Linear Methods for Regression: Subset selection

-Statistical Machine Learning-

Lecturer: Darren Homrighausen, PhD

SUMMARY

The overall scheme is a three(four?)-fold process

1. Select a method suited to your task
2. Choose a risk estimation method that has the properties that you desire (e.g. end of previous slides)
3. Perform the necessary computations to minimize 2. constrained to be in the family of procedures in 1.
4. Show theoretically that your procedure has desirable properties

Summary

The overall scheme is a three(four?)-fold process

1. Select a method suited to your task
2. Choose a risk estimation method that has the properties that you desire (e.g. end of previous slides)
3. Perform the necessary computations to minimize 2. constrained to be in the family of procedures in 1.
4. Show theoretically that your procedure has desirable properties

Brief optimization and convexity detour

Optimization

An optimization problem (program) can be generally formulated as

$$\begin{aligned} \text{minimize } & F(x) && (1) \\ \text{subject to } & f_j(x) \leq 0 \text{ for } j = 1, \ldots, m && (2) \\ & h_k(x) = 0 \text{ for } k = 1, \ldots, q && (3) \end{aligned}$$

Here

$x = (x_1, \ldots, x_n)^\top$ are the parameters

$F : \mathbb{R}^n \to \mathbb{R}$ is the objective function

$f_j, h_k : \mathbb{R}^n \to \mathbb{R}$ are constraint functions

The optimal solution x^* is such that $F(x^*) \leq F(x)$ for any x^*, x that satisfies equations (2) and (3).

Convexity

The main dichotomy of optimization programs is convex vs. nonconvex

Generally speaking, a convex program is one in which the objective and contraint functions are all convex, that is $\forall t \in [0,1], \forall x \in D = \left(\bigcap_{j=1}^{m} \mathrm{dom}\ f_i\right) \cap \left(\bigcap_{k=1}^{q} \mathrm{dom}\ h_k\right) \cap (\mathrm{dom\ F})$, and $\forall f \in \{f_1, \ldots, f_m, h_1, \ldots, h_q, F\}$

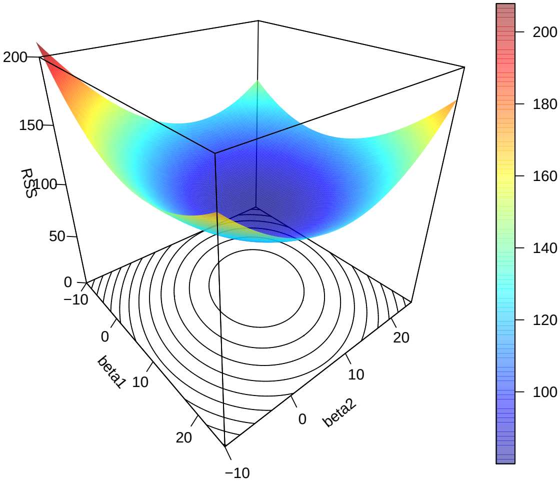
$$f(tx + (1-t)x') \leq tf(x) + (1-t)f(x')$$

This can be thought of (for smooth enough f)

$$f(x') \geq f(x) + (\nabla f|_x)^\top (x' - x)$$

Intuition: This means that the function values at a point x' are above the supporting hyperplane given by the tangent space at any point x

Convexity example



$||Y - \mathbb{X}\beta||_2^2$ for $p = 2$

Convexity

Methods for convex optimization programs are (roughly) always global and fast

For general nonconvex problems, we have to give up one of these:

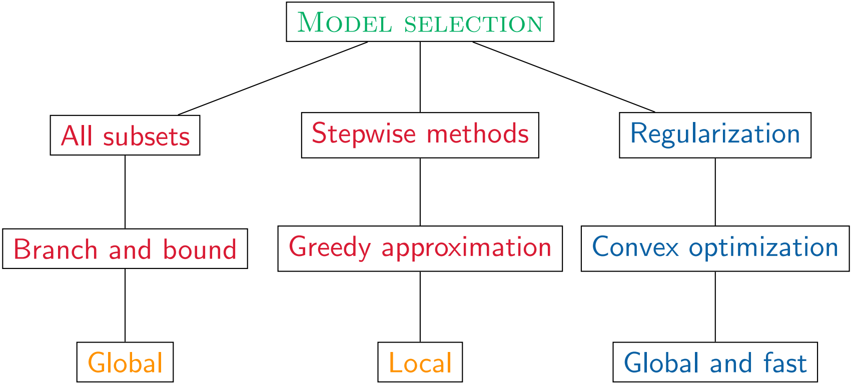
- Local optimization methods that are fast, but need not find global solution

  (So called greedy approximations)

- Global optimization methods that find global solutions, but are not always fast (indeed, are often slow)

  (Usually exhaustive search type approaches)

Model selection



Some comments:

Non convex programs

Can be seen as a convex relaxation of the nonconvex program giving all subsets[1]

[1] We'll return to this shortly

All Subsets Regression

First, identify all considered covariates and transformation and put them in the feature matrix $\mathbb{X} \in \mathbb{R}^{n \times p}$

Best subset selection algorithm: For $k = 1, \ldots, p$

1. Find $\hat{R}_{\mathrm{train}}$ for the $\binom{p}{k}$ models of size k
2. Save the model that minimizes $\hat{R}_{\mathrm{train}}$

Now, report the model that minimizes one of the risk estimates from the previous slide over these p models

All Subsets Regression in R

We can find with the function regsubsets in the package leaps (See code on website) The syntax and associated objects look like:

```
allsubsets.out = regsubsets(Y˜.,data=X,nvmax=pmax)
> summary(allsubsets.out)
[1] "which"  "req"  "rss"  "adjr2"  "cp"  "bic  "outmat"  "obj"
```

- The nvmax $=$ pmax controls the max size of models considered. The default is 8 and that is usually far too small.
- Now, we can pick among the pmax models that minimize $\hat{R}$ for a given model size using BIC or Cp

This can be done in some cases, though there is a problem

All Subsets Regression: A Big Problem (Literally)

If there are p predictors then there are 2^p possible models

(Without considering interactions or transformations)

In general, this is a nonconvex problem

If $p = 40$ (which is considered a small problem these days), then the number of possible models is

$$2^{40} \approx 1,099,512,000,000 \Rightarrow \text{More than 1 trillion!}$$

If $p = 265$, then the number of possible models is more than the number of atoms in the universe[2]

We must sift through the models in a computationally feasible way

[2] It is estimated there are 10^{80} atoms in the universe.

All Subsets Regression

The leaps package in R uses a technique known as branch and bound

(The statistical implementation is based on the paper Furnival and Wilson (1974))

It is a widely used tool for solving large scale NP-hard combinatorial optimization problems.

Note, however, that though it can speed up the optimization immensely, it cannot reduce the complexity of the problem

(Still exponential)

Branch and bound

Let $M = M_1 \cup \ldots \cup M_K$ be the set of all possible solutions and a partition comprised of branches, respectively.

(Statistically, we think of M as the set of all possible models.)

Suppose for objective function F we want to find

$$F_* := \max_{m \in M} F(m)$$

For each M_k, define

$$F_k := \max_{m \in M_k} F(m)$$

and let $\underline{F}_k, \overline{F}_k$ be a bracket such that

$$\underline{F}_k \leq F_k \leq \overline{F}_k$$

(Note that F_k is in general not explicitly constructed)

Then

$$\max_k \underline{F}_k := \underline{F} \leq F_*$$

Branch and bound

The main realization is that the branch M_k does not need to be explored if either of the following occur

i. Bound

$$\overline{F}_k \leq \underline{F}$$

ii. Optimality

$$\max_{m \in M_k} F(m) \text{ has been found}$$

Branch and bound

The two main questions remain:

1. How to choose the partition(s)?
2. How to form the bracket?

   (Note that to be helpful, the bracket must be easy to compute)

These are very case specific. Let's return to model selection

Branch and bound for model selection

Let's suppose we set[3]

$$F(m) = n\log(\hat{R}_{\text{train}}(\hat{\beta}_m)) + 2|m|$$

For a set of models M_k, let

$m_{k,inf}$ be the largest model contained[4] in every model in M_k

$m_{k,sup}$ be a smallest model that contains every model in M_k

[3] Note: we are trying to minimize F, not maximize

[4] This does not have to be in M_k

Branch and bound for model selection

Example: Let $x_1, \ldots, x_5$ be covariates

$$M = \cup_{k=1}^{3} M_k,$$

where

$$\begin{aligned} M_1 &= \{\{x_1, x_3\}, \{x_2\}\}, \\ M_2 &= \{\{x_2, x_3, x_4\}, \{x_3, x_4\}\}, \\ M_3 &= \{\{x_3, x_5\}, \{x_3\}\}, \end{aligned}$$

Branch and bound for model selection

Example: Let $x_1, \ldots, x_5$ be covariates

$$M = \cup_{k=1}^{3} M_k,$$

where

$$\begin{aligned} M_1 &= \{\{x_1, x_3\}, \{x_2\}\}, \\ M_2 &= \{\{x_2, x_3, x_4\}, \{x_3, x_4\}\}, \\ M_3 &= \{\{x_3, x_5\}, \{x_3\}\}, \end{aligned}$$

$$\begin{aligned} m_{2,inf} &= \{x_3, x_4\} \\ m_{2,sup} &= \{x_2, x_3, x_4\} \end{aligned}$$

Branch and bound for model selection

Reminder:

For the M_k, let

$m_{k,inf}$ be the largest model contained in every model in M_k

$m_{k,sup}$ be a smallest model that contains every model in M_k

Then:

$$\forall m \in M_k$$

$$F(m) \geq n \log(\hat{R}_{\text{train}}(\hat{\beta}_{m_{k,\text{sup}}})) + 2|m_{k,\text{inf}}| = L_k$$

$$F(m) \leq n \log(\hat{R}_{\text{train}}(\hat{\beta}_{m_{k,\text{inf}}})) + 2|m_{k,\text{sup}}| = U_k$$

(We don't actually need U_k, though)

Branch and bound for model selection: An algorithm

1. Define a global variable $b = F(m)$ for any $m \in M$

   (As an aside, every time $F(m)$ is computed, update b if $F(m) < b$)
2. Partition $M = \{M_1, \ldots, M_K\}$
3. For each k, if $L_k > b$, eliminate the branch M_k
4. Gather each remaining M_k and set union equal to M
5. Else, recurse and return to 2.

Greedy approximations

Forward stepwise selection

In the likely event that 2^p is too large to be searched over exhaustively, a common greedy approximation is the following

Let $\hat{R}$ be any risk estimate

1. Find $\hat{R}(\emptyset)$: That is, the intercept only model
2. Search over all p single feature models, computing $\hat{R}$ for each one. Say including x_j minimizes $\hat{R}$ with a value $\hat{R}(x_j)$. If $\hat{R}(x_j) < \hat{R}(\emptyset)$, add x_j to the model and continue. Otherwise terminate
3. Now search over all $p - 1$ models that contain x_j and find the $x_{j'}$ that minimizes $\hat{R}$. If $\hat{R}(x_j, x_{j'}) < \hat{R}(x_j)$, add $x_{j'}$ to the model and continue. Otherwise terminate
4. $\cdots$

Forward stepwise selection

```
regsubsets(Y~.,data=X,nvmax=pmax,method='forward')
```

Pros:

- This approach can be used effectively in either the Big Data or High Dimensional regimes
- It tends to produce sensible answers that are not too different from all-subsets

Cons:

- Can get trapped in a poor local minimum

General stepwise selection

This algorithm can can adapted to..

- start with the full model and stepwise remove covariates. This is known as backward stepwise selection

  ```
  regsubsets(Y~.,data=X,nvmax=pmax,method='backward')
  ```

  (useful if the full model isn't too large and a superset of the important covariates is desired)
- consider both adding and removing covariates at each step. This is known as stepwise selection

  ```
  regsubsets(Y~.,data=X,nvmax=pmax,method='seqrep')
  ```

Important comments

After using any of these model selection approaches, we produce estimates $\hat{\beta}$ and predictions $\hat{Y} = \hat{\beta}^\top X_{\text{select}}$ where X_{select} includes only the selected features

This can be interpreted as these covariates are most important for predicting Y from the features included in $X \in \mathbb{R}^p$

(The usual caveats apply: linearity (correlation), there are surely some important coefficients left out/unimportant ones included)

If we run out $=$ lm(Y$\sim \mathbb{X}_{\text{select}}$), then summary(out) will produce the usual significance tests: these are not valid after model selection

Important comments

- If we want to be sure to include all the important covariates, then we can use AIC/Cp + backward stepwise selection
- If we want to be sure to only include important covariates, then we can use BIC + forward stepwise selection
- If we want to do predictions, use AIC/Cp, but it isn't clear what method is the best

(See website for example code for doing model selection in R)