Need for Heuristics

Combinatorial Optimization ⇒ Finding global maximum is not possible.

2^{100} (practical time limit)

- Necessary to abandon algorithms that are guaranteed to find the global maximum under suitable conditions but will never succeed within a practical time limit. Better to use algorithms that can find a good local maximum within tolerable time.

  Computational Time Good Solution

  Trade-off

- Heuristics: Intend to find a globablly competitive candidate solution with an explicit trade of global optimality for speed.
  - iterative improvement of a current candidate solution, and
  - limitation of the search to a local neighborhood at any particular iteration.

Example : Local Search.

Example: Variable Selection in Regression

Sales Traverse Problem : Industrial Engineering.

Stepwise Selection
Lasso / Ridge

- Consider a multiple linear regression problem with p potential predictor variables and try to select a suitable model.
- Given a dependent variable Y and a set of candidate predictors $x_1, \cdots, x_p$, we must find the best model of the form

$$Y = \beta_0 + \sum_{j=1}^{s} \beta_{i_j} x_{i_j} + \epsilon,$$

where $\{i_1, \cdots, i_s\}$ is a subset of $\{1, \cdots, p\}$ and ϵ denote a random error.

with intercept.

- The variable selection problem requires an optimization over a space of 2^{p+1} possible models.

Find the best $2^{p+1} - 1$.

(model within.)

Example: Variable Selection in Regression

- Two ways to find the best model
  - Use the Akaike information criterion. Seek to find the subset of predictors that minimizes the fitted model AIC,

    $$\mathrm{AIC} = N \log\{\mathrm{RSS}/N\} + 2(s+2),$$

    where N is the sample size, s is the number of predictors in the model, and RSS is the sum of squared residuals
  - Use Bayesian regression with the normal-gamma conjugate class of priors

    $$\beta \sim N(\mu, \sigma^2 V) \quad \text{and} \quad \nu\lambda/\sigma^2 \sim \chi^2_\nu$$

    and find the subset of predictors corresponding to the model that maximizes the posterior model probability.

Basic Local Search

Regression $x_1, x_2, x_3, x_4, x_5, x_6, \ldots, x_p$

Current model $1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0, \ldots, 1$

(Include 1, Not Include 0) (eg) $1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0, \ldots, 1$

$1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0, \ldots, 1$

both of them are neighborhood.

0 set : the set of variables not included in model

1 set : the set of variable included in model

- It is an iterative procedure that updates a current candidate solution $\theta^{(t)}$ at iteration t to $\theta^{(t+1)}$.
- One or more possible moves (updates) are identified from a neighborhood of $\theta^{(t)}$, $\mathcal{N}(\theta^{(t)})$. (1) Choose one candidate from a neighborhood
- A neighborhood of the current candidate solution, $\mathcal{N}(\theta^{(t)})$, contains candidate solutions that are near $\theta^{(t)}$.

(2) Evaluate candidate.

(3) If the candidate model is better than current one, we update the model.

Basic Local Search

$\underline{X_1, \ldots, X_p}$ * Neighborhood : flip one variable

sample(1:p, 1, without replacement) ↳ choose $\underbrace{\text{one neighborhood}}_{1}$ among $\underbrace{\text{possible neighborhoods}}_{p}$

prob = equal

- The advantage of local search over global search is that only a tiny portion of Θ need be searched at any iteration, and large portions of θ may never be examined.

  ↳ advantage : computational speed.

  disadvantage : hard to reach global (good local) optimum.

- The disadvantage is that the search is likely to terminate at an uncompetitive local maximum.
- If the neighborhood is defined by allowing as many as k changes to the current candidate solution in order to produce the next candidate, then it is a k-neighborhood.

  → possible candidate : pC_k

  sample(1:p, k, w/o replacement)

random current state $M^{(0)} \downarrow M^{(1)} \downarrow M^{(2)} \leftarrow \cdots M^{(m)}$

m : total number of iteration (pre-fixed)

Neighborhood : flip two variables. → 2 neighborhood

possible pC_2

sample(1:p, 2, w/o replacement, prob=equal)

Local Search - Ascent Algorithm

Find all neighborhood from current candidate.
=> Evaluate them. => Find the best candidate.

- *Steepest Ascent*: An obvious strategy at each iteration is to choose the best among all candidates in the current neighborhood.

From all neighborhoods, choose one candidate. => Evaluate this candidate

- *Random Ascent*: To speed performance, one might instead select the first randomly chosen neighbor for which the objective function exceeds its previous value.
- If k-neighborhoods are used for a steepest ascent algorithm, the solution is said to *k-optimal*.
- Although the ascent is not the steepest possible within $\mathcal{N}(\theta^{(t)})$, any local search algorithm that choose $\theta^{(t+1)}$ uphill from $\theta^{(t)}$ is an *ascent algorithm*.

Local Search - Local Optimal Values

easy updating : a narrow focus enabling quick moves.
avoid local trap : sometimes it require a big jump

- The sequential selection of steps that are optimal in small neighborhoods, disregarding the global problem, is reminiscent of a *greedy algorithm*. : Search all possible cases in an algorithm
- Wise selection of a new candidate solution from a neighborhood of the current candidate must balance the need for a narrow focus enabling quick moves against the need to find a globally competitive solution.
- To avoid entrapment in poor local maxima, it might be reasonable to eschew some of the best neighbors of $\theta^{(t)}$ in favor of a direction whose rewards are later realized.

Local Search - Variable Depth Local Search

1, --- , p
O O $p-2$ nC_k $k=3$

- When *k* is greater than 1 or 2, searching within the current neighborhood for a *k*-change steepest ascent move can be difficult because the size of the neighborhood increases rapidly with *k*.
  exponential.
- For larger *k*, it can be useful to break the *k*-change up into smaller parts, sequentially selecting the best candidate solutions in smaller neighborhoods.
- This *variable-depth local search* permit a potentially better step away from the current candidate solution, even though it will not likely be optimal within *k*-neighborhood.
  n chain => n results.
- *Random starts local search* run a simple ascent algorithm repeatedly with a large number of starting points.

Example: Baseball Salaries

- Application of the random starts local search method to a regression model selection problem.
- There are 27 baseball performance statistics, which were collected for 337 players in 1991 and players' 1992 salaries may be related to these variables. Use the log of the salary variable as the response variable.
- Find the best subset of predictors to predict log salary using a log linear regression model. There are $2^{27} = 134,217,728$ possible models.

Example: Baseball Salaries

HW1 Due : 10/7

Implementation (HW2)

- The application of a random starts local search algorithm to minimize the AIC w.r.t. regression variable selection.
- The problem can be posed as maximizing the negative of the AIC, thus preserving our preference for uphill search.
- Neighborhoods were limited to 1-changes generated from the current model by either adding or deleting one predictor.
- Search was started from 5 randomly selected subsets of predictors and 14 additional steps were allocated to each start.