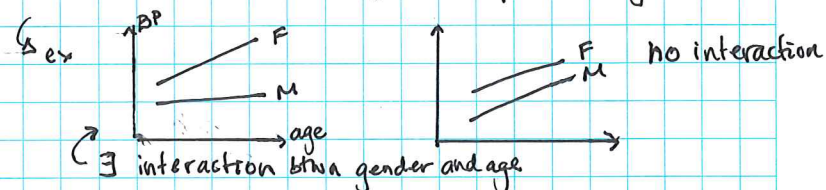
- Recall Generalized Additive Models (GAMs): (can use GAMs for regression and for classification)

$$y_i = \beta_0 + \sum_{j=1}^{p} f_j(x_{ij}) + \varepsilon_i$$

- non-linearity makes GAM generalizable
- GAMs do not allow interactions, ie: where X_j and X_k interact in their "effects" (non-causal) on Y if the "effect" of X_j on Y depends only on the value of X_k
  - ex: BP, F, M, age
  - ∃ interaction btwn gender and age
  - no interaction
  - ⇒ simplifies the model.

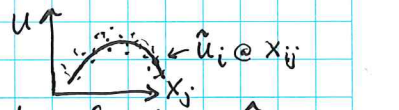


- (RCode11)
  - line 152 :: ns(·) := natural spline
    - education is an ordinal variable in the model and has dummy variables ∀ category (ex. HighSchool, College, ...)
    - plots:
      - $ns^{(yr)}$ vs yr : trend is from inflation and ~~stocks~~
      - $ns^{(a)}$ vs age : increased salaries over time and retirement (pensions) starting
  - line 162: backfitting for GAM
  - line 177: • we go with model m2 because it has the smallest Pr(>F)
    • a good model has: simple structure, (non linearity), low Pr(>F)

Backfitting (1985 Breiman & Friedman)

(comparable to (cf) Gauss-Seidel method for solving systems of linear equations)

- Let $S_j : (u_1, \dots, u_n)^T \to (\hat{u}_1, \dots, \hat{u}_n)$ denote a smoother of $u_1, \dots, u_n$ against $x_{1j}, \dots, x_{nj}$



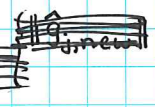
Procedure:

1. Initialize: est. mean $\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} y_i$, set all predictor functions $\hat{g}_j = 0$, $j = 1, \dots, p$
2. Do until convergence:
   - ~~Predictor~~ ∀ $j^{th} = 1, \dots, p$ predictor:
     - (a) fit $\hat{g}_j = S_j\left(\underbrace{y - \hat{\mu}\mathbf{1}}_{\text{center } y} - \underbrace{\sum_{k \neq j} \hat{g}_k}_{\text{sum of vectors} \in \mathbb{R}^n}\right)$. For $j = 1$, $\sum_{k \neq 1} \hat{g}_k = 0$.
     - (b) Normalize: $\hat{g}_j \leftarrow \hat{g}_j - \frac{1}{n}\sum_{i=1}^{n} \hat{g}_j(x_{ij})$,

       where $\hat{g}_j = (\hat{g}_j(x_{1j}), \dots, \hat{g}_j(x_{nj}))^T$, ie $(\hat{g}_j(x_{ij}))$. $\hat{g}_j$ is the contribution of the j^{th} variable

       $Y = (y_1, \dots, y_n)^T$

       ie: we fix all variables $k \neq j$ and smooth on the residuals.
     - (c) Measure convergence (if we have) by, eg. the relative change of $\hat{g}_j$ vector.

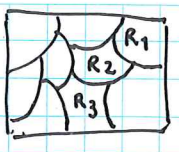


If $\max_j \left\{ \frac{\|\hat{g}_{j,new} - \hat{g}_{j,old}\|_2}{\|\hat{g}_{j,old}\|_2} \right\} \leq$ tolerance, then we have reached convergence.

⇒ other side pg. 2.

Computational Statistics Lecture — Thu 24 May 2018

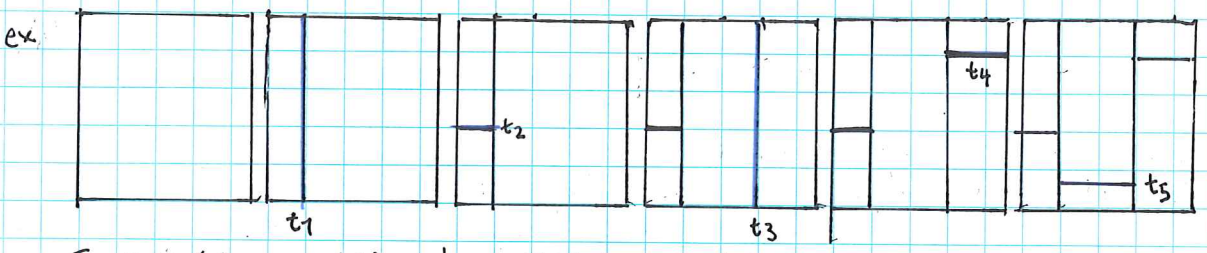
- **Trees** (ISLR chapter 8). **CART**.

ex 

- CART: Classification And Regression Trees (1984 Breiman).
  - The underlying model: $g_{tree}(x) = \sum_{r=1}^{M} \beta_r \mathbf{1}_{\{x \in R_r\}}$, where $x \in \mathbb{R}^p$. g_{tree} is piecewise ~~constant~~ constant
  - $P = \{R_1, \dots, R_M\}$ is a partition of $\mathbb{R}^p$ (ie union of disjoint sets $\bigsqcup_{r=1}^{M} R_r = \mathbb{R}^p$) — P covers wholespace.
  - If the partition is given, then things are easy. (

$$\hat{\beta}_j = \frac{\sum_{i=1} y_i \mathbf{1}_{\{x_i \in R_j\}}}{\sum_{i=1} \mathbf{1}_{\{x_i \in R_j\}}} \equiv \text{average } Y \text{ value among observations in } R_j \text{ (subspace of } \mathbb{R}^p)$$

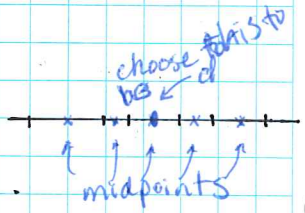
  * How do we find a good partition? (See Fig 8.1, 2 (ISLR) in Slides)

ex 

  - Fig. 8.3 (lower right): shows that the model is piecewise constant function
  - If we grow the tree too deeply, then we overfit.

Estimation (of the best partition).

1. Start with $M=1$. Then $R_1 = \mathbb{R}^p$, $P = \{R_1\}$.
2. Refine the partition $R_1 = R_{1left} \sqcup R_{1right}$, (disjoint sets)
   where $R_{left} = \mathbb{R} \times \mathbb{R} \times \dots \times \mathbb{R} \times (-\infty, d] \times \mathbb{R} \times \dots \times \mathbb{R}$
   $R_{right} = \mathbb{R} \times \mathbb{R} \times \dots \times \mathbb{R} \times (d, \infty) \times \mathbb{R} \times \dots \times \mathbb{R}$
   (d at the jth axis.)



   This involves choosing an axis j (ie a predictor X_j) and a split point $d \in$ {midpoints between X_j's}.
   - Choose them such that the (-log-likelihood) is reduced the most; in certain cases, this is ~~just~~ the same as reducing the RSS the most.
3. Refine the current partition by refining one of the partition cells.
4. Iterate until we have a large tree.
5. Then, prune the tree to get a simpler model using cross-validation.

cost complexity of pruning $= \min_{T \subseteq T_0} \{ err(T) + \alpha |T| \}$

- T_0: full tree
- $err(T)$: we want low error
- $\alpha |T|$: penalty term for # of nodes ie. # of parameters so that model is not unnecessarily complex.

(RCode13)
- line 19: "high" is a binary variable.

// End of lecture //

Trees

$$y_i = \sum_{r=1}^{M} \beta_r \mathbb{1}_{\{X_i \in R_r\}} + \varepsilon_i$$

- Difficult part is getting partition.
- We limit ourselves to R_r being boxes.
- Find partition using greedy approach. → recursive binary splitting
  ↳ Pick the partition that reduces (-(log-likelihood)) the most

Pruning: $\min_{T \subseteq T_0} \; err(T) + \alpha |T|$

↑ $-\log like(T)$: misclassification rate

ex. 20 observations : 16 yes, 4 no.

- Put them in one leaf node: $n_y \log(\hat{p}_y) + n_n \log(\hat{p}_n)$ ("yes", "no")

$$= 16 \log\left(\tfrac{16}{20}\right) + 4 \log\left(\tfrac{4}{20}\right).$$

- Split them over two leaf nodes → $-\log$ likelihood: 10.008

9Y → $9\log(1) + 0\log(0)$
7Y, 4N → $7\log(7/11) + 4\log(4/11)$
→ $-\log like$: 7.23.

(R code 13)
- line 55 : "dev" ≡ deviance ≡ # of misclassifications.
  "size" ≡ # of nodes (remaining) in tree after each pruning step.
- line 71 : "prune.{##}" → best tree with ≥ {##} nodes

Bagging : "Bootstrap Aggregating"

Procedure: Consider a base procedure, ex. a tree, that gives us some estimated function $\hat{g}$: ex. $\hat{g}$: $\mathbb{R}^p \to \mathbb{R}$ (prediction)
$\mathbb{R}^p \to [0,1]$ (classification)

Algorithm: (1) Generate a bootstrap sample $(X_1^*, Y_1^*), \ldots, (X_n^*, Y_n^*)$ and compute the corresponding estimator $\hat{g}^*(\cdot)$ by running our base procedure on the bootstrap sample.

(2) Repeat this B times, yielding $\hat{g}^{*1}(\cdot), \ldots, \hat{g}^{*B}(\cdot)$
(store these tree objects s.t. we can predict @ any point.)

(3). Aggregate the bootstrap estimates by averaging all the bootstrap trees.
- The bagged estimator $\hat{g}_{bag}(\cdot) = \frac{1}{B} \sum_{b=1}^{B} \hat{g}^{*b}(\cdot)$ is an approximation of $E^* \hat{g}^*(\cdot)$, which is exact if $B = \infty$.

$\hat{g}_{bag}(\cdot)$: aggregation reduces variance but doesn't incur much bias, ↳ individual trees have low bias, high variance
bag has low bias, low variance

// End of Lecture //

Computational Statistics Section

Fri 25 May 2018

- Ref last week's series, problem 1: (correction) hint: justify that $y_i^{(-i)}$ coincides with $\hat{z}_1$, where $\hat{z}$ is the fitted value obtained from regression of z on $x_1, \dots, x_p$, where z is defined as:
  $z_j = y_j \quad \forall j \neq i$
  $z_i = y_i^{(-i)}$
- $H_0 : y = \beta_0 + \varepsilon$ (global null)
  $H_1 : y = \beta_0 + \sum_j \beta_j x_j + \varepsilon$
- T_1 : $H_0' : \beta_1 = 0$ in the model $y = \beta_0 + \sum_j \beta_j x_j + \varepsilon$
- If H_0 is true, then $\rightarrow H_0'$
- $p_1 = P_0(|x| > |T_1|)$
  $p_1 = P_0(x^2 > T_1^2) = 1 - F_0(T_1^2)$
  where F_0 is the CDF of T_1^2 under the null
  - for a newly generated variable x
  - p_1 changes, has a distribution
- Goal: Show that for $\alpha \in [0,1]$, (Prob) $P(p_1 \leq \alpha \mid H_0) = \alpha \quad \forall \alpha \in [0,1]$
  ↑ sample from null under
- $p_1 \sim \text{Unif}(0,1) \Rightarrow 1 - p_1 \sim \text{Unif}(0,1)$

$$P(1-p_1 \leq \alpha \mid H_0) = P(F_0(T_1^2) \leq \alpha \mid H_0) = P(T_1^2 \leq F_0^{-1}(\alpha) \mid H_0)$$

($T_1^2 \sim F_0$ under null) ↑ property of monotonicity of F_0 and F_0^{-1}

"What is the probability that a random variable will be < quantile of distribution from which we are asking the problem?".

(T-test $\Rightarrow$ continuous distribution)

So, $P(T_1^2 \leq F_0^{-1}(\alpha) \mid H_0) = \alpha$ under continuity of F_0.

- Assuming under the null, there is an equal chance of getting a very low or ~~and~~ getting a very high pval.

  ie. Under the null, the p value is (completely) random, so ~~it makes no~~ we can't use ~~it to describe model~~ it for anything really....

// End of Section //