Variable Selection

Lecture 16

Alexandra Chronopoulou

ILLINOIS

COLLEGE OF LIBERAL ARTS & SCIENCES

Department of Statistics
101 Illini Hall, MC-374
725 S. Wright St.
Champaign, IL 61820-5710

©Alexandra Chronopoulou. Do not distribute without permission of the author.

Variable Selection

- Consider a MLR model $Y \sim 1 + X_1 + X_2 + \ldots + X_p$ where we have p non-intercept predictors.
- In many applications nowadays, we have a lot of explanatory variables, i.e., p is large and we could even have $p >> n$, but only a small portion of the p variables are believed to be relevant to Y.
- Of interest is to find the following subset of the p predictors:

$$S = \{j : \beta_j \neq 0\}$$

Variable Selection

- In some applications as sales prediction, the key question we need to answer is to identify this set S, e.g., which variables among the p variables are really effective for boosting the sales (Y).
- If our goal is simply to do well on prediction, then should we care about variable selection?

  Recall that the LS estimate $\hat{\beta}$ is unbiased, i.e., estimates for irrelevant $\hat{\beta}_j$ (with $j \in S^c$) will eventually go to zero anyway.

  To understand this, let's examine the training and the test errors.

Test Error vs. Training Error

- Training data: $(\mathbf{x}_i, y_i)_{i=1}^n$
- Test data: $(\mathbf{x}_i, y_i^*)_{i=1}^n$ is an independent (imaginary) data set collected at the same location $\mathbf{x}_i$'s (also known as in-sample prediction)
- Assume the data comes from a linear model:
  $\mathbf{y}_{n\times 1}$, $\mathbf{y}^*_{n\times 1}$ are i.i.d $\sim N_n(\mu, \sigma^2\mathbf{I}_n)$ and $\mu = \mathbf{X}\beta$
- We can also write:

$$\mathbf{y} = \mathbf{X}\beta + \varepsilon$$
$$\mathbf{y}^* = \mathbf{X}\beta + \varepsilon^*$$

  with $\varepsilon_{n\times 1}$, $\varepsilon^*_{n\times 1}$ $\quad i.i.d \sim \mathcal{N}_n(\mathbf{0}, \sigma^2\mathbf{I}_n)$ are independent.

Mean square Testing Error and Mean square Training Error

$$
\begin{aligned}
\mathbb{E}(\text{Test Error})^2 &= \mathbb{E}||\mathbf{y}^* - \mathbf{X}\hat{\beta}||^2 \\
&= \mathbb{E}||(\mathbf{y}^* - \mathbf{X}\beta) + (\mathbf{X}\beta - \mathbf{X}\hat{\beta})||^2 \\
&= \mathbb{E}||\mathbf{y}^* - \mu||^2 + \mathbb{E}||\mathbf{X}\beta - \mathbf{X}\hat{\beta}||^2 \\
&= \mathbb{E}||\varepsilon^*||^2 + Tr(\mathbf{X}Cov(\hat{\beta})\mathbf{X}^\top) \\
&= n.\sigma^2 + \sigma^2 Tr\mathbf{H} = n.\sigma^2 + p.\sigma^2
\end{aligned}
$$

$$
\begin{aligned}
\mathbb{E}(\text{Train Error})^2 &= \mathbb{E}||\mathbf{y} - \hat{\mathbf{y}}||^2 = \mathbb{E}||(\mathbf{I} - \mathbf{H})\mathbf{y}||^2 \\
&= Tr((\mathbf{I} - \mathbf{H})Cov(\mathbf{y})(\mathbf{I} - \mathbf{H})^\top) \\
&= \sigma^2 Tr((\mathbf{I} - \mathbf{H})) = (n - p).\sigma^2
\end{aligned}
$$

Mean square testing error and Mean square training error

From the previous equations we can conclude:

- Testing error increases with p
- Training error decreases with p
- When adding more variables (p large) the testing error increases. If our goal is pure prediction, adding more variables to matrix $\mathbf{X}$ is not the best option. We should remove some irrelevant variables.
- The analysis on the previous slide might indicate that the best model (i.e., the one with the smallest expected test error), is the intercept-only model with $p = 0$.
- This of course is not true. The previous analysis is based on the assumption that the mean of $\mathbf{y}$ is in the column space of $\mathbf{X}$, i.e., there exists some coefficient vector β such that $\mu = \mathbf{X}\beta$. In general, we run a linear regression model using only a subset of the columns of $\mathbf{X}$. This means there will be an additional Bias term.

Model Index γ

- Index each model (i.e., each subset of the p variables) by a p-dimensional binary vector γ:

$$\gamma = (\gamma_1, \gamma_2, \ldots, \gamma_p), \quad \gamma_j = 0/1$$

  where $\gamma_j = 1$ indicates that X_j is included in the model, and $\gamma_j = 0$ otherwise.

- So there are a total of 2^p possible subsets or sub-models. In particular

$$\gamma = (1, 1, \ldots, 1)$$

  refers to the full model including all p variables (largest dim), and

$$\gamma = (0, 0, \ldots, 0)$$

  refers to the intercept-only model (smallest dim).

Model Index γ

Suppose that $\mu = \mathbf{X}\beta$ where μ is the mean of $\mathbf{y}$. If we fit the data $\mathbf{y}$ with respect to model γ, i.e., we fit a linear model with a sub-design matrix X_γ where X_γ contains only columns from $\mathbf{X}$ such that $\gamma_j = 1$.

We can show that the Testing Error and the Training error for model γ are:

$$\mathbb{E}(\text{Test Error}) = n\sigma^2 + p\sigma^2 + Bias_\gamma$$
$$\mathbb{E}(\text{Training Error}) = n\sigma^2 - p\sigma^2 + Bias_\gamma$$

Bigger model (i.e., p large) $\rightarrow$ small Bias, but large variance ($p\sigma^2$); Smaller model (i.e., p small) $\rightarrow$ large Bias, but small variance ($p\sigma^2$). So to reduce the test error (i.e., prediction error), the key is to find the best trade-off between Bias and Variance.

Model selection procedures

- Testing-based procedures: Select best model based on statistical tests for model comparison.
- Criterion-based procedures: Select best model based on an information criteria (combining model fit and model complexity) for model comparison.

Backward elimination

– Start with all the predictors in the model.

1. Remove the predictor with highest $p-$value $> \alpha_0$ (most insignificant).
2. Refit the model, and repeat the above process.
3. Stop when all $p-$values $\leq \alpha_0$.
   (α_0 is often set to 15% or 20% which is higher than usual)

Forward elimination

1. Start with the intercept-only model.
2. For all predictors not in the model, check their p-value if being added to the model. Add the one with the lowest $p-$value $\leq \alpha_0$ (most significant).
3. Refit the model, and repeat the above process.
4. Stop when no more predictors can be added.

Pros and Cons of Testing-based procedures

- Main advantage: Computation cost is low.
- Due to the "one-at-a-time" nature of adding/dropping variables, this type of procedures does not compare all possible models. So it's possible to miss the "optimal" model.
- It's not clear how to choose α_0, the cut-off for p-values.

Criterion-based procedures

1. Score each model according to an information criteria
2. Use a searching algorithm to find the optimal model

Model selection criteria/scores often takes the following form:

$$\text{Training error} + \text{Complexity-penalty}$$

- In the context of linear regression models, complexity of a model increases with the number of predictor variables (i.e., p_γ).
- Why we do not use R^2 or RSS?

AIC/BIC

$$AIC : -2 \times loglik_\gamma + 2p_\gamma$$
$$BIC : -2 \times loglik_\gamma + \log(n)p_\gamma$$

where p_γ is the number of predictors included in model γ

For the linear regression model:

$$-2 \times loglik_\gamma = n \log \frac{RSS_\gamma}{n}$$

The lower the AIC/BIC the better. Note that when n is large, adding an additional predictor costs a lot more in BIC than AIC. So AIC tends to pick a bigger model than BIC.

Adjusted-R^2 for model γ

$$\begin{aligned} R_a^2 &= 1 - \frac{RSS/(n - p_\gamma - 1)}{TSS/(n-1)} \\ &= 1 - (1 - R^2)(\frac{n-1}{n - p_\gamma - 1}) \\ &= 1 - \frac{\hat{\sigma}_\gamma^2}{\hat{\sigma}_0^2} \end{aligned}$$

The higher the R_a^2 the better.

Mallow's C_p

$$C_p = \frac{RSS_\gamma}{\hat{\sigma}^2} + 2p_\gamma - n$$

where $\hat{\sigma}^2$ is the estimate of the error variance from the full model. Mallow's C_p behaves very similar to AIC.

- Leap and Bounds: return the global optimal solution among all possible models, but only feasible for less than 50 variables.
  - Find the p models with the smallest RSS amongst all models of the same size[1].
  - Then evaluate the score on the p models and report the optimal one.

[1] Note that step 1, we do not need to visit every model. For example, suppose we know that $RSS(X1, X2) < RSS(X3, X4, X5, X6)$; then we do not need to visit any size 2 or 3 sub-models of set $(X3, X4, X5, X6)$, which can be leaped over

Searching Algorithms

- Greedy algorithms: fast, but only return a local optimal solution (which might be good enough in practice).
  - Backward: start with the full model and sequentially delete predictors until the score does not improve.
  - Forward: start with the null model and sequentially add predictors until the score does not improve.
  - Stepwise: consider both deleting and adding one predictor at each stage.

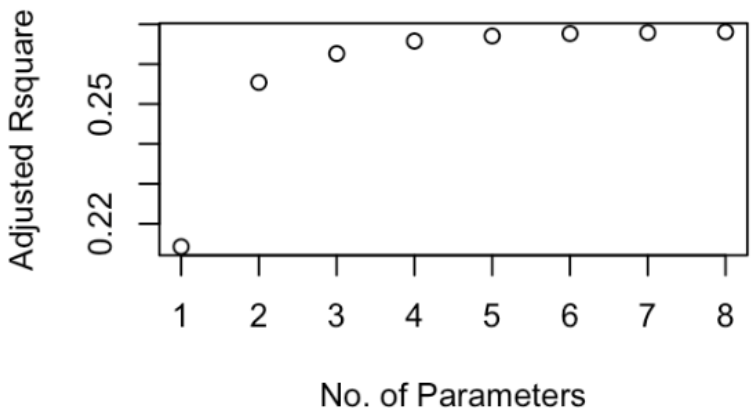
Bike Share Counts Data

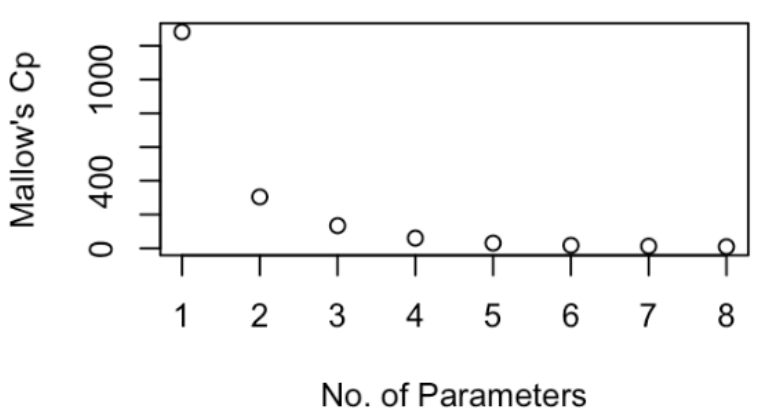
```
library(leaps)
b=regsubsets(cnt~., data = bikeshares)
rs = summary(b)
rs$which
```

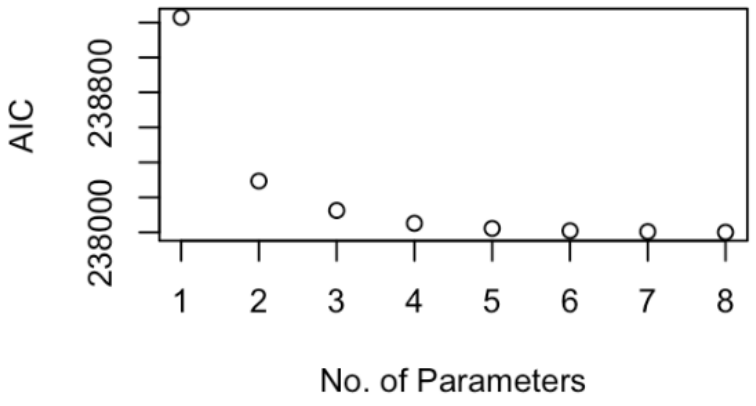
Leap and Bounds method

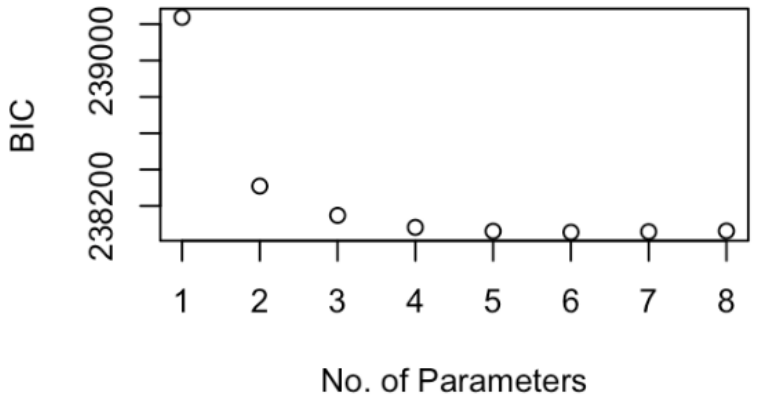
Use function *regsubsets* from library **leaps** to evaluate different scores for sub-sets of models up to size p (including the intercept).

Searching methods









Searching methods

Use function *step* from the **stats** library to apply searching algorithms based on the AIC (default) or BIC criteria ($k = \log(n)$). The option direction=both uses the Stepwise searching algorithm. You can also use the options direction=forward and direction=backward.

```
step(full.model, direction="both")
step(full.model, direction="both", k=log(n))
```