Lecturer: Prof. Homrighausen Scribe: Soo-Young Kim

# 1 Brief optimization and convexity detour

## 1.1 Optimization

An optimization problem can be generally formulated as

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

where

$x = (x_1, \cdots, x_n)^T$ 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.

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

## 1.2 Convexity

The main dichotomy of optimization programs is convex vs. nonconvex. A convex program is one in which the objective and contraint functions are all convex. The function $f$ is called convex if:

$$f(tx + (1-t)x') \leq tf(x) + (1-t)f(x') \qquad \forall x, x' \in D, \forall t \in [0,1]. \tag{4}$$

Methods for convex optimization programs are (roughly) always global and fast, but not for nonconvex problems. For nonconvex problems,

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

# 2 Model selection

## 2.1 All subsets regression

There is a problem for the all subsets regression. In general, it is a nonconvex problem. If there are predictors, then there are $2^p$ possible models without considering interactions. Branch and bound, proposed by Furnival and Wilson [1], is a widely used tool for solving large scale NP-hard combination problems, but it cannot reduce the complexity of the problem. The function `regsubsets` from `leaps` package is available in `R` for the model selection.

## 2.2 Branch and bound

Let $M = M_1 \cup \cdots \cup M_K$ 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. Let $F$ be an objective function, then 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

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

Then

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

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

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

The two main questions remain:

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

## 2.3 Branch and bound for model selection

We want to minimize

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

For a set of models $M_k$, let

$m_{k,inf}$ be the largest model contained[1] in every model in $M_k$

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

then, $\forall m \in M_k$

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

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

[1] This does not have to be in $M_k$

### 2.4 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.

## 3 Greedy approximations

### 3.1 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 can be used effectively to produce sensible answers in either big data or high dimensional regimes, but it might get trapped in a poor local minimum.

### 3.2 General stepwise selection

- Backward stepwise selection: it starts with the full model and stepwise remove covariates.
- Stepwise selection: this consider both adding and removing covariates at each step.
- 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

## References

[1] Furnival, G. M. and Wilson, Jr., R. W. (2000). Regressions by leaps and bounds. *Technometrics*, 42(1):69–79.