Data Mining (W4240 Section 001)
Subset Selection

Giovanni Motta

Columbia University, Department of Statistics

November 9, 2015

Outline

Motivation: Linear Regression

Subset Selection

Optimism

Model Selection Criteria

Example

Outline

Motivation: Linear Regression

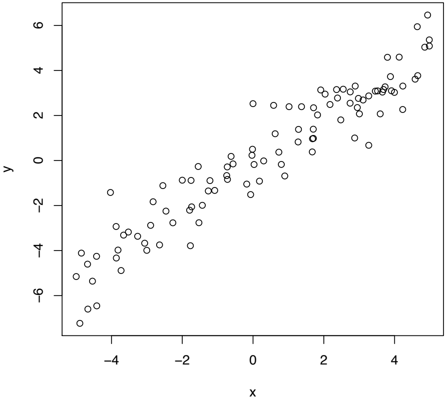
Subset Selection

Optimism

Model Selection Criteria

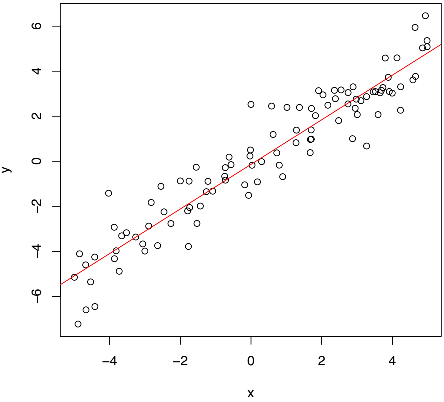
Example

Linear Regression



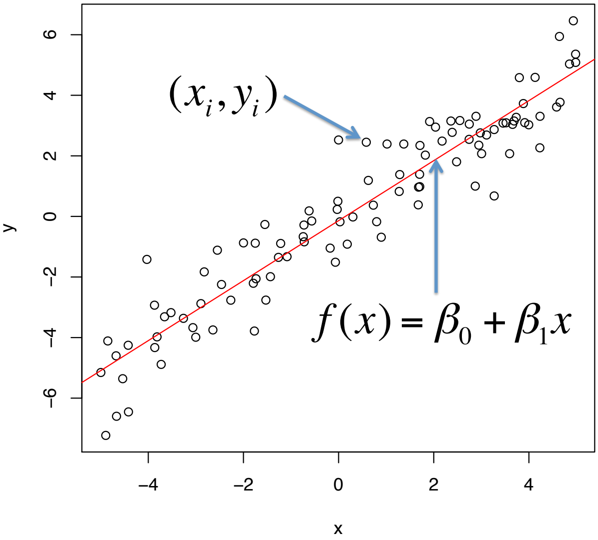
Training data are the set of inputs and outputs, $\mathcal{T} = \{(x_i, y_i)\}_{i=1}^{n}$

Linear Regression



In *linear regression*, the goal is to predict y from x using a linear function

Linear Regression



Linear Regression

Let's begin with some linear regression in R.

```
> n <- 100
> p <- 95
> x <- rnorm(n*p)
> dim(x) <- c(n,p)
> y <- x[,1] - 1.2*x[,2] + rnorm(n)
> fit.lm <- lm(y ~ x)
```

What are the coefficients? What about the residuals? Let's do this a few times.

High Dimensional Data

This is an example of a high-dimensional problem: $n \approx p$.

What are some legitimate assumptions for this type of problem?

- how many covariates actually matter?
- why would some not matter?
- should we fit a simple model or a complex model?
- how can we do it?

High Dimensional Data

This is an example of a high-dimensional problem: $n \approx p$.

What are some legitimate assumptions for this type of problem?

- how many covariates actually matter?
- why would some not matter?
- should we fit a simple model or a complex model?
- how can we do it?

Note: this $n \approx p$ problem motivates *subset selection*, but it is useful in many settings.

Outline

Motivation: Linear Regression

Subset Selection

Optimism

Model Selection Criteria

Example

Subset Selection

Pick the best k ($\leq p$) covariates to use in linear regression

Subset Selection

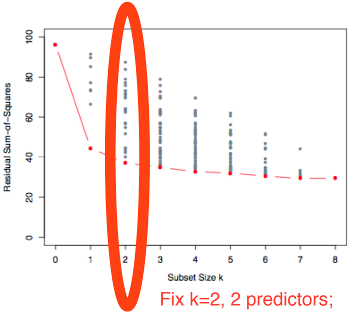
Pick the best k ($\leq p$) covariates to use in linear regression

Why?

- *Predictive Accuracy:* Linear least squares estimator has low bias, high variance. Reduce number of covariates, get a bit more bias but much less variance.
- *Interpretability:* Which variables matter? Which do not? Interpretability allows your model to say something about the data vs. just giving a prediction.

Subset Selection

How to pick the best k $(\leq p)$ covariates for linear regression?



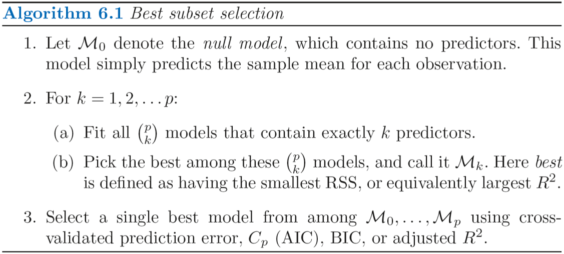
Fix k=2, 2 predictors;
AmongC(p,2) subsets, fit a OLR, compute ResidualSSE
Choose the one with min

Best Subset Selection:

- enumerate possible subsets in a smart way for each k
- for each k, select subset that minimizes RSS
- pick best k: cross-validation or other model selection methods
- good method for $p < 30$ or 40

Subset Selection

How to pick the best k ($\leq p$) covariates for linear regression?



Algorithm 6.1 *Best subset selection*

1. Let $\mathcal{M}_0$ denote the *null model*, which contains no predictors. This model simply predicts the sample mean for each observation.

2. For $k = 1, 2, \ldots p$:

   (a) Fit all $\binom{p}{k}$ models that contain exactly k predictors.

   (b) Pick the best among these $\binom{p}{k}$ models, and call it $\mathcal{M}_k$. Here *best* is defined as having the smallest RSS, or equivalently largest R^2.

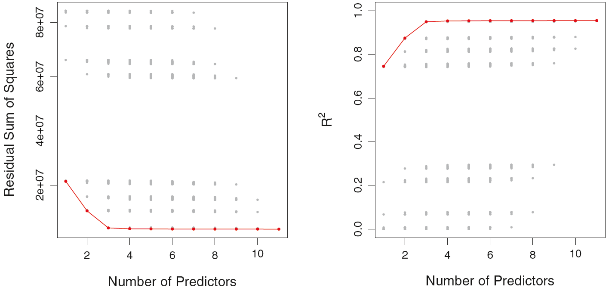
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2.

Best Subset Selection:

- at each step, fit $\binom{p}{k}$ models
- $\sum_{k=0}^{p} \binom{p}{k} = 2^p$ models

Subset Selection: credit card dataset

How to pick the best k $(\leq p)$ covariates for linear regression?

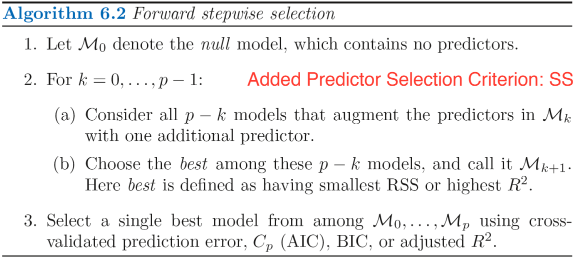


Best Subset Selection:

- for $k = 1, \ldots, 11$, fit $\binom{11}{k}$ models
- $2^{11} = 2048$ models !

Subset Selection

How to pick the best k ($\leq p$) covariates for linear regression?



```
Algorithm 6.2 Forward stepwise selection

1. Let $\mathcal{M}_0$ denote the *null* model, which contains no predictors.
2. For $k = 0, \ldots, p-1$:
   (a) Consider all $p-k$ models that augment the predictors in $\mathcal{M}_k$
       with one additional predictor.
   (b) Choose the *best* among these $p-k$ models, and call it $\mathcal{M}_{k+1}$.
       Here *best* is defined as having smallest RSS or highest $R^2$.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-
   validated prediction error, $C_p$ (AIC), BIC, or adjusted $R^2$.
```

Added Predictor Selection Criterion: SSR or R^2

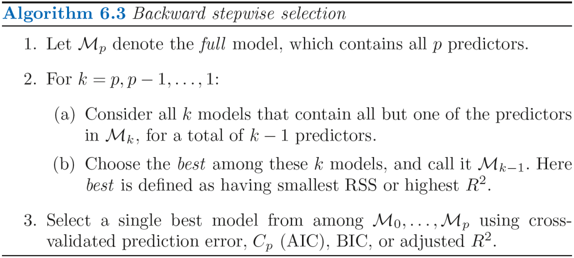
Stopping Criterion: Cp AIC BIC

Forward stepwise Selection:

- at each step, fit $p - k$ models
- $1 + \sum_{k=0}^{p-1}(p-k) = 1 + \frac{p(p+1)}{2}$ models
- 67 rather that 2048 models

Subset Selection

How to pick the best k $(\leq p)$ covariates for linear regression?



Algorithm 6.3 *Backward stepwise selection*

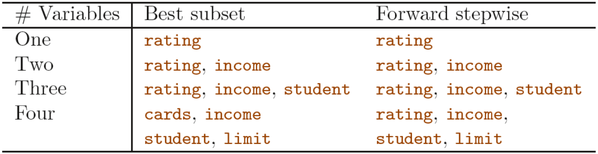
```
1. Let $\mathcal{M}_p$ denote the *full* model, which contains all $p$ predictors.

2. For $k = p, p-1, \ldots, 1$:

    (a) Consider all $k$ models that contain all but one of the predictors
        in $\mathcal{M}_k$, for a total of $k-1$ predictors.

    (b) Choose the *best* among these $k$ models, and call it $\mathcal{M}_{k-1}$. Here
        *best* is defined as having smallest RSS or highest $R^2$.

3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-
   validated prediction error, $C_p$ (AIC), BIC, or adjusted $R^2$.
```

Backward stepwise Selection:

- at each step, fit $p - k$ models
- $1 + \sum_{k=0}^{p-1}(p-k) = 1 + \frac{p(p+1)}{2}$ models
- 67 rather that 2048 models
- Backward selection requires that $n > p$. In contrast, forward stepwise can be used even when $n < p$, and so is the only viable subset method when p is very large.

Subset Selection: credit card dataset

How to pick the best k ($\leq p$) covariates for linear regression?



# Variables	Best subset	Forward stepwise
One	rating	rating
Two	rating, income	rating, income
Three	rating, income, student	rating, income, student
Four	cards, income student, limit	rating, income, student, limit

- forward stepwise tends to do well in practice,
- HOWEVER: it is not guaranteed to find the best possible model out of all 2^p models containing subsets of the p predictors.

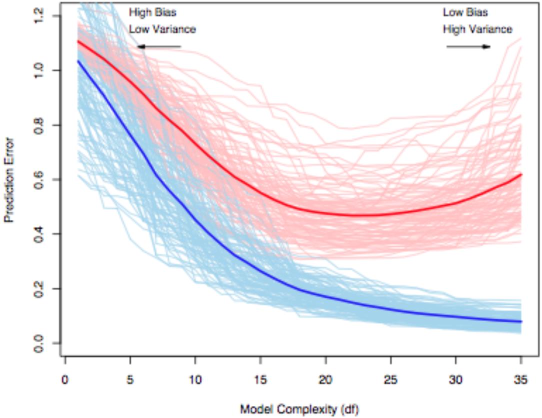
Model Selection

Cross-validation is not always the answer:

- here n is small compared to p *by definition*
- cross-validation may be too expensive since you have to fit all possible model combinations

Other methods like AIC and BIC adjust training error to try to estimate testing error

Training Error



The training error is *optimistic*: it under estimates the testing error. By how much? (Use corrected training error in place of testing error!)

Outline

Motivation: Linear Regression

Subset Selection

Optimism

Model Selection Criteria

Example

Training Error Optimism

Training data: $\mathcal{T} = \{(x_1, y_1), \dots, (x_n, y_n)\}$

New data: X^0, Y^0

Generalization error (extra-sample error):

$$\mathrm{Err}_{\mathcal{T}} = \mathbb{E}_{X^0,Y^0}[L(Y^0, \hat{f}(X^0)) \,|\, \mathcal{T}]$$

Expected error (we asked about re: bootstrap):

$$\mathrm{Err} = \underline{\mathbb{E}_{\mathcal{T}}}\mathbb{E}_{X^0,Y^0}[L(Y^0, \hat{f}(X^0)) \,|\, \mathcal{T}]$$

Training error:

$$\mathrm{Err}_{train} = \frac{1}{n}\sum_{i=1}^{n} L(y_i, \hat{f}(x_i))$$

Training Error Optimism

To understand training error,

$$\mathrm{Err}_{train} = \frac{1}{n}\sum_{i=1}^{n} L(y_i, \hat{f}(x_i)),$$

look at *in-sample error* (not a training error!):

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

(Fix covariates, randomize responses.)

The *optimism* is the difference between Err_{in} and Err_{train}:

$$\mathrm{op} \equiv \mathrm{Err}_{in} - \mathrm{Err}_{train}$$

The average optimism is the expectation over the training sets

$$\mathbb{E}_y(\mathrm{op})$$

Training Error Optimism

In-sample error vs. training sample error vs. extra-sample error:

- **Extra-sample error:** expected error over new covariates and new responses
  - need to approximate distribution of responses and covariates
- **In-sample error:** expected error over new responses for given covariates
  - current covariate sample approximates true distribution
  - expectation over *new* responses eliminates bias from correlation between observed responses and fitted responses
- **Training sample error:** error averaged over training samples
  - correlation between y_i and $\hat{y}_i$ causes underestimate of error
  - but, hey, it is easy to compute

Training Error Optimism

Can show for loss functions,

$$\mathbb{E}_y(\mathrm{op}) = \frac{2}{n}\sum_{i=1}^{n}\mathrm{Cov}(\hat{y}_i, y_i)$$

If method overfits, this value will be high.

$$\mathbb{E}_y(\mathrm{Err}_{in}) = \mathbb{E}_y(\mathrm{Err}_{train}) + \frac{2}{n}\sum_{i=1}^{n}\mathrm{Cov}(\hat{y}_i, y_i)$$

In the case of a linear model,

$$\mathbb{E}_y(\mathrm{Err}_{in}) = \mathbb{E}_y(\mathrm{Err}_{train}) + \boxed{\frac{2p}{n}\sigma_\epsilon^2}$$

Can use this to get in-sample estimates of prediction error

Estimating In-Sample Prediction Error

Model selection criteria vs. cross-validation:

- **Cross-validation:**
  - possibly more accurate
  - no need for asymptotic approximations (is n large enough to justify asymptotics?)
  - more flexible (can be used for things other than MLE)
- **Model selection criteria:**
  - often easy to compute
  - theoretically justifiable

Outline

Motivation: Linear Regression

Subset Selection

Optimism

Model Selection Criteria

Example

Estimating In-Sample Prediction Error

In general, an estimate of the in-sample error is the training sample error plus an estimate of the optimism,

$$\hat{\mathrm{Err}}_{in} = \mathrm{Err}_{train} + \hat{\mathrm{op}}$$

Suppose that we use a log-likelihood loss function ($-$squared error $=$ Gaussian log-likelihood). The *Akaike Information Criterion* is an asymptotic approximation for Err_{in}:

$$-2\mathbb{E}[\log\,\mathrm{Pr}_{\hat{\theta}}] \approx -\frac{2}{n}\sum_{i=1}^{n}\log\,\mathrm{Pr}_{\hat{\theta}}(y_i) + 2\frac{d(\alpha)}{n}$$

$$AIC(\alpha) = \mathrm{Err}_{train}(\alpha) + 2\frac{d(\alpha)}{n}\hat{\sigma}_{\epsilon}^{2}$$

Here $\hat{\theta}$ is the MLE estimate. Choose α that minimizes $AIC(\alpha)$.

Estimating In-Sample Prediction Error

Are there other ways to estimate $\hat{\omega}$? Of course.

The *Bayesian Information Criterion* uses the approximation $\log(n)\frac{d(\alpha)}{n}\hat{\sigma}_\epsilon^2$ instead of $2\frac{d(\alpha)}{n}\hat{\sigma}_\epsilon^2$,

$$
\begin{aligned}
AIC(\alpha) &= \mathrm{Err}_{train}(\alpha) + 2\frac{d(\alpha)}{n}\hat{\sigma}_\epsilon^2 \\
BIC(\alpha) &= \frac{n}{\sigma_\epsilon^2}\left[\mathrm{Err}_{train}(\alpha) + (\log n)\frac{d(\alpha)}{n}\hat{\sigma}_\epsilon^2\right]
\end{aligned}
$$

BIC:

- chooses right model size as $n \to \infty$
- ...but chooses too simple models when n is small

AIC:

- chooses better models with small n
- ...but chooses too complicated models when n is large

Adjusted R^2

Recall from linear regression:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

R^2 explains the reduction in variance of a model.... but a model with a large p might be overfitting.

We can adjust the R^2 for the number of explanatory terms relative to the number of data points: with more data, more explanatory terms are acceptable.

$$\bar{R}^2 = 1 - (1 - R^2)\frac{n-1}{n-p-1}$$

The adjusted R^2 corrects for the extra degrees of freedom associated with more predictors.

Estimating In-Sample Prediction Error

Model selection criteria vs. cross-validation:

- **Model selection criteria:**
  - often easy to compute
  - theoretically justifiable
- **Cross-validation:**
  - possibly more accurate
  - no need for asymptotic approximations (is n large enough to justify asymptotics?)
  - more flexible (can be used for things other than MLE)

Outline

Motivation: Linear Regression

Subset Selection

Optimism

Model Selection Criteria

Example

Example: Prostate Data

Data in Prostate.txt (also available on ESL website)

Predictors (columns 1–8): lcavol (log cancer volume), lweight (log weight), age, lbph (log amount of benign prostatic hyperplasia), svi (seminal vesicle inversion), lcp (log capsular penetration), gleason, pgg45 (percentage of Gleason scores 4 or 5)

outcome (column 9): lpsa (level of prostate-specific antigen)

train/test indicator (column 10)

```
> prostate <- read.table("Prostate.txt",header=TRUE, sep="\t")
> names(prostate)
 [1] "X"       "lcavol"  "lweight" "age"
 [5] "lbph"    "svi"     "lcp"     "gleason"
 [9] "pgg45"   "lpsa"    "train"
> prostate.train <- prostate[prostate$train==T,2:10]
> prostate.test <- prostate[prostate$train==F,2:10]
```

Example: Prostate Data

```
> prostate.lm <- lm(lpsa ~ lcavol + lweight + age + lbph
  + svi + lcp + gleason + pgg45, data=prostate.train)
> # Other way:
> # prostate.lm <- lm(lpsa ~., data=prostate.train)
> # Exclude intercept by:
> # prostate.lm <- lm(lpsa ~ lcavol + lweight + age + lbph
  + svi + lcp + gleason + pgg45 - 1, data=prostate.train)
> y.pred.lm <- predict(prostate.lm,prostate.test)
> mean((y.pred.lm-prostate.test$lpsa)^2)
[1] 0.521274
```

Note: the data in ESL was scaled before use, so $\hat{\beta}$ differs

Best Subset Selection

Use the package `leaps`

```
> library(leaps)
> prostate.bss <- regsubsets(lpsa ˜ ., data=prostate.train)
> # Let’s see the outputs
> summary(prostate.bss)
> coef(prostate.bss,1:4)
> plot(prostate.bss, scale="bic")
> # Get a prediction
> coef.bss <- coef(prostate.bss,2)
> y.pred.bss <- coef.bss[1]
  + coef.bss[2]*prostate.test$lcavol
  + coef.bss[3]*prostate.test$lweight
> mean((y.pred.bss-prostate.test$lpsa)ˆ2)
[1] 0.4924823
```

Forward and Backward Subset Selection

What happens if $p > 40$? We can't search all subsets...

Forward stepwise selection:

- start with intercept
- add in one predictor that improves the fit the most
- repeat until we run out of predictors
- select k through cross-validation, AIC, BIC, adjusted R^2
- "fit improvement" determined by $F-$statistics or AIC scores

This is called a *greedy algorithm*

Forward and Backward Subset Selection

Why greedy algorithms?

- computational: only search through $\mathcal{O}(p \min(n,p))$ subsets (at most)
- statistical: more constrained search means some additional estimator bias, but less variance

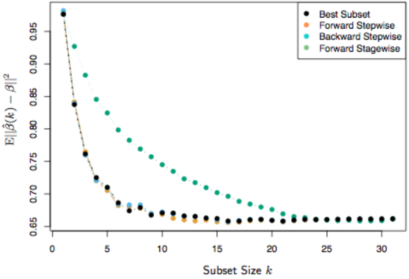


FIGURE 3.6. *Comparison of four subset-selection techniques on a simulated linear regression problem $Y = X^T\beta + \varepsilon$. There are $N = 300$ observations on $p = 31$ standard Gaussian variables, with pairwise correlations all equal to 0.85. For 10 of the variables, the coefficients are drawn at random from a $N(0, 0.4)$ distribution; the rest are zero. The noise $\varepsilon \sim N(0, 6.25)$, resulting in a signal-to-noise ratio of 0.64. Results are averaged over 50 simulations. Shown is the mean-squared error of the estimated coefficient $\hat{\beta}(k)$ at each step from the true β.*

Forward and Backward Subset Selection

Backward stepwise selection:

- start with all predictors
- remove one that contributes the least
- repeat until we are left with the intercept
- select k through cross-validation, AIC, BIC, adjusted R^2
- “fit improvement” determined by $F-$statistics or AIC scores
- note: only works if $n > p$

Forward and Backward Subset Selection

Use the function step (you can also use regsubsets() with method="forward" or method="backward")

```
> prostate.fwd <- step(prostate.lm)
> summary(prostate.fwd)
> y.pred.fwd <- predict(prostate.fwd,prostate.test)
> mean((y.pred.fwd-prostate.test$lpsa)^2)
[1] 0.5165135
```

...or we can step(...   , direction="backward").

Forward Stagewise Regression

Forward stagewise regression:

- start with intercept as mean
- compute residuals based on current model
- compute correlation between each covariate and residuals
- compute simple linear regression on residuals against variable with highest correlation
- add coefficient to existing coefficient for that variable
- repeat until no correlation between residuals and variables

Takes many steps to converge, but often good for very high dimensional problems; can be implemented in `lars` package