Supervised Statistical Learning

Fabrizio Cipollini
DiSIA *G. Parenti*, Università di Firenze

MaBiDa – Florence, May-July, 2017

References

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2014). *An Introduction to Statistical Learning: With Applications in R*. Springer Publishing Company, Incorporated.

- **Background**: Chapters 3 and 4 (and `R`)
- **This course**: Chapters 2, 5 (partially), 6, 7, 8
- **Additional** material provided during the course

Framework

- Data: one dependent variable, p independent variables

$$(y_i, \underbrace{x_{i,1}, \dots, x_{i,p}}_{\boldsymbol{x}_i'}) \qquad i = 1, \dots, n$$

- Models:
  1. **Quantitative** $y \rightarrow$ Regression

$$y = \mu(\boldsymbol{x}) + \varepsilon \quad \text{s.t.} \quad \varepsilon \perp \boldsymbol{x},\ E(\varepsilon) = 0$$
$$y|\boldsymbol{x} \sim \left[\mu(\boldsymbol{x}), \sigma^2\right] \qquad \mu(\boldsymbol{x}) = E(y|\boldsymbol{x}) = f(\boldsymbol{x})$$

  2. **Categorical** $(0/1)$ $y \rightarrow$ Classification

$$y|\boldsymbol{x} \sim Be(\mu(\boldsymbol{x})) \qquad \mu(\boldsymbol{x}) = E(y|\boldsymbol{x}) = \frac{e^{f(\boldsymbol{x})}}{1 + e^{f(\boldsymbol{x})}}$$

$f(.)$ is **unknown**!

Challenge

$\boxed{f(\cdot)\ \textbf{unknown} \rightarrow \widehat{f}(\cdot)\ \textbf{estimated}\ \text{using data}}$

A challenge involving selection of

- Variables: $f(\cdot)$ may depend on a **subset** of the independent variables in the data
  (in principle, it could also depend on **omitted** variables!)
- Shape: For a given set of independent variables, each x_j may give a **linear** or a **non-linear** ($\ldots$) contribution to $f(\cdot)$
- "Complexity": For a given set of independent variables, we can have **interactions** among the x_j's into $f(\cdot)$
- All such aspects are **interconnected**!

Statistics vs Statistical Learning

Aspect	**Statistics** view	**Statistical Learning** view
Slang	Conditional analysis In/out-of sample	Supervised learning Training/Test sets
Objective	Inference	Prediction
Focus	Model interpretation (parameters)	Prediction accuracy (error measures)
$\widehat{f}$	We need to know it	May be treated as black-box
Flexibility	Low	High

- Tools and methods are similar
- This course: Statistical Learning view

Definition and Decomposition

- Consider quantitative y: $y = f(\boldsymbol{x}) + \varepsilon$, $\varepsilon \perp \boldsymbol{x}$ and $E(\varepsilon) = 0$.
- **Mean Squared Error**:

$$MSE := E[(y - \widehat{f})^2] \underset{-/+f}{=} \underbrace{V(\varepsilon)}_{\text{Irreducible}} + \underbrace{E[(f - \widehat{f})^2]}_{\text{Reducible}}$$

$$\underset{-/+E(\widehat{f})}{=} \sigma^2 + bias^2(\widehat{f}) + V(\widehat{f})$$

- $V(\varepsilon) = \sigma^2 =$ "Irreducible error"
- $bias(\widehat{f}) = E(\widehat{f}) - f =$ "Error induced in approximating a (possibly) complex f by a (relatively) simple model"
- $V(\widehat{f}) =$ "How much $\widehat{f}$ would change if estimated using different data"
- **Training MSE** (MSE_{tr}) vs **Test MSE** (MSE_{ts})
- How to gain a better feeling? `MBD2016-MSE-20160503.R`

What we learn (theoretical point of view)

1. **Interpretation** of the MSE and its components (“Irreducible Error”, Variance, Bias^2)
2. $\widehat{f}(\cdot)$ **correctly specified** (it includes all “terms” in $f(\cdot)$) $\Longrightarrow$ Bias $= 0$
3. Model **complexity** $\uparrow \Longrightarrow$ Bias$^2 \downarrow$ and Variance $\uparrow$ $\Longrightarrow$ **Bias/Variance trade-off.** An easy compromize?
4. **No**:
   - In simulations we know $f(\cdot)$ and we have many replications, so it is possible to compute Bias and Variance
   - In practice we cannot! (We have only one Training dataset)

What we learn (practical point of view)

5. We are interested in a small MSE_{ts} but we manage **one Training dataset**, guided by an estimate of the MSE_{tr}:

$$\widehat{MSE}_{tr} = \frac{1}{n} \underbrace{\sum_{i=1}^{n} [y_i - \widehat{f}(\boldsymbol{x}_i)]^2}_{SSR}$$

$\widehat{MSE}_{tr}$ (SSR) can be reduced over and over, by increasing model complexity…but if we exaggerate, this increases MSE_{ts}!

Thus, we need a compromise:

Fit **training data** as better as possible (low SSR),
but …
don't go too far with **model complexity**!

Assembling Ingredients

- Recall our challenge:

$$\boxed{\text{Estimate the unknown } f(\boldsymbol{x}) \text{ using data}}$$

- To avoid indigestion, we split this lunch in three dishes:
  1. Variable Selection in a linear world $\rightarrow$ Chapter 6
  2. Shape Selection an additive world $\rightarrow$ Chapter 7
  3. “Complexity” Selection $\rightarrow$ Chapter 8
- Always the target is the smallest MSE_{ts}, but we pursue this goal following a compromise strategy:
  Minimize *SSR* penalizing excessive model complexity

$$\boxed{\min PSSR = \min(SSR + penalty(complexity))}$$

- In practice,

$$\boxed{\text{Penalized LS!}}$$

Introduction

- We assume a **linear** $f(\boldsymbol{x})$:

$$f(\boldsymbol{x}) = \beta_0 + \beta_1 x_1 \ldots + \beta_p x_p$$

  (although we use the same symbol, here p can be lower that the number of variables in the data)
- Reasons for assuming **linearity**:
  - Can be a **good approximation** of the true $f(\boldsymbol{x})$ (related to first order Taylor approximation of a math function)
  - We know what **linear** means while **non-linear** does not have a precise identity
  - By far more **simple** then more general approaches (think to OLS and the interpretation of coefficients)
  - Surprisingly **competitive** against more general approaches in many applications

Different flavors

- Subset selection:
  - **Idea**: Fit a lot of models based of different subsets of variables; then use some criterion to choose the best one
  - **Variants**:
    - **Best subset**
    - **Stepwise**: forward, backward, hybrid
- Shrinkage (regularization) methods:
  - **Idea**: Use all p independent variables, but penalize complexity by shrinking their coefficients toward zero $\rightarrow$ Reduce Variance paying some price in terms of Bias^2
  - **Variants**: (differing in the kind of penalty)
    - **Ridge** regression
    - **Lasso** regression
    - **Elastic Nets**
  - They work also with $p > n$!

Subset Selection: Best subset

- Algorithm:
  1. Start from the ***null model*** (no independent variables) $\mathcal{M}_0$
  2. For $k = 1, \dots, p$
     - 2.1 Fit all models with exactly k predictors
     - 2.2 Select the best one and call it $\mathcal{M}_k$
  3. **Select the best** model among $\mathcal{M}_k : k = 1, \dots, p$
- It requires $p < n$
- $\#$ models $= 2^p$ ($p = 20 \rightarrow \#$ models > 1000000)

Subset Selection: Forward Stepwise

- Algorithm:
  1. Start from the ***null model*** (no independent variables) $\mathcal{M}_0$
  2. For $k = 0, \dots, p - 1$
     - 2.1 Fit all $p - k$ models increasing $\mathcal{M}_k$ by 1 predictor
     - 2.2 **Select the best** one and call it $\mathcal{M}_{k+1}$
  3. Select the best model among $\mathcal{M}_k : k = 1, \dots, p$
- It can be used also when $p > n$
- $\#$ models $= p(p+1)/2 + 1$ ($p = 20 \rightarrow \#$ models $= 211$)

Subset Selection: Backward Stepwise

- Algorithm:
  1. Start from the ***full model*** (all independent variables) $\mathcal{M}_p$
  2. For $k = p, \ldots, 1$
     - 2.1 Fit all k models decreasing $\mathcal{M}_k$ by 1 predictor
     - 2.2 Select the best one and call it $\mathcal{M}_{k-1}$
  3. **Select the best** model among $\mathcal{M}_k : k = 1, \ldots, p$
- It requires $p < n$
- $\#$ models $= p(p+1)/2 + 1$ ($p = 20 \rightarrow \#$ models $= 211$)

Subset Selection: Hybrid Stepwise

- Algorithm: intermediate between forward and backward
- Idea:
  - In a forward based algorithm: after adding variables it may go back removing variable not improving fit
  - In a backward based algorithm: after removing variables it may go forward adding variables improving fit

Subset Selection: Selecting the best

- Minimize **Akaike** or **Bayesian** Information Criteria

$$AIC = SSR + 2k\widehat{\sigma}_*^2$$

$$BIC = SSR + \ln(n)k\widehat{\sigma}_*^2$$

  - $k = \#$parameters
  - $\widehat{\sigma}_*^2 =$ estimate of σ^2 based on a "large" (?) model
- Comments:
  - Structure: $SSR + penalty(k)$
  - Formulas proportional to those in the book (pp. 211-212)
  - Further criteria in the book (not in `step()`)
- Commonly used versions are different (cf `step()`):

$AIC = n\ln(SSR) + 2k + const$

$BIC = n\ln(SSR) + \ln(n)k + const \qquad const = n(\ln(2\pi/n) + 1)$

Subset Selection: Let’s Eat!

- **Appliances Energy Consumption** data: energy use of appliances in a low energy building
- **Dependent variable**: energy use of appliances
- **Independent variables**:
  - temperature and humidity measures from sensors of different building spaces
  - weather from a nearby airport station
  - recorded energy use of lighting fixtures
- $n = 19737$ vs $p = 33$ predictors
- `MBD2017-Energy-20170513.R`

Shrinkage Methods: Ridge Regression

- Estimate parameters **minimizing**

$$\boxed{SSR + \underbrace{\lambda \frac{1}{2} \sum_{j=1}^{p} \beta_j^2}_{penalty}}$$

- $\lambda \geq 0$ is the penalty/shrinkage/regularization parameter
- $\lambda = 0$ ($\to$ OLS) vs $\lambda = \infty$ ($\to \widehat{\beta}_1 = \ldots = \widehat{\beta}_p = 0$)
- Further comments on: method idea, L_2-norm, $1/2$ factor
- **Not** *scale equivariant*: standardize predictors!
- Large gain when OLS have large variance: large p; highly correlated predictors
- **Very fast**: computations required to estimate parameters for **all** λ's $\approx$ OLS computations
- Let's try: `MBD2017-Energy-20170513.R`

Ridge Regression: How to Choose λ?

- Use Cross Validation (CV, chapter 5), an approach that can be applied to almost any statistical learning method
- *k*-fold CV:
  - Randomly divide data in k groups (typically $k = 5, 10, 20$) of approximately equal size ($\approx n/k$)
  - For $j = 1, \ldots, k$
    - Use the j-th group as **test** set and the remaining as **training** set
    - **Estimate** using the **training** set
    - Use it to **predict** y in the j-th and **compute** $\widehat{MSE}_j$
  - Then

$$\boxed{CV_{(k)} = \frac{1}{k} \sum_{j=1}^{k} \widehat{MSE}_j}$$

- **Leave One Out CV** (LOOCV): Special case obtained for $k = n$ (each group has one observation) $\rightarrow$ comments
- Let's finish: `MBD2017-Energy-20170513.R`

Shrinkage Methods: Lasso Regression

- Only a "small" difference with Ridge Regression: Estimate parameters **minimizing**

$$\boxed{SSR + \underbrace{\lambda \sum_{j=1}^{p} |\beta_j|}_{penalty}}$$

- $\lambda = 0$ ($\rightarrow$ OLS) vs $\lambda = \infty$ ($\rightarrow \widehat{\beta}_1 = \ldots = \widehat{\beta}_p = 0$)
- L_1 penalty instead of L_2 penalty
- Why L_1 in place of L_2? $\rightarrow$ **Sparsity**

Shrinkage Methods: Lasso vs Ridge (1)

Lasso

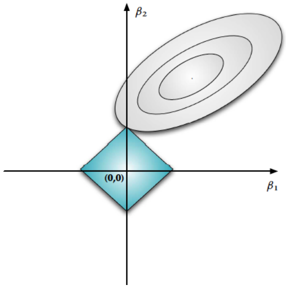
$$\min\left(SSR + \lambda \sum_{j=1}^{p} |\beta_j|\right)$$

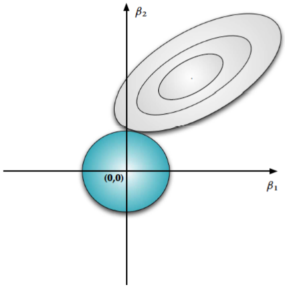
$$\min(SSR) \text{ s.t. } \sum_{j=1}^{p} |\beta_j| \leq s$$

Ridge

$$\min\left(SSR + \lambda \frac{1}{2} \sum_{j=1}^{p} \beta_j^2\right)$$

$$\min(SSR) \text{ s.t. } \frac{1}{2} \sum_{j=1}^{p} \beta_j^2 \leq s$$





Shrinkage Methods: Lasso vs Ridge (2)

	Lasso	**Ridge**
Sparsity (parsimony)	Yes	No
Interpretation	High	Low
Smooth coefficients path	No	Yes
Selection with a group of **highly correlated** predictors	At most one predictor (does not care which one)	All
Performance with $p < n$ **and highly correlated predictors**	Ridge tends to win	
Selection with $p > n$	At most n predictors	All

and so?

Shrinkage Methods: Elastic Nets (1)

- Estimate parameters **minimizing**

$$\boxed{SSR + \lambda \underbrace{\left((1-\alpha)\frac{1}{2}\sum_{j=1}^{p} \beta_j^2 + \alpha \sum_{j=1}^{p} |\beta_j| \right)}_{\textit{penalty}}}$$

where $\alpha \in [0, 1]$

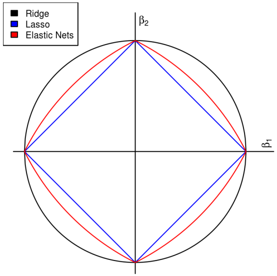
- Equivalent to

$$\min(SSR) \quad \text{s.t.} \quad (1-\alpha)\frac{1}{2}\sum_{j=1}^{p} \beta_j^2 + \alpha \sum_{j=1}^{p} |\beta_j| \leq s$$

(see next slide)

Shrinkage Methods: Elastic Nets (2)

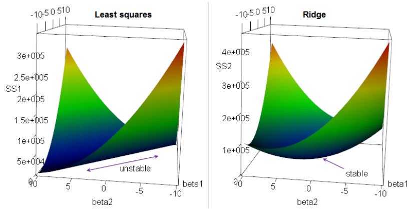
- Parameter's feasible set:



- Motivations:
  - The L_1 part of the penalty generates **sparsity**
  - The L_2 part of the penalty
    - Removes the limitation on the **number** of selected variables
    - Encourages **grouping** effect
    - Stabilizes the **coefficients path**

Names

- **Ridge**



  In case of multicollinearity, the criterion function (*SSR*) has a long valley, corresponding to a **ridge** if $-SSR$ is considered.
  Ridge regression “fixes” the ridge, adding a penalty that turns the ridge into a nice peak
- **LASSO** = Least Absolute Shrinkage and Selection Operator
- **Elastic Net** = Elastic net between the Ridge and the Lasso

Introduction

- We move from a **linear** “world”

$$f(\boldsymbol{x}) = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p$$

  to a possibly **non-linear** one

$$f(\boldsymbol{x}) = \beta_0 + f_1(x_1) + \ldots + f_p(x_p)$$

  while retaining *additivity* of each variable
- We start from the case $p = 1$

$$y = f(x) + \varepsilon$$

  How to structure the (possibly) non-linear $f(\cdot)$?
  Since $f(\cdot)$ is unknown, we need a *flexible* but *simple* way to **represent** it.

How to Represent $f(\cdot)$?

- 1-st idea: polynomial

$$f(\boldsymbol{x}) = \beta_0 + \beta_1 x + \ldots + \beta_k x^d$$

  Pros: simple, easy interpretation
  Cons:
  - Numerically **unstable** for high exponents
  - **Global** structure: polynomials fit well in regions where there are lot of data but take "crazy" shapes elsewhere
- 2-nd idea: piecewise polynomials $\rightarrow$ splines
  - Divide the x range in **intervals**, separated at **boundary points** ($\rightarrow$ **knots**)
  - Use a **low degree polynomial** in each interval ($\rightarrow$ typically, $d = 1$ or $d = \mathbf{3}$)
  - Impose some kind of **smoothness** at boundary points

  Pros: numerical **stability**, **local** structure

Cubic Splines

- **Definition**: 3-rd degree piecewise polynomials C^2 at knots
- Consider K knots $\rightarrow \xi_1 < \ldots < \xi_K$
- How many free coefficients?

$$\underbrace{4(K+1)}_{\text{number of parameters}} - (\underbrace{K+K+K}_{C^2 \text{ continuity}}) = K+4$$

- Direct **math representation** incorporating knot constraints:

$$f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 (x - \xi_1)^3_+ + \ldots + \beta_{K+4} (x - \xi_K)^3_+$$

  Important: **linear** in the parameters!
- There are computationally more efficient representations (although more cumbersome to write). **Linearity** in the parameters is preserved

Natural Cubic Splines

- **Outside the external knots** there are usually few data $\rightarrow$ Variance of $\widehat{f}(\cdot)$ large here $\rightarrow$ How to remedy?
- Impose **boundary constraints**: linearity outside the external knots (equivalent to 2-nd and 3-rd derivatives equal to zero)
- **Natural Cubic Splines** $=$ cubic splines $+$ boundary constraints
- Number of free coefficients

$$\underbrace{4(K+1)}_{\text{number of parameters}} - (\underbrace{K+K+K}_{C^2 \text{ continuity}}) - (\underbrace{2+2}_{\text{boundary contraints}}) = K$$

- Direct **math representation** incorporating knot constraints not simple to write. Anyway, **linearity** in the parameters is retained
- Hereafter natural cubic splines!

How Many Knots? And Where?

Two approaches:

- Regression Splines: $K << n$ and knots placed at "strategic" positions
- Smoothing Splines: knots at unique x values (ties removed), so that $K = n$ (or $\approx n$ with ties)

Regression Splines

For given K ($<< n$) and **location** of knots, it is similar to linear regression (coefficients estimated by minimizing SSR)

- How much K?
  - K large $\Rightarrow$ more flexible $\widehat{f}(\cdot) \rightarrow$ *small* Bias^2, *large* Variance
    K small $\Rightarrow$ less flexible $\widehat{f}(\cdot) \rightarrow$ *large* Bias^2, *small* Variance
  - How to find a good compromise? Compare different K's using k-fold CV
- Where to place knots (for a given K)?
  - Equally spaced **values** over the x range
  - Equally spaced **quantiles** of x (without ties)
  - Adaptive schemes (usually not employed)
- Then, **minimize**

$$SSR = \sum_{i=1}^{n} [y_i - s(x_i)]^2 \qquad s(x) = \beta_1 s_1(x) + \ldots + \beta_K s_K(x)$$

Smoothing Splines

Knots are placed at all unique x values but the SSR is penalized by a term related to the curvature of $\widehat{f}$

- Then, **minimize**

$$PSSR = \sum_{i=1}^{n} [y_i - s(x_i)]^2 + \lambda \int (s''(t))^2 dt$$

$$s(x) = \beta_1 s_1(x) + \ldots + \beta_n s_n(x)$$

- *Remark*: natural cubic splines minimize

$$\sum_{i=1}^{n} [y_i - g(x_i)]^2 + \lambda \int (g''(t))^2 dt$$

for any $g(\cdot)$!

- *Remark*: n coefficients … but they are **not free**!

λ and Degrees of Freedom

- Smoothing splines have n **coefficients**, but they are **not free.** We want to derive an "**equivalent number of free coefficients**" = degrees of freedom = *df*
- In **linear models**, fitted values can be computed as

$$\underset{(n,1)}{\widehat{\boldsymbol{y}}} = \underset{(n,p)}{\boldsymbol{X}}\underset{(p,1)}{\widehat{\beta}} = \underbrace{\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'}_{(n,n)}\underset{(n,1)}{\boldsymbol{y}} = \boldsymbol{H}\boldsymbol{y}$$

  $df = p = \#$ of columns of $X = \text{trace}(\boldsymbol{H})$
- Same approach in **smoothing splines**:

$$\underset{(n,1)}{\widehat{\boldsymbol{y}}} = \underset{(n,n)}{\boldsymbol{H}_\lambda}\underset{(n,1)}{\boldsymbol{y}} \qquad \boxed{df = \text{trace}(\boldsymbol{H}_\lambda)}$$

  $\lambda \uparrow \Leftrightarrow df \downarrow$
- But how to choose λ or *df*?

Choosing λ or *df*

- Use CV, in particular LOOCV based on SSR

$$LOOCV_\lambda = \sum_{i=1}^{n}[y_i - \underbrace{\widehat{y}_i^{(-i)}}_{\text{depends on } \lambda}]^2 = \sum_{i=1}^{n}\left[\frac{y_i - \widehat{y}_i}{1 - \{\boldsymbol{H}_\lambda\}_{i,i}}\right]^2$$

  Also named **PRESS** (Predicted Residual Error Sum of Squares) statistic
- Computationally simple formula, based on the **original fit** to **all** data. This happens any time the model is *linear in the parameters*
- In practice: compute $LOOCV_\lambda$ for different choices of λ (or *df*) then select the best one

Additive Models

- Go back to the complete model

$$y = f(\boldsymbol{x}) + \varepsilon \qquad f(\boldsymbol{x}) = \beta_0 + f_1(x_1) \ldots + f_p(x_p)$$

- Use the approach described for any $f_j(\cdot)$ ($j = 1, \ldots, p$).
- To identify β_0 and the $f_j(\cdot)$'s we impose

$$\sum_{i=1}^{n} f_j(x_{i,j}) = 0 \quad j = 1, \ldots, p \quad \Rightarrow \quad \widehat{\beta}_0 = \overline{y}$$

- $\widehat{f_j}(\cdot)$'s are estimated by **backfitting**. For $j = 1, \ldots, p$:
  - $\widehat{f_j}(\cdot)$ estimated by using partial residuals (residuals on all independent variables but x_j) as dependent variable
  - The procedure is iterated until convergence
- Snack time! `MBD2017-Energy-GAM-20170513.R`

How to test for linearity of one component

- Fit the model under analysis but with the variable under testing (say x) taken linear (in `R`: `x` instead of `s(x)`)
- F-test (D = Residual Deviance = SSE; 0 = linear model)

$$F = \frac{\frac{D_0 - D}{df - df_0}}{\frac{D}{n - df}} | H_0 \approx F(df - df_0, n - df_0)$$

or

$$F = \frac{\frac{D_0 - D}{rdf_0 - rdf}}{\frac{D}{rdf}} | H_0 \approx F(rdf_0 - rdf, rdf_0)$$

(rdf = 'residual df' = $n - df$, see the `R` output)

Introduction

- **Aim**: To build a predictive model without too much assumptions regarding
  - **which variables**
  - their corresponding **functional form**
  - their **interaction**
- **Basic idea**: To resort to iterative/adaptive methods, trying to capture complexity where it is stronger
  1. **Start** from a **“null”** model (no independent variables)
  2. **“Add”** the contribution of the best fitting variable in the data
  3. **“Add”** the contribution of the best fitting variable in the data, possibly in interaction with the variables already included
  4. **Continue** as in 3 until some **stopping criterion** is satisfied
  5. (Possibly) Since the increase in “complexity” (2-4) can happen in a suboptimal way, the model in 4 is **pruned** to reach a reasonable compromise between fit and complexity

Recursive Partitioning

- **Recursive Partitioning** means Recursive Binary Splitting
- Procedure:
  1. Start from a "**null**" model (no predictors)
  2. Find x_j and the cutpoint s which "best" split the data into **two regions**

     $$R_1^{(L)} = \{i : x_{i,j} < s\} \qquad R_1^{(R)} = \{i : x_{i,j} \geq s\}$$

     "Best" means to minimize SSR:

     $$\sum_{i \in R_1^{(L)}} \left(y_i - \overline{y}_{R_1^{(L)}}\right)^2 + \sum_{i \in R_1^{(R)}} \left(y_i - \overline{y}_{R_1^{(R)}}\right)^2$$

  3. Find x_j and the cutpoint s which "best" splits **one region into two sub-regions**
  4. **Continue** as in 3 until some **stopping criterion** is satisfied
  5. The model built can be too complex; use CV to **trim back** the full tree

Recursive Partitioning: Final Product

- *M* **exhaustive non-overlapping** regions (“rectangles”)

$$R_m : m = 1, \ldots, M$$

  Each *i* belongs to only one R_m
- In each region the **prediction** is constant:

$$\bar{y}_{R_m} = \text{mean}\{y_i : i \in R_m\}$$

- Such regions **minimize**

$$SSR = \sum_{m=1}^{M} SSR_m = \sum_{m=1}^{M} \sum_{i \in R_m} \left(y_i - \bar{y}_{R_m}\right)^2$$

- The final product can be visualized by a binary tree where knots are characterized by (*variable*, *cutpoint*, *left*/*right*)

Recursive Partitioning: the Name

- Recursive Partitioning is sometimes known with as CART (Classification And Regression Trees)
- A curiosity:
  - `CART` is trademarked
  - `tree` has been used for an S-Plus package (also available in `R`)
  - The name Recursive PARTitioning (`rpart`) was chosen for the `R` package. Today, `rpart` is more common than the original `CART`...the power of free software!
- Let’s taste: `MBD2017-Energy-rpart-20170606.R`

Introduction

- Topic not covered in the book
- **So far**: We aimed at predicting y using information in $\boldsymbol{x}$
- **Reference formulation**: $y = f(\boldsymbol{x}) + \varepsilon$, $\varepsilon \perp \boldsymbol{x}$ and $\varepsilon \sim [0, \sigma^2]$
- Although interested in predicting y using predictors $\boldsymbol{x}$, there may be reasons to model a transformation of y

$$w = f(\boldsymbol{x}) + \varepsilon \qquad \varepsilon \perp \boldsymbol{x} \text{ and } \varepsilon \sim [0, \sigma^2]$$

  with $w = g(y)$ 1-to-1. To go back, $y = g^{-1}(w)$
- **Examples**:

y	$g(\cdot)$	$g^{-1}(\cdot)$
Energy	$w = \ln y$	$y = \exp(w)$
$\in (0, \infty)$	$w = \sqrt{y}$	$y = w^2$
N. medical examinations	$w = \ln y$	$y = \exp(w)$
$\in [1, \infty)$	$w = \sqrt{y - 1}$	$y = w^2 + 1$

Motivations of and Issues

- **Why transform?**
  - **Model diagnostics** on w can look better than for y: for example, homoskedastic and/or less skewed residuals
  - **Predictions** of y can be better passing through w than directly modeling y
- **Issues**:
  - **Goodness of fit statistics** (R^2, SSR, AIC and similar) on different transformations of y cannot be compared
  - An unbiased **prediction** of y **is not** simply $g^{-1}(\widehat{w})$:

$$\widehat{y} = E(y|\boldsymbol{x}) \underset{y=g^{-1}(w)}{=} E(g^{-1}(w)|\boldsymbol{x}) \underset{g^{-1}(\cdot)\text{ non linear}}{\neq} g^{-1}(E(w|\boldsymbol{x})) = g^{-1}(\widehat{w})$$

So? How to get $\widehat{y}$?

Predicting y when Modeling $w = g(y)$

- **Exact** (case by case)

w	$\widehat{y}$	Assumption
$\sqrt{y}$	$\widehat{w}^2 + \widehat{\sigma}^2$	$\varepsilon \sim (0, \sigma^2)$
$\sqrt{y-1}$	$\widehat{w}^2 + \widehat{\sigma}^2 + 1$	$\varepsilon \sim (0, \sigma^2)$
$\ln(y)$	$e^{\widehat{w}} \widehat{E}(e^{\varepsilon})$	$\varepsilon \sim (0, \sigma^2)$
$\ln(y)$	$e^{\widehat{w}} e^{\widehat{\sigma}^2/2}$	$\varepsilon \sim N(0, \sigma^2)$
$\ln(y/(1-y))$	?	?

- 2-nd order **Taylor expansion** of $g^{-1}(w)$ around $\widehat{w}$

$$\widehat{y} \approx g^{-1}(\widehat{w}) + g^{-1\prime\prime}(\widehat{w})\widehat{\sigma}^2/2$$

- Average using residuals

$$\widehat{y} = \frac{1}{n}\sum_{i=1}^{n} g^{-1}(\widehat{w} + \widehat{\varepsilon}_i)$$

Introduction

Statistical Learning	**Statistics**	Example
Regression	y quantitative (continuous)	`expenditure` $\in [0, +\infty)$
Classification	y qualitative (categorical)	`buy?` $\in \{no, yes\}$

- We could have more than two categories (example?)
- In case of more categories, they can be ordered or not (example)? Models for orderer and unordered categories differ
- Hereafter, we focus on models with two categories, identified as 1 and 0

Modeling Framework (Essential Ideas)

	Distribution	Link	(Linear) Predictor
Regression	$y \mid \boldsymbol{x} \sim N(\mu(\boldsymbol{x}), \sigma^2)$	$\mu = \eta$	$\eta(\boldsymbol{x}) = \boldsymbol{x}'\beta$
Classification	$y \mid \boldsymbol{x} \sim Be(\mu(\boldsymbol{x}))$	$\mu = \frac{e^\eta}{1+e^\eta}$	$\eta(\boldsymbol{x}) = \boldsymbol{x}'\beta$

- The approach can be extended to different **distributional** assumptions (for example to model *counting data*) $\rightarrow$ family of Generalized Linear Models (GLM)
- **Link functions** are monotone
- The linear **predictor** can be replaced by non-linear formulations in $\boldsymbol{x}$ (example GAM)
- **Penalizing** β, we can get Ridge, Lasso or Elastic Nets versions of GLM's
- Estimated by Maximum Likelihood (GLM) or their penalized counterparts (GAM, Ridge, Lasso, Elastic Nets)

Predictions

How to transform

$$\widehat{\mu}(\boldsymbol{x}) = \widehat{P}(y = 1|\boldsymbol{x}) \in [0, 1]$$

to

$$\widehat{y} \in \{0, 1\}?$$

- General classification rule:

$$\boxed{\widehat{y} = 1 \Leftrightarrow \widehat{\mu}(\boldsymbol{x}) > c}$$

- How to set the cut-off c?

Setting the cut-off

- **Typical** classification rule sets $\boxed{c = 0.5}$.
  It makes sense only when 0/1 in the data are balanced.
- **Uninformed** classification rule sets $\boxed{c = \widehat{\mu}}$.
  It takes the *unconditional* probability of 1, $\widehat{\mu}$, as reference.
  It may lead to *biased* in sample classification, in the sense $\sum_{i=1}^{n} \widehat{y}_i \neq \sum_{i=1}^{n} y_i$
- **Unbiased** classification rule sets $\boxed{c = \text{quantile}_{1-\widehat{\mu}}(\widehat{y})}$.
  It implies $\sum_{i=1}^{n} \widehat{y}_i \simeq \sum_{i=1}^{n} y_i$
- **Minimum Cost** rule sets $\boxed{c}$ to minimize the **miss-classification cost** $\rightarrow$ next slide

Minimum Cost Rule

		$\widehat{y}$	
y	$\widehat{P}(y = *\|\boldsymbol{x})$	0	1
0	$1 - \widehat{\mu}(\boldsymbol{x})$	c_{00}	c_{01}
1	$\widehat{\mu}(\boldsymbol{x})$	c_{10}	c_{11}
	$\widehat{E}(cost\|\widehat{y} = *)$	$c_{00}(1-\widehat{\mu}(\boldsymbol{x})) + c_{10}\widehat{\mu}(\boldsymbol{x})$	$c_{01}(1-\widehat{\mu}(\boldsymbol{x})) + c_{11}\widehat{\mu}(\boldsymbol{x})$

$$\widehat{y} = 1 \Leftrightarrow \widehat{E}(cost|\widehat{y} = 1) \leq \widehat{E}(cost|\widehat{y} = 0)$$

$$\Leftrightarrow \widehat{\mu}(\boldsymbol{x}) \geq \frac{1}{1 + \frac{c_{10}-c_{11}}{c_{01}-c_{00}}} \underbrace{=}_{c_{00}=c_{11}=0, c_{10}=1} \frac{c_{01}}{1 + c_{01}}$$

Particular cases:

- If $c_{10} = 1 \rightarrow \widehat{\mu}(\boldsymbol{x}) \geq 0.5$ (typical rule)
- If $c_{10} = \frac{\widehat{\mu}}{1-\widehat{\mu}} \rightarrow \widehat{\mu}(\boldsymbol{x}) \geq \widehat{\mu}$ (uninformed rule)
- If $c_{10} = \frac{qnt}{1-qnt} \rightarrow \widehat{\mu}(\boldsymbol{x}) \geq qnt$ (unbiased rule)

Evaluating Predictions: Confusion Matrix

- **Cells** and **names**

	$\widehat{y}$	
y	0	1
0	True negative	False positive
1	False negative	True positive

- Using a classification rule, one can get a matrix of **frequencies** by cell

	$\widehat{y}$		
y	0	1	
0	n_{00}	n_{01}	$n_{0.}$
1	n_{10}	n_{11}	$n_{1.}$
	$n_{.0}$	$n_{.1}$	n

Evaluating Predictions: Statistics

- The matrix of **frequencies**

$$\begin{array}{c|cc|c} & \multicolumn{2}{c}{\widehat{y}} & \\ y & 0 & 1 & \\ \hline 0 & n_{00} & n_{01} & n_{0.} \\ 1 & n_{10} & n_{11} & n_{1.} \\ \hline & n_{.0} & n_{.1} & n \end{array}$$

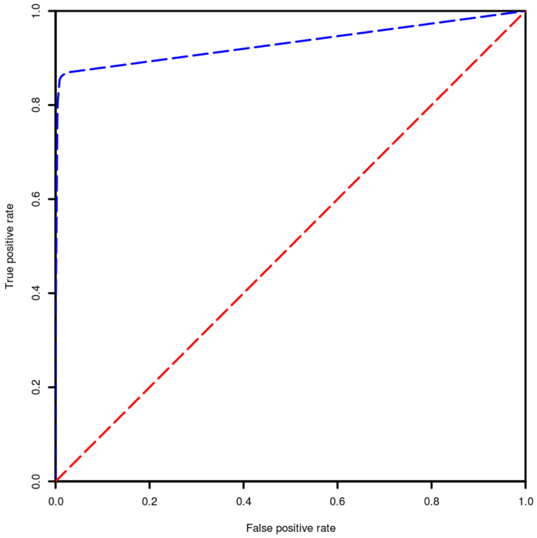
- Classification cost: $n_{10} + n_{01} c_{01}$
- **Classification accuracy**: $(n_{00} + n_{11})/n$
- **Classification error**: $(n_{10} + n_{01})/n$
- (*) **Sensitivity** (True Positive Rate): $\widehat{P}(\widehat{y} = 1|y = 1) = n_{11}/n_{1.}$
- **Specificity** (True Negative Rate): $\widehat{P}(\widehat{y} = 0|y = 0) = n_{00}/n_{0.}$
- (*) **False Positive Rate** (1−Specificity): $\widehat{P}(\widehat{y} = 1|y = 0) = n_{01}/n_{0.} = 1 - n_{00}/n_{0.}$

Evaluating Predictions: The ROC Curve

Plot of $(1 - \textit{Specificity}, \textit{Sensitivity}) = (\textit{FPR}, \textit{TPR})$
when the **cut-off** c **moves** from 0 to 1

- To **understand**: on a blank graph, **report** $(\textit{FPR}, \textit{TPR})$ for the following **key** configurations
  - $c = 0 \rightarrow ?$
  - $c = 1 \rightarrow ?$
  - Perfect predictions (no errors) for any $c \rightarrow ?$
  - Purely random predictions for any $c \rightarrow ?$
- To **paint**: apply the definition ROC Plot
- To **synthesize** (graphical ideas ROC Plot):
  - AUC $\in [0.5, 1]$
  - Accuracy Ratio = Gini $= 2 \cdot AUC - 1 \in [0, 1]$

Evaluating Predictions: ROC Plot



ROC Definition

Further Error/Performance Measures: R^2

$$\boxed{R^2 = \frac{dev(REG)}{dev(y)} \in [0, 1]}$$

- It is based on the **deviance decomposition**

$$dev(y) = dev(REG) + dev(RES)$$

But what such *dev*'s? No more as in the linear model...

- l_0 (**null** log-lik) $\leq l$ (**model** log-lik) $\leq l_S$ (**saturated** log-lik)

$$dev(y) = 2(l_S - l_0)\sigma^2$$

$$dev(REG) = 2(l - l_0)\sigma^2 \qquad dev(RES) = 2(l_S - l)\sigma^2$$

- For each GLM member, such *dev*'s have specific formulas

Further Error/Performance Measures: Cross-Entropy

- With a $0/1$ dependent,

$$dev(RES) = 2 \cdot \underbrace{\left(- \sum_{i=1}^{n} [y_i \ln \mu(\boldsymbol{x}_i) + (1 - y_i) \ln(1 - \mu(\boldsymbol{x}_i))] \right)}_{n \cdot \text{Cross-Entropy}}$$

- Then

$$\boxed{CE = -\frac{1}{n} \sum_{i=1}^{n} [y_i \ln \mu(\boldsymbol{x}_i) + (1 - y_i) \ln(1 - \mu(\boldsymbol{x}_i))] \in [0, \ln 0.5]}$$

Classification Trees vs Regression Trees

- Regression Trees are used y is **quantitative**
- Classification Trees are very similar, with the difference that the dependent variable is **qualitative** $(0/1)$
- **Criterion** to increase/prune the tree: denoting $\widehat{p}_m/\widehat{q}_m =$ estimated proportion of $1/0$ in R_m,
  - **Increasing** based on the sum of

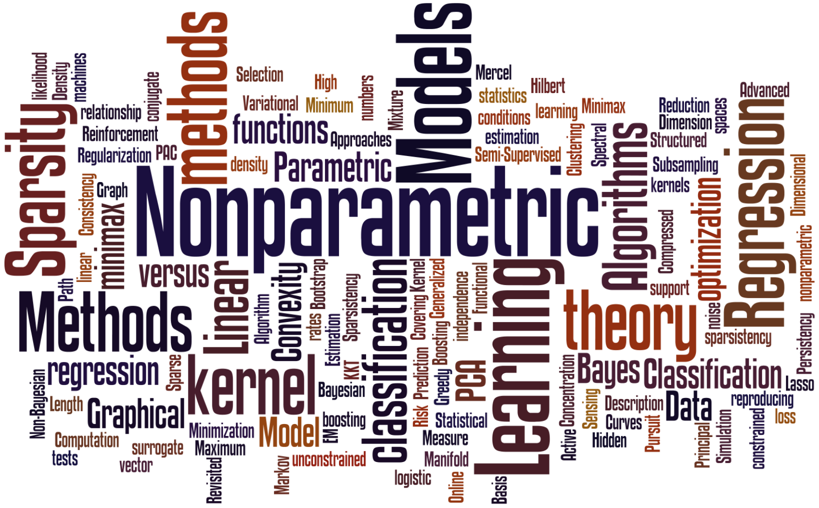
$$\begin{array}{ll} G_m = \widehat{p}_m(1-\widehat{p}_m) + \widehat{q}_m(1-\widehat{q}_m) = 2\widehat{p}_m\widehat{q}_m & \text{Gini Index} \\ D_m = -\left[\widehat{p}_m \log \widehat{p}_m + \widehat{q}_m \log \widehat{q}_m\right] & \text{Cross-Entropy} \end{array}$$

  - **Pruning** based on the sum of

$$E_m = 1 - max\{\widehat{p}_m, \widehat{q}_m\} \quad \text{Classification Error Rate}$$

- For everything else, they work very similarly

Today


Nonparametric
Models
methods
Sparsity
Methods
Learning
Regression
theory
Algorithms
classification
kernel
Linear
Parametric
functions
optimization
Classification
Bayes
Data
Graphical
regression
Model
versus
Convexity
minimax
PCA
likelihood
Density
machines
relationship
conjugate
Reinforcement
Regularization
PAC
Consistency
Graph
linear
Path
Selection
Variational
High
Minimum
numbers
Mixture
Approaches
density
Mercel
statistics
conditions
Hilbert
learning
Minimax
estimation
Semi-Supervised
Clustering
Spectral
Reduction
Dimension
spaces
Structured
Advanced
Subsampling
kernels
Compressed
Dimensional
nonparametric
support
noise
sparsistency
Persistency
Lasso
reproducing
loss
constrained
Simulation
Principal
Pursuit
Description
Curves
Hidden
Sensing
Active Concentration
Basis
Online
Manifold
Measure
Statistical
logistic
Greedy Boosting Generalized
Covering Kernel
Risk Prediction
independence
functional
KKT
Sparsistency
Estimation
rates Bootstrap
Algorithm
Bayesian
boosting
EM
unconstrained
Markov
Minimization
Maximum
Revisited
Sparse
Length
Non-Bayesian
Computation
tests
surrogate
vector

Tomorrow