (50%): estimate PE.

Training and Test Error as A Function of Model Complexity



The Elements of Statistical Learning by FHT

Cross Validation

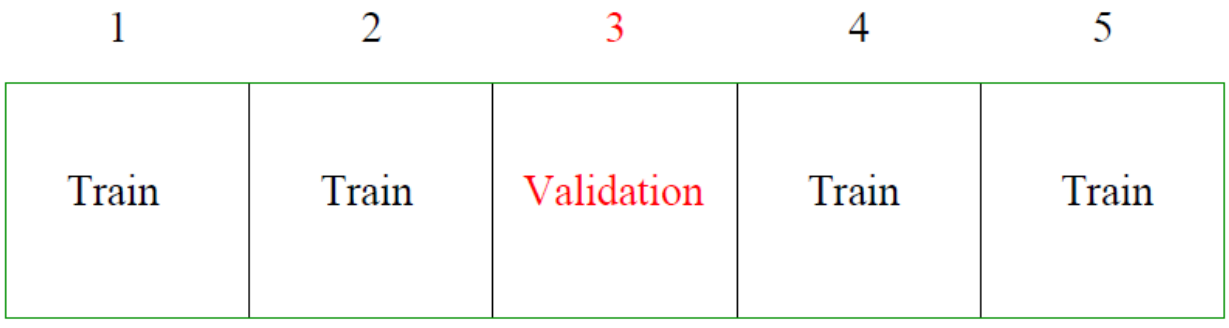
- Among the methods available for estimating prediction error, the most widely used is cross-validation (Stone, 1974).
- We separate the data into training and test datasets, e.g. 50% for each.
- An estimate of PE obtained from the test set is

$$\hat{PE}(\mathcal{D}^{test}, \hat{\beta}) = \frac{2}{n} \sum_{i=1}^{n/2} (Y_i^{test} - \hat{f}(X_i^{test}))^2.$$

  where $\hat{f}$ is learned from the training set.
- The training set and the test set are then switched, and the resulting two estimates of PE are averaged to yield a final estimate.

K-fold Cross Validation

- To generalize the previous procedure, assume that $n = Km$, where $K \geq 2$ is a small integer, such as 5 or 10.
- We split the data set randomly into K disjoint subsets with equal sizes.



- We next create K different versions of the training-test data set, each has a training set consisting of $K - 1$ of the subsets, and a test set of the remaining subset.

K-fold Cross Validation

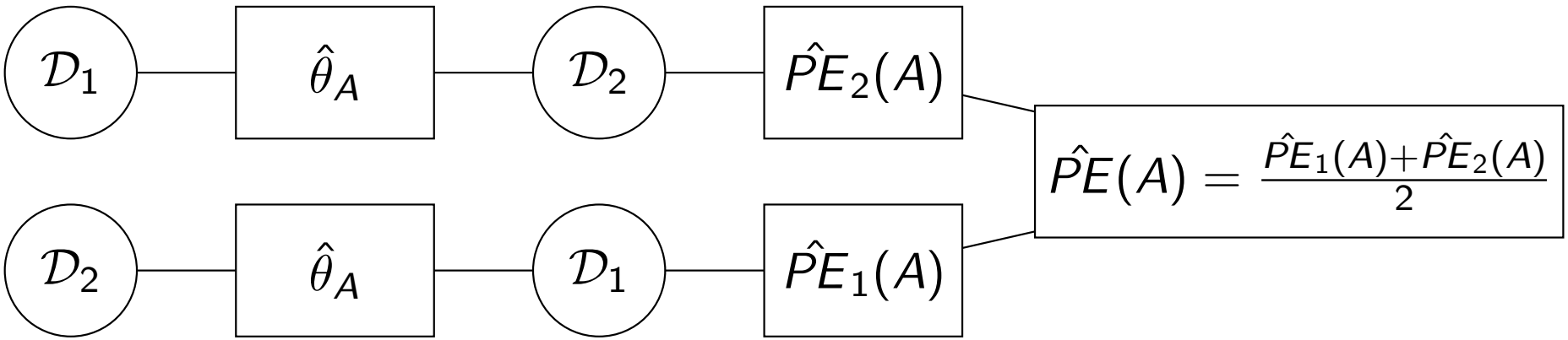
- We compute the prediction error from the kth test set by using the parameters estimated from the training data.
- Repeat the procedure K times, while cycling through each of the test sets, $Data_1^{test}, Data_2^{test}, \ldots, Data_K^{test}$.
- Combining these results gives us a CV/K-estimate of PE,

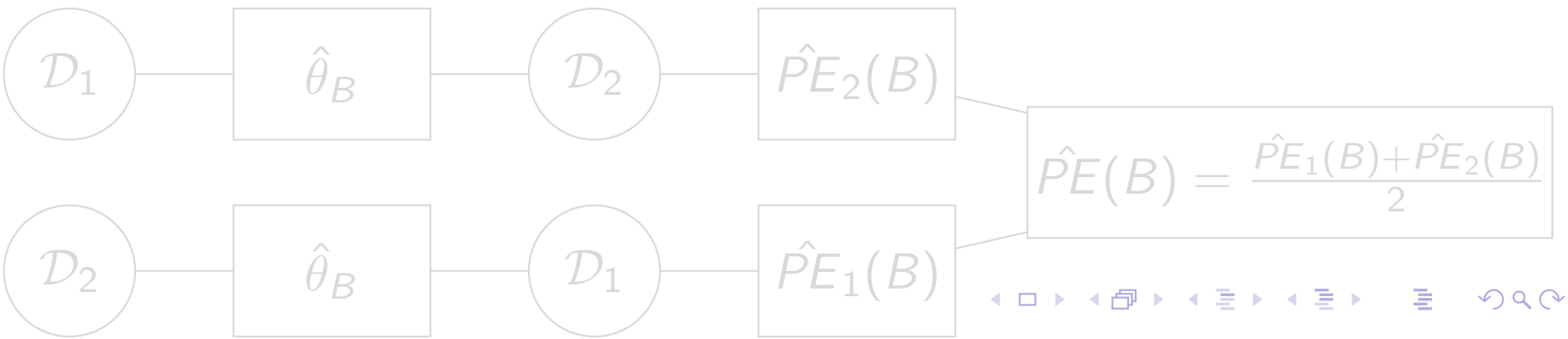
$$\hat{PE}_{CV/K} = \frac{1}{K} \sum_{k=1}^{K} \hat{PE}_k.$$

- This procedure is called K-fold cross-validation.

Cross-Validation

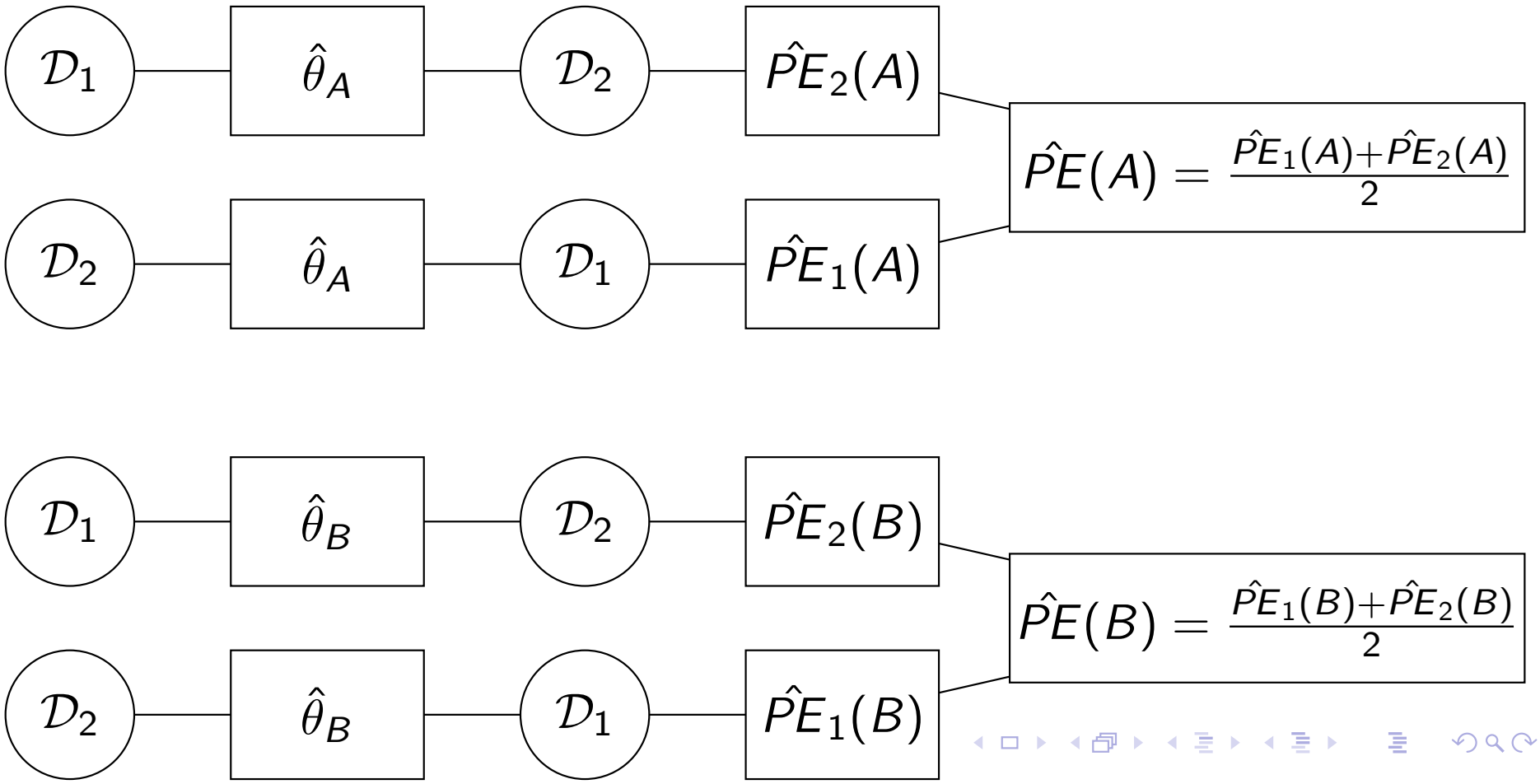
- We always evaluate the same model while switching the training and test data.





Cross-Validation

- We always evaluate the same model while switching the training and test data.



Leave-One-Out Cross Validation

- The special case, $K = n$, is known as leave-one-out cross validation.
- For the ith observation, the fit is computed using all the data except the ith.
- Leave-one-out CV is asymptotically equivalent to AIC.
- It is unbiased for true prediction error, but can be very computationally intensive.
- Overall, five- or tenfold cross validation are recommended as a good compromise. (Hastie et al. 2009, Section 7.10)

R^2 Adjustment

$$
\begin{aligned}
R^2 &= \frac{\text{Variation of Y being explained by X}}{\text{Variation of Y}}, \\
&= \frac{\text{Variation of Y being explained by X}}{\text{Variation explained by X} + \text{Variation NOT explained by X}}.
\end{aligned}
$$

- Variation of Y being explained by X through $X'\beta$,
- Variation of Y NOT being explained by X are ϵ's.

R^2 Adjustment by H. Wang (2016)

$$R^2 = \frac{Var(X'\beta)}{Var(Y)} = 1 - \frac{Var(\epsilon)}{Var(Y)}$$

- In classic R^2, we define $R^2 = 1 - \frac{RSS/n}{SST/n}$, where

$$RSS = \sum_{i=1}^{N}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{N}(y_i - X_i\hat{\beta})^2.$$

- In Wang's R^2 it is a parameter, and an unbiased estimate is

$$\hat{R}^2 = 1 - \frac{RSS^{(\text{test data})}/n}{SST^{(\text{test data})}/n}.$$

3-Way Split

- Discuss whether *PE* evaluated in the validation stage is unbiased for the selected model.

  The *PE* might be biased in the validation stage too, especially if there are too many candidate models.

- Hastie et al. (2009, Section 7.2) describe how one would ideally split the data into three portions:
  - One part being used to fit (or train) models,
  - A second (validation) part to choose a model,
  - And a third part to estimate the prediction error of the final model on a test dataset. Ideally, the test set is only used at the end of the data analysis.

- A typical split is 50% for training, 25% each for validation and testing.

3-Way Split

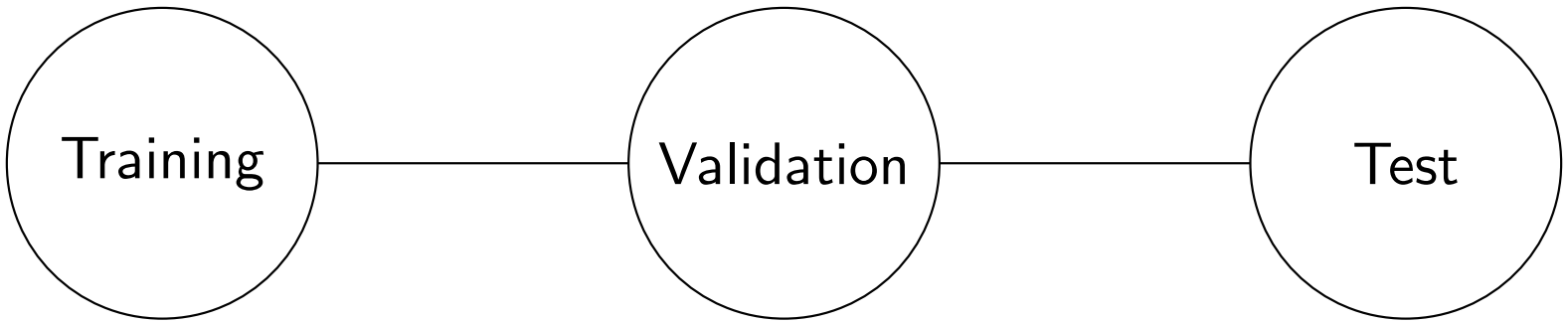
- Discuss whether PE evaluated in the validation stage is unbiased for the selected model.

  The PE might be biased in the validation stage too, especially if there are too many candidate models.

- Hastie et al. (2009, Section 7.2) describe how one would ideally split the data into three portions:
  - One part being used to fit (or train) models,
  - A second (validation) part to choose a model,
  - And a third part to estimate the prediction error of the final model on a test dataset. Ideally, the test set is only used at the end of the data analysis.

- A typical split is 50% for training, 25% each for validation and testing.

The 3-Way Split



- Training Data (50%): model fit;
- Validation Data (25%): model selection;
- Test Data (25%): final model evaluation.

Comments

- All Subsets Regression allows the analyst to choose among a number of "good" models. You may want to use this approach if the related computational cost is affordable.
- When there are a huge number of variables, it may be necessary to screen the variables before using all subsets regression. Backward stepwise regression can be used to eliminate the variables that are clearly of no use in predicting the response variable.

  However, backward elimination is not feasible if the number of variables exceeds or is near the number of observations.
- Forward stepwise regression is the most unreliable but often used in practice (as it's very simple). It does quickly pick a "good" but not always the best model.

Recommendation? screening + backward elimination

Comments

- All Subsets Regression allows the analyst to choose among a number of "good" models. You may want to use this approach if the related computational cost is affordable.
- When there are a huge number of variables, it may be necessary to screen the variables before using all subsets regression. Backward stepwise regression can be used to eliminate the variables that are clearly of no use in predicting the response variable.

  However, backward elimination is not feasible if the number of variables exceeds or is near the number of observations.
- Forward stepwise regression is the most unreliable but often used in practice (as it's very simple). It does quickly pick a "good" but not always the best model.

Recommendation? screening + backward elimination

Comments

- Your selected models are only the models satisfying some criterion based on the sample.

  A true model may not exist; or if it does exist, some variables in the true model may not observable in your sample.

- To obtain fair estimates of prediction accuracy, we need to take an independent test sample, and fit the selected model on this sample.

  Training/Validation/Test samples are independent samples from the same population.

Summer School in Astroinformatics: Ridge Regression

Le Bao

Department of Statistics
The Pennsylvania State University

Email: lebao@psu.edu

Motivation

LS estimates depend upon $(X'X)^{-1}$. We would have problems in computing β_{LS} if $X'X$ were singular or nearly singular, i.e., has zero eigen values.

- Too many predictors: it is not unusual to see the number of input variables greatly exceed the number of observations, e.g. micro-array data analysis, environmental pollution studies.
- Highly correlated predictors: small changes to the elements of X lead to large changes in $(X'X)^{-1}$, and hence large predictive intervals. (see examples in R).

Motivation

Simulate two predictors and response with p reflecting the dependence between X_1 and X_2.

```
p = 0
x1 = rnorm(1000, 0, 1)
x2 = x1*p + rnorm(1000, 0, 1) * (1-p)
y = x1 + x2 + rnorm(1000, 0, 1)
summary(lm(y~-1+x1+x2))
```

Biased Regression Methods

- One way out of this situation is to abandon the requirement of an unbiased estimator.

  β is still the solution of an optimization problem, but not necessarily unbiased.

- We assume only that X's and Y have been centered, so that we have no need for a constant term in the regression, and the scale of X does not matter:
  - X is a $n \times p$ matrix with standardized columns,
  - Y is a centered n-vector.

Ridge Regression

- Hoerl and Kennard (1970) proposed that potential instability in the LS estimator

$$\hat{\beta} = (X'X)^{-1}X'Y,$$

could be improved by adding a small constant value λ to the diagonal entries of the matrix $X'X$ before taking its inverse.

- The result is the ridge regression estimator

$$\hat{\beta}_{ridge} = (X'X + \lambda I_p)^{-1}X'Y$$

- Ridge regression places a particular form of constraint on the parameters (β's).

Ridge Regression

- $\hat{\beta}_{ridge}$ is chosen to minimize the penalized sum of squares:

$$\sum_{i=1}^{n}(y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

which is equivalent to minimization of

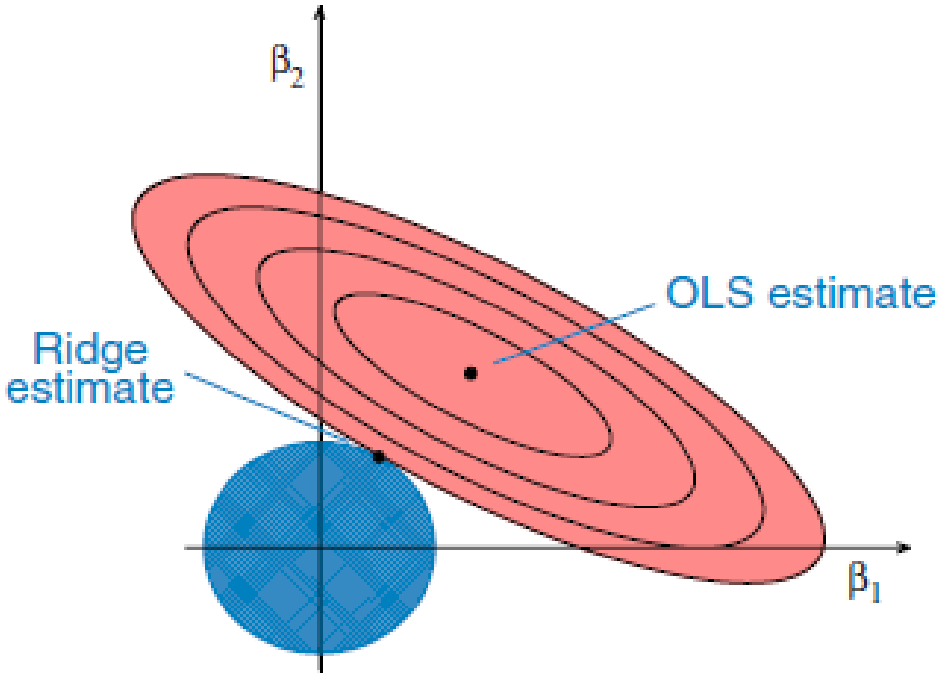
$$\sum_{i=1}^{n}(y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2$$

subject to, for some $c > 0$,

$$\sum_{j=1}^{p} \beta_j^2 < c$$

i.e. constraining the sum of the squared coefficients.

Ridge Regression with Two Parameters, β_1 and β_2



The Elements of Statistical Learning by FHT

The ellipses correspond to the contours of residual sum of squares (RSS): the inner ellipse has smaller RSS, and RSS is minimized at OLS estimates.

For $p = 2$, the constraint in Ridge regression corresponds to a circle, $\sum_{j=1}^{p} \beta_j^2$. The Ridge estimate is given by the point at which the ellipse and the circle touch.

Properties of Ridge Estimator

- $\hat{\beta}_{ridge}$ is a biased estimator of β.
- For orthogonal covariates, $X'X = nI_p$,

$$\hat{\beta}_{ridge} = \frac{n}{n+\lambda}\hat{\beta}_{ls}$$

  Hence, in this case, the ridge estimator always produces shrinkage towards 0.

- λ controls the amount of shrinkage.

Properties of Ridge Estimator

- The intercept β_0 has been left out of the penalty term because Y has been centered.
- Penalization of the intercept would make the procedure depend on the origin chosen of Y.
- Since the ridge estimator is linear, it is straightforward to calculate the variance-covariance matrix,

$$var(\hat{\beta}_{ridge}) = \sigma^2(X'X + \lambda I_p)^{-1}X'X(X'X + \lambda I_p)^{-1}$$

Effective Degrees of Freedom

- An important concept in shrinkage is the "effective" degrees of freedom associated with a set of parameters.
- In a ridge regression setting:
  - If we choose $\lambda = 0$, we have p parameters (since there is no penalization).
  - if λ is large, the parameters are heavily constrained and the degrees of freedom will effectively be lower, tending to 0 as $\lambda \to \infty$.

Effective Degrees of Freedom

- The effective degrees of freedom associated with $\beta_1, \beta_2, \ldots, \beta_p$ is defined as

$$df(\lambda) = tr(X(X'X + \lambda I_p)^{-1}X') = \sum_{j=1}^{p} \frac{d_j^2}{d_j^2 + \lambda},$$

  where d_j^2 are non-negative eigenvalues of $X'X$.

- Notice that $\lambda = 0$, which corresponds to no shrinkage, gives $df(\lambda) = p$ (as long as $X'X$ is non-singular).

- There is a 1:1 mapping between λ and the degrees of freedom, so in practice one may simply pick the effective degrees of freedom that one would like associated with the fit, and solve for λ.

- As an alternative to a user-chosen λ, cross validation is often used in choosing λ.

Ridge Solutions

- Whereas the least squares solutions $\hat{\beta}_{ls} = (X'X)^{-1}X'Y$ are unbiased if model is correctly specified, ridge solutions are biased,

$$E(\hat{\beta_{ridge}}) \neq \beta.$$

- However, at the cost of bias, Ridge reduces the variance, and thus might reduce the mean squared error (MSE).

$$\begin{aligned} ME &= Bias^2 + Variance, \\ E[(X\hat{\beta} - X\beta)^2] &= [E(X\hat{\beta} - X\beta)]^2 + Var(X\hat{\beta}). \end{aligned}$$

- Ridge solutions are hard to interpret, because it is not sparse.

  Sparse: some β's are set exactly to 0.

A Bayesian Formulation: Ridge Regression

- Consider the linear regression model with normal errors:

$$Y_i = \sum_{j=1}^{p} X_{ij}\beta_j + \epsilon_i$$

  ϵ_i is i.i.d. normal errors with mean 0 and known variance σ^2.

- Assume β_j has the prior distribution

$$\beta_j \sim_{iid} N(0, \sigma^2/\lambda)$$

- A large value of λ corresponds to a prior that is more tightly concentrated around zero, and hence leads to greater shrinkage towards zero.

A Bayesian Formulation: Ridge Regression

- Since a λ is applied to each β_j, people often standardize all of the covariates to make them comparable.
- The posterior is (try to derive by yourself)

$$\beta|Y \sim N(\hat{\beta}, \sigma^2(X'X + \lambda I_p)^{-1}X'X(X'X + \lambda I_p)^{-1}),$$

where

$$\hat{\beta} = \hat{\beta}_{ridge} = (X'X + \lambda I_p)^{-1}X'Y,$$

confirming that the posterior mean (and mode) of the Bayesian linear model correspond to the ridge regression estimator.

Ridge Estimator

- The ridge regression estimator can be viewed in four different ways:
  - A penalized least square estimator.
  - A shrinkage estimator that shrinks the least squares estimator toward the origin.
  - An estimator with restricted length that minimizes the residual sum of squares.
  - A Bayes estimator.

Summer School in Astroinformatics: Lasso and SCAD

Le Bao

Department of Statistics
The Pennsylvania State University

Email: lebao@psu.edu

$L1$ Penalties

- What if we constrain the $L1$ norm instead of the Euclidean ($L2$) norm?

$$\text{minimize:} \sum_{i=1}^{n}(y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2,$$

$$\text{Ridge subject to:} \sum_{j=1}^{p}(\beta_j)^2 < c.$$

$$\text{Lasso subject to:} \sum_{j=1}^{p}|\beta_j| < c.$$

- This is a subtle, but important change.

Lasso

- The least absolute shrinkage and selection operator, or *lasso*, as described in Tibshirani (1996), is a technique that has received a great deal of interest.
- As with ridge regression we assume the covariates are standardized.
- Lasso estimates of the coefficients (Tibshirani, 1996) achieve that

$$\min_{\beta}(Y - X\beta)'(Y - X\beta) + \lambda \sum_{j=1}^{p} |\beta_j|$$

so that the L2 penalty of ridge regression $\sum_{j=1}^{p}(\beta_j)^2$ is replaced by an L1 penalty $\sum_{j=1}^{p} |\beta_j|$.

Lasso

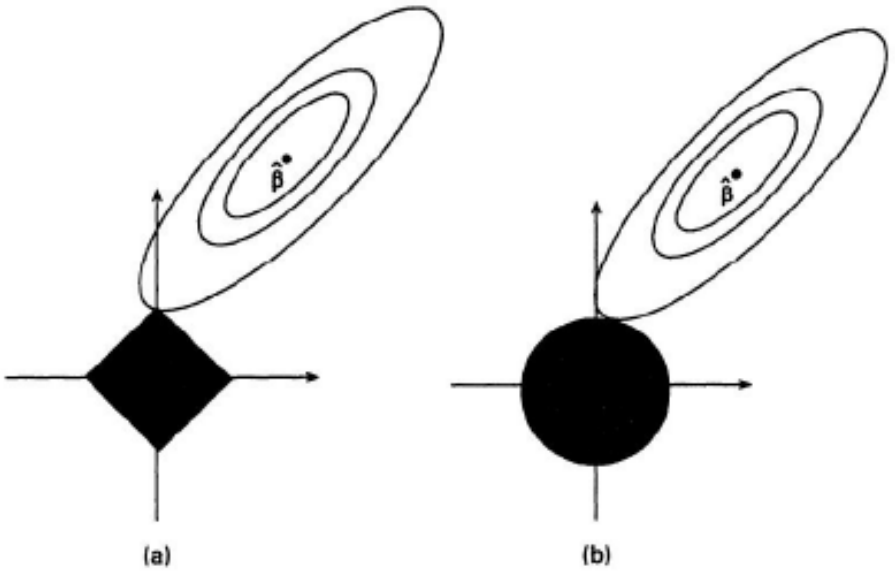
A lasso is a loop of rope that is designed to be thrown around a target and tighten when pulled.

The Elements of Statistical Learning by FHT

Lasso

- Let $c_0 = \sum_{j=1}^{p} |\hat{\beta}_{LS,j}|$ denote the absolute size of least squares estimate.
- Values of $0 < c < c_0$ cause shrinkage towards zero.
- If, for example, $c = c_0/2$ the average shrinkage of the least squares coefficients is 50%.
- If λ is sufficiently large, some of the coefficients are driven to zero, leading to a *sparse* mode.

Lasso and Ridge Regression with Two Parameters, β_1 and β_2



The Elements of Statistical Learning by FHT

The lasso performs $L1$ shrinkage, so that there are "corners" in the constraint which corresponds to a diamond.

If the sum of squares "hits" one of these corners, then the coefficient corresponding to the axis is shrunk to zero.

* Try to derived the condition of lasso solution sitting on the edges of the diamond instead of corners.

Lasso and Subset Selection

- As p increases the multidimensional diamond has an increasing number of corners, and so it is highly likely that some coefficients will be set equal to zero.
- Hence, the lasso performs shrinkage and (effectively) subset selection.
- Lasso solutions are sparse.
- In contrast with subset selection, Lasso performs a *soft thresholding*: as the smoothing parameter is varied, the sample path of the estimates moves continuously to zero.

LARS

- Lasso loss function is no longer quadratic, but is still convex:

$$\text{Minimize: } \sum_{i=1}^{n}(Y_i - \sum_{j=1}^{p} X_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

- Unlike Ridge, there is no analytic solution for the LASSO, because the solution is nonlinear in Y.

- The entire path of Lasso estimates for all values of λ can be efficiently computed through a modification of the *Least Angle Regression* (LARS) algorithm (Efron et al. 2003).

Group Lasso

- In some contexts, we may wish to treat a set of regressors as a group, for example, when we have a categorical covariate with more than two levels.
- The grouped lasso Yuan and Lin (2007) addresses this problem by considering the simultaneous shrinkage of (pre-defined) groups of coefficients.

Inference of Lasso Estimation

- The ordinary Lasso does not access the uncertainty of parameter estimation; standard errors for β's are not immediately available.
- For inference using the lasso estimator, various standard error estimators have been proposed:
  - Tibshirani (1996) suggested the bootstrap (Efron, 1979) for the estimation of standard errors and derived an approximate closed form estimate.
  - Fan and Li (2001) derived the sandwich formula in the likelihood setting as an estimator for the covariance of the estimates.
- However, all of the above approximate covariance matrices give an estimated variance 0 for predictors with $\hat{\beta}_j = 0$.

The Bayesian Lasso

- The Bayesian Lasso of Park and Casella (2008) provides valid standard errors of β's and provides more stable point estimates by using the posterior median.
- The lasso estimate is equivalent to the mode of the posterior distribution under the normal likelihood, and an independent Laplace (double exponential) prior:

$$\pi(\beta) = \frac{\lambda}{2} \exp(-\lambda|\beta_j|)$$

- The Bayesian Lasso estimates (posterior median) appear to be a compromise between the ordinary Lasso and the ridge regression.

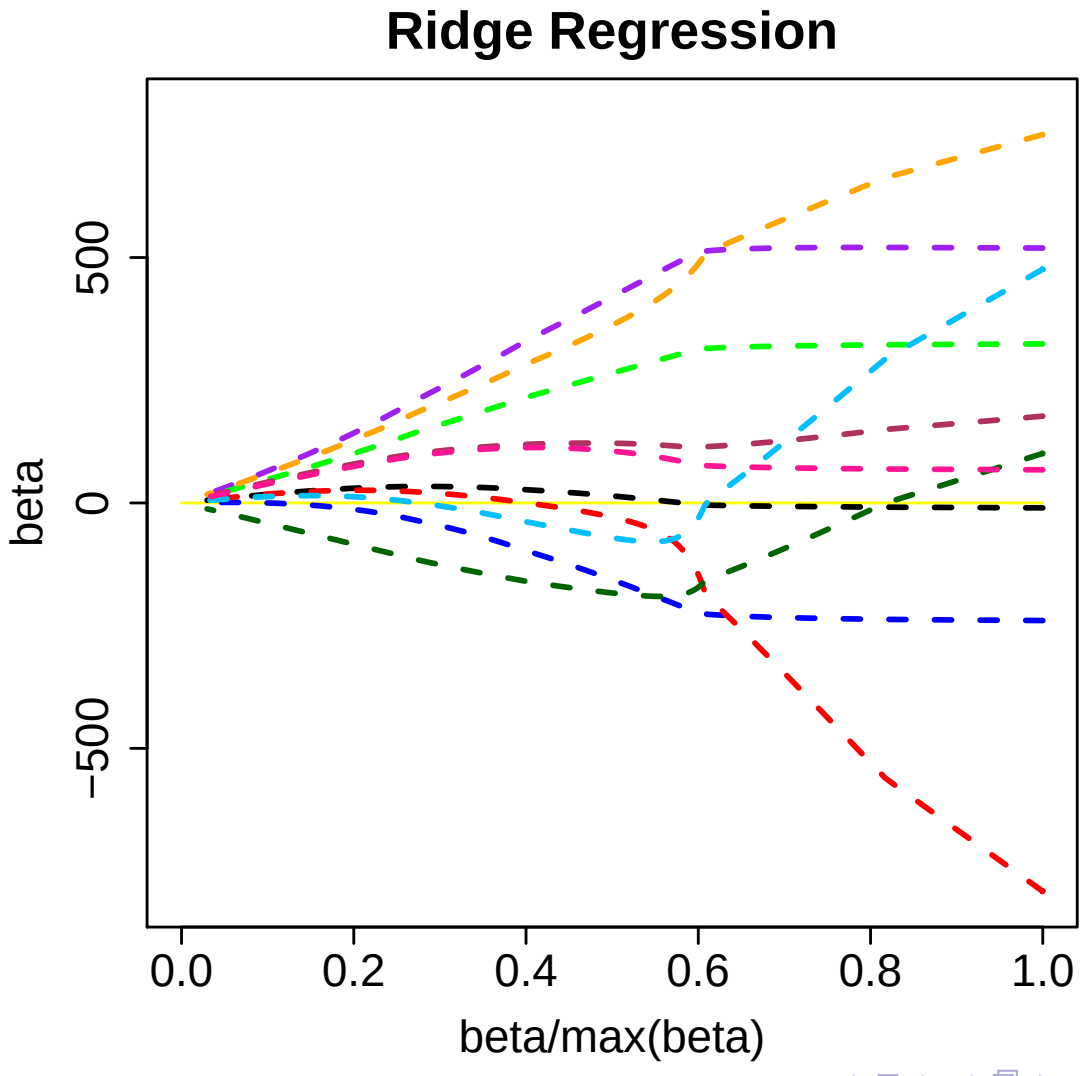
The Bayesian Lasso

- Park and Casella (2008) showed that the the posterior density was unimodal based on a conditional Laplace prior $\lambda|\sigma$, and the Gibbs algorithm was available for sampling the posterior distribution.

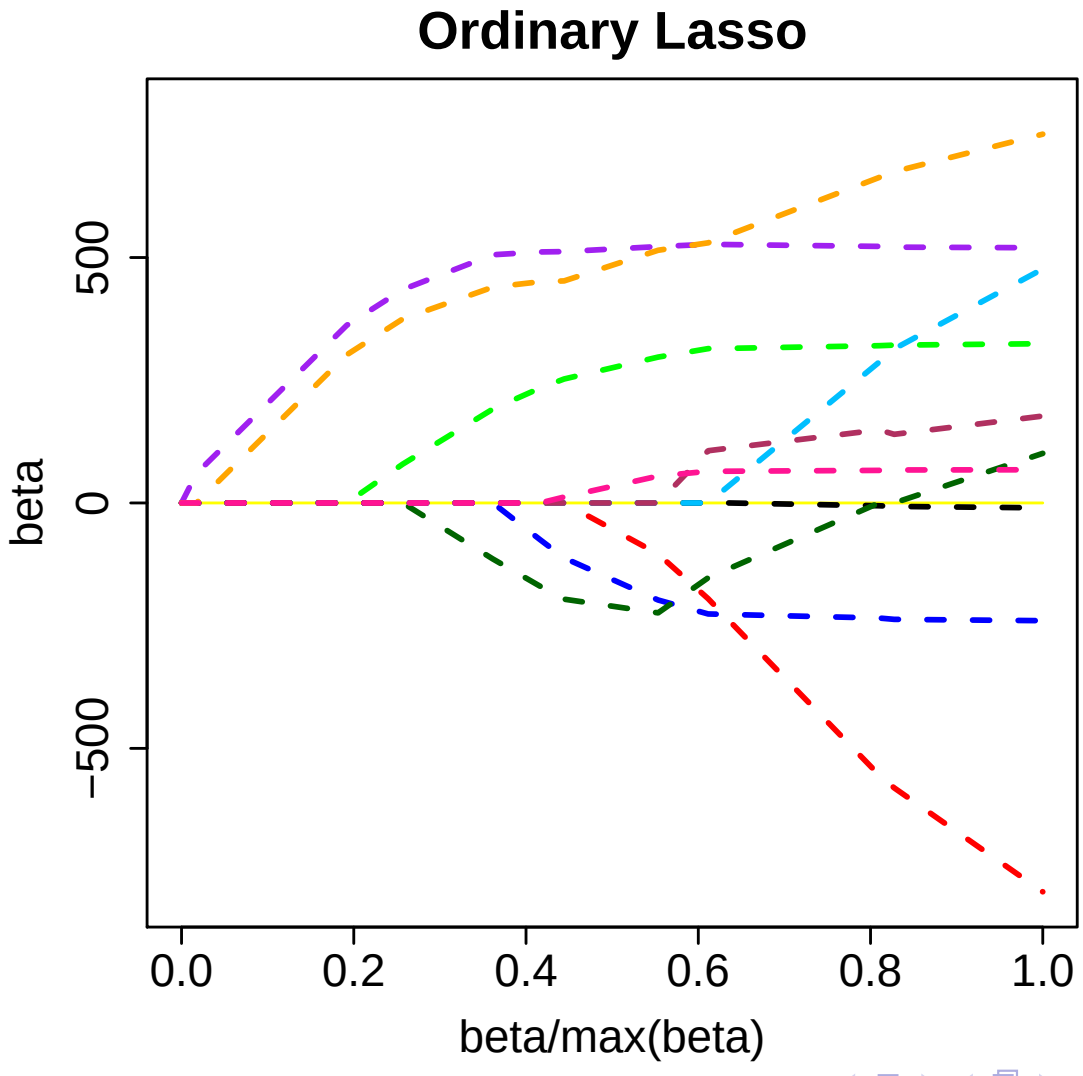
$$\pi(\beta|\sigma^2) = \prod_{j=1}^{p} \frac{\lambda}{2\sqrt{\sigma^2}} \exp(-\frac{\lambda|\beta_j|}{2\sqrt{\sigma^2}})$$

  and the noninformative marginal prior $\pi(\sigma^2) \propto 1/\sigma^2$.

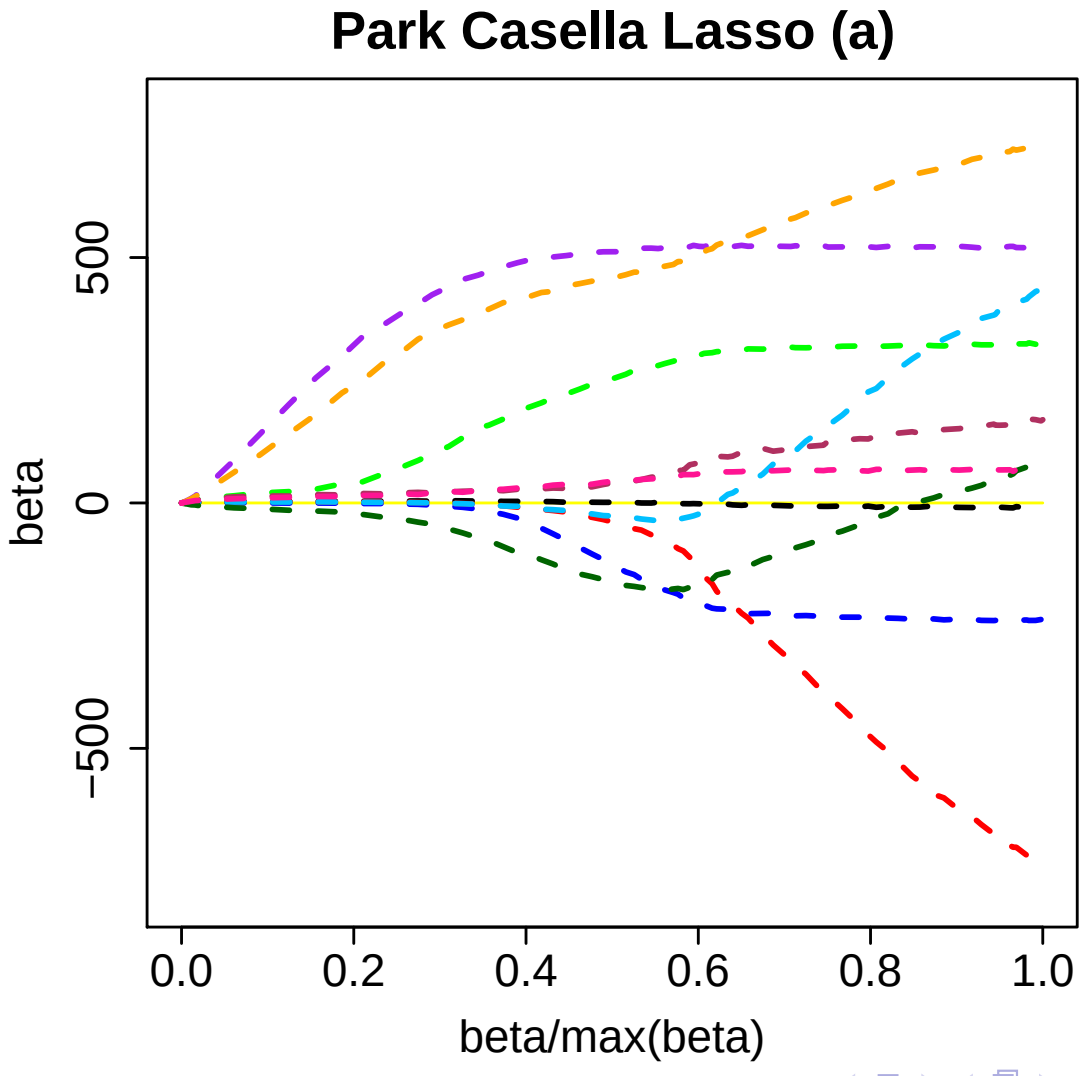
Lasso and Ridge Regression with Two Parameters, β_1 and β_2



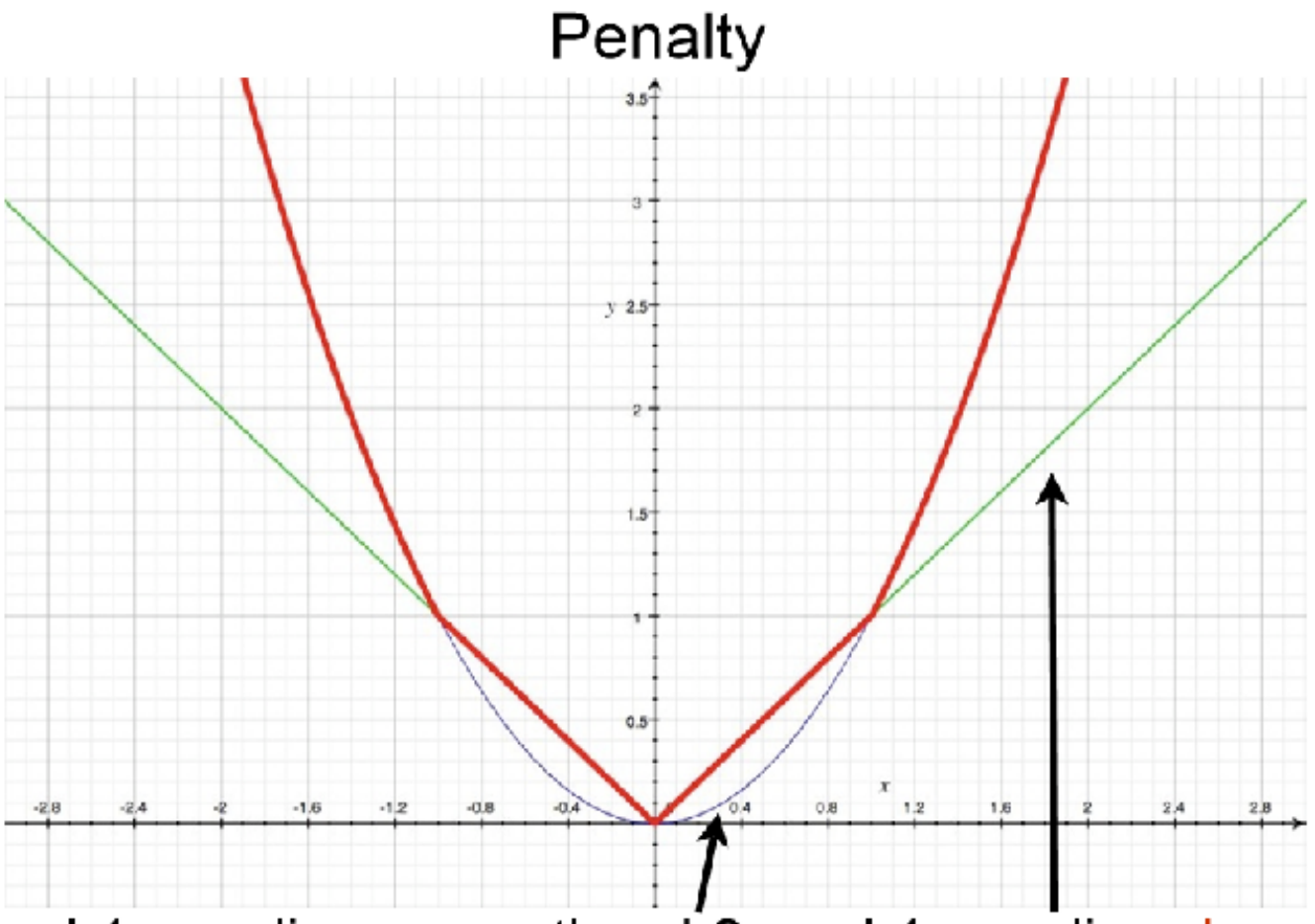
Lasso and Ridge Regression with Two Parameters, β_1 and β_2



Lasso and Ridge Regression with Two Parameters, β_1 and β_2



Lasso and Ridge Regression Penalties



L1 penalizes more than L2 when x is small (use this for sparsity)

L1 penalizes less than L2 when x is big (use this for robustness)

Elastic Net Regularization

It linearly combines the L1 and L2 penalties of the Lasso and ridge methods.

$$\text{Minimize:} \sum_{i=1}^{n}(Y_i - \sum_{j=1}^{p} X_{ij}\beta_j)^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=1}^{p} \beta_j^2$$

The naive version uses a two-stage procedure:

- first find the ridge regression coefficients,
- and then does a LASSO type shrinkage.

The Diabetes Data

- We use the diabetes data used in Efron et al. (2003).
- Ten baseline variables, age, sex, body mass index, average blood pressure, and six blood serum measurements were obtained for each of n $= 442$ diabetes patients, as well as the response of interest, a quantitative measure of disease progression one year after baseline.
- Each variable has been standardized to have zero mean and one standard deviation.

References: Efron, Hastie, Johnstone and Tibshirani (2003) "Least Angle Regression" (with discussion) Annals of Statistics.

The Diabetes Data

Install packages: "lars", load the library, and get data from "lars":

- library(lars)
- data(diabetes)
- attach(diabetes)

The Diabetes Data

The diabetes data frame are as follows:

- diabetes$x is a matrix with 10 columns, which has been standardized to have unit $L2$ norm in each column and zero mean;
- diabetes$x2 is a matrix with 64 columns which consists of x plus certain interactions;
- diabetes$y is numeric vector.

Useful R functions

- Fit a set of linear models and run stepwise model selection: *step*() or *regsubsets*().
- Fit a linear model by ridge regression: *lm.ridge*(). (*MASS*)
- Fit a linear model by lasso: *cv.lars*(). (*lars*)
- Run K-fold cross validation: *crossval*(). (*bootstrap*)
- A more generic function for ridge regression, lasso, and cross validation: *glmnet*() and *cv.glmnet*(). (*glmnet*)

Penalized least squares and Penalized likelihood

- The general form of penalized least squares is as follows:

$$\min_{\beta} \left((Y - X\beta)'(Y - X\beta) + \sum_{j=1}^{p} p_\lambda(|\beta_j|) \right).$$

- As the general extension of the penalized least squares, penalized likelihood is defined as follows:

$$\min_{\beta} \left(-l(\beta) + \sum_{j=1}^{p} p_\lambda(|\beta_j|) \right),$$

where $l(\beta)$ is the log-likelihood function, $p_\lambda(.)$ is a penalty function indexed by the regularization parameter $\lambda \geq 0$.

$p_\lambda(|\beta_j|)$ is required to be nondecreasing in $|\beta_j|$.

- Different regularization methods correspond different penalties.

Penalized least squares and Penalized likelihood

- Ridge Regression (L_2 Penalty): $p_\lambda(|\beta_j|) = \lambda \times |\beta_j|^2$.
- Lasso (L_1 Penalty): $p_\lambda(|\beta_j|) = \lambda \times |\beta_j|$.
  When X_j is weakly related with Y, Lasso pulls β_j to zero faster than ridge regression.
- Best subset selection (L_0 Penalty): $p_\lambda(|\beta_j|) = \lambda \times I(|\beta_j| > 0)$.
  It possesses variable selection feature and gives a nice interpretation of best subset selection, but it is a non-convex optimization problem.
- Bridge Regression (L_q Penalty): $p_\lambda(|\beta_j|) = \lambda \times |\beta_j|^q$, and $0 < q < 2$.

Due to the nonlinearity of the solution, the effective degrees of freedom cannot be defined as

$$df(\lambda) = tr(X(X'X + \lambda I_p)^{-1}X')$$

Properties of penalized estimators

To get a better penalty, Fan and Li (2001) advocate penalty functions that give estimators with three properties.

- Sparsity: The resulting estimator automatically sets small estimated coefficients to zero to accomplish variable selection and reduce model complexity.
- Unbiasedness: The resulting estimator is nearly unbiased, especially when the true coefficient j is large, to reduce model bias.
- Continuity: The resulting estimator is continuous in the data to reduce instability in model prediction (Breiman (1996)).

SCAD

Under these considerations, Fan (1997) and Fan and Li (2001) introduce the smoothly clipped absolute deviation (SCAD), whose derivative is given by:

$$p'_\lambda = \lambda \times \left(I(t \leq \lambda) + \frac{(a\lambda - t)_+}{(a-1)\lambda} I(t > \lambda) \right).$$

A penalty of similar spirit is the minimax concave penalty (MCP) in Zhang (2009), whose derivative is given by:

$$p'_\lambda = \frac{(a\lambda - t)_+}{\lambda}.$$

These two penalities both satisfy the previous three requirements.