

Module 4: Coping with Multiple Predictors

Regression Trees

STAT/BIOSTAT 527, University of Washington

Emily Fox

May 15th, 2014

©Emily Fox 2014

1

Recursive Binary Partitions

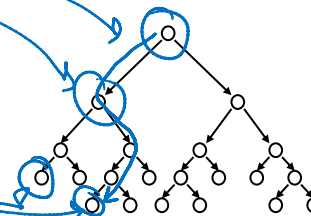
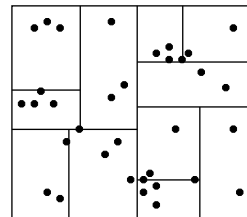
- To simplify the process and interpretability, consider recursive binary partitions

- Described via a rooted tree

- Every node of the tree corresponds to split decision
- Leaves contain a subset of the data that satisfy the conditions

- all conditions on path from root to leaf

- think of pinball falling to leaf

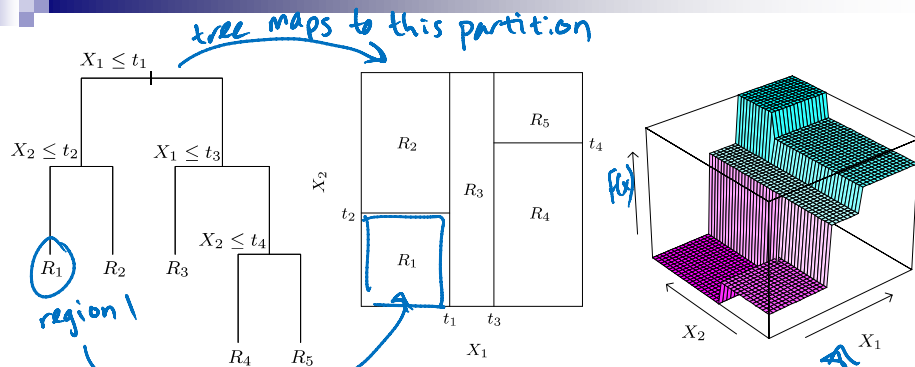


Figures from Andrew Moore kd-tree tutorial

©Emily Fox 2014

2

Resulting Model



- Model the response as constant within each region

$$f(x) = \sum_{m=1}^M \beta_m I(x \in R_m)$$

Figures from Hastie, Tibshirani, Friedman book

©Emily Fox 2014

3

Basis Expansion Interpretation

- Equivalent to a basis expansion

$$f(x) = \sum_{m=1}^M \beta_m h_m(x)$$

indicators on regions

- In this example:

$$h_1(x_1, x_2) = I(x_1 \leq t_1)I(x_2 \leq t_2)$$

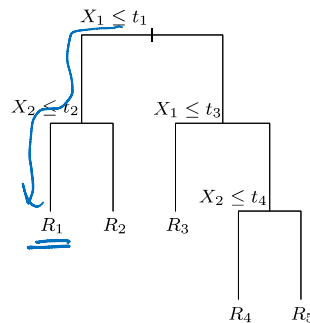
$$h_2(x_1, x_2) = I(x_1 \leq t_1)I(x_2 > t_2)$$

$$h_3(x_1, x_2) = I(x_1 > t_1)I(x_1 \leq t_3)$$

$$h_4(x_1, x_2) = I(x_1 > t_1)I(x_1 > t_3)I(x_2 \leq t_4)$$

$$h_5(x_1, x_2) = I(x_1 > t_1)I(x_1 > t_3)I(x_2 > t_4)$$

reduced tensor product spline w/ step fn basis



©Emily Fox 2014

4

Choosing a Split Decision

- Starting with all of the data, consider splitting on variable j at point s

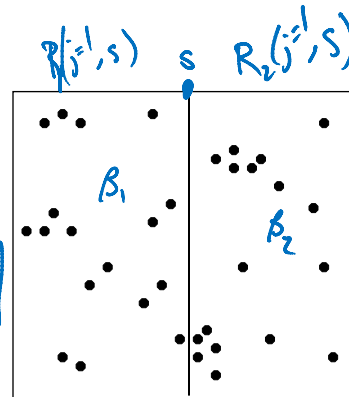
- Define

$$R_1(j, s) = \{x \mid x_j \leq s\}$$

$$R_2(j, s) = \{x \mid x_j > s\}$$

- Our objective is

$$\min_{j,s} \left[\min_{\beta_1} \sum_{x_i \in R_1(j,s)} (y_i - \beta_1)^2 + \min_{\beta_2} \sum_{x_i \in R_2(j,s)} (y_i - \beta_2)^2 \right]$$



- For any (j, s) , the inner minimization is solved by

$$\hat{\beta}_k = \text{avg}(y_i \mid x_i \in R_k(j, s)) \quad k=1,2$$

©Emily Fox 2014

5

Cost-Complexity Pruning

- Searching over all subtrees and selecting using AIC or CV is not possible since there is an exponentially large set of subtrees

→ look at penalized RSS instead

- Define a subtree $T \subset T_0$ to be any tree obtained by pruning T_0

prune = collapse an internal node

and $|T| = \# \text{ of leaf nodes}$

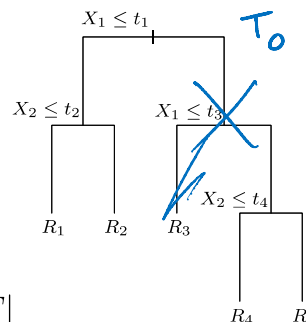
$$n_m = |\{x_i \in R_m\}|$$

$$\hat{\beta}_m = \frac{1}{n_m} \sum_{x_i \in R_m} y_i$$

$$Q_m(T) = \frac{1}{n_m} \sum_{x_i \in R_m} (y_i - \hat{\beta}_m)^2$$

- We examine a complexity criterion

$$C_\lambda(T) = \sum_{m=1}^{|T|} n_m Q_m(T) + \lambda |T|$$



©Emily Fox 2014

6

Cost-Complexity Pruning

compute for λ and all trees in sequence

$$C_\lambda(T) = \sum_{m=1}^{|T|} n_m Q_m(T) + \lambda |T|$$

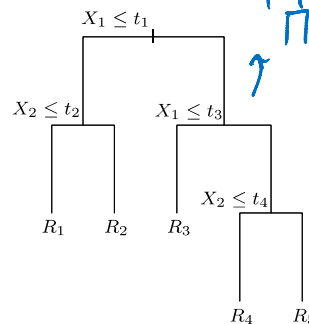
- Can find using *weakest link pruning*
 - Successively collapse the internal node that produces smallest increase in RSS

$$\sum_m n_m Q_m(t)$$

- Continue until at single-node (root) tree
- Produces a finite sequence of subtrees, which must contain T_λ
- See Breiman et al. (1984) or Ripley (1996)

- Choose λ via 5- or 10-fold CV

- Final tree: $T_{\hat{\lambda}}$



©Emily Fox 2014

7

Issues

- Unordered categorical predictors
 - With unordered categorical predictors with q possible values, there are $2^{q-1}-1$ possible choices of partition points to consider for each variable
 - Prohibitive for large q
 - Can deal with this for binary y ...will come back to this in "classification"
- Missing predictor values...how to cope?
 - Can discard
 - Can fill in, e.g., with mean of other variables
 - With trees, there are better approaches
 - Categorical predictors: make new category "missing"
 - Split on observed data. For every split, create an ordered list of "surrogate" splits (predictor/value) that create similar divides of the data. When examining observation with a missing predictor, when splitting on that dimension, use top-most surrogate that is available instead

©Emily Fox 2014

8

Issues

■ Binary splits

- Could split into more regions at every node
- However, this more rapidly fragments the data leaving insufficient data and subsequent levels
- Multiway splits can be achieved via a sequence of binary splits, so binary splits are generally preferred

■ Instability

- Can exhibit high variance
- Small changes in the data → big changes in the tree
- Errors in the top split propagates all the way down
- **Bagging** averages many trees to reduce variance

■ Inference

- Hard...need to account for stepwise search algorithm

©Emily Fox 2014

9

Issues

■ Lack of smoothness

- Fits piecewise constant models...unlikely to believe this structure
- **MARS** address this issue (can view as modification to CART)

■ Difficulty in capturing additive structure

- Imagine true structure is

$$y = \beta_1 I(x_1 < t_1) + \beta_2 I(x_2 < t_2) + \epsilon$$

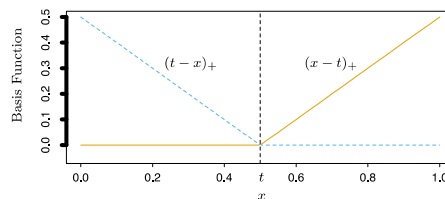
- No encouragement to find this structure

©Emily Fox 2014

10

Multiple Adaptive Regression Splines

- MARS is an adaptive procedure for regression
 - Well-suited to high-dimensional covariate spaces
- Can be viewed as:
 - Generalization of step-wise linear regression
 - Modification of CART
- Consider a basis expansion in terms of piecewise linear basis functions (linear splines)



From Hastie, Tibshirani, Friedman book

©Emily Fox 2014

11

Multiple Adaptive Regression Splines

- Take knots at all observed x_{ij}

$$\mathcal{C} = \{(x_j - t)_{+}, (t - x_j)_{+}\}$$
 - If all locations are unique, then $2 \cdot n \cdot d$ basis functions
 - Treat each basis function as a function on x , just varying with x_j

$$h_m(x) =$$

- The resulting model has the form

$$f(x) = \beta_0 + \sum_{m=1}^M \beta_m h_m(x)$$

- Built in a forward stepwise manner in terms of this basis

©Emily Fox 2014

12

MARS Forward Stepwise

- Given a set of h_m , estimation of β_m proceeds as with any linear basis expansion (i.e., minimizing the RSS)
- How do we choose the set of h_m ?

1. Start with $h_0(x) = 1$ and $M=0$
2. Consider product of all h_m in current model with reflected pairs in C
 - Add terms of the form
$$\hat{\beta}_{M+1}h_\ell(x)(x_j - t)_+ + \hat{\beta}_{M+2}h_\ell(x)(t - x_j)_+$$
 - Select the one that decreases the training error most
3. Increment M and repeat
4. Stop when preset M is hit
5. Typically end with a large (overfit) model, so backward delete
 - Remove term with smallest increase in RSS
 - Choose model based on generalized CV

©Emily Fox 2014

13

MARS Forward Stepwise Example

$$\hat{\beta}_{M+1}h_\ell(x)(x_j - t)_+ + \hat{\beta}_{M+2}h_\ell(x)(t - x_j)_+$$

- At the first stage, add term of form

$$\beta_1(x_j - t)_+ + \beta_2(t - x_j)_+$$

with the optimal pair being

- Add pair to the model and then consider including a pair like

$$\beta_3h_m(x)(x_j - t)_+ + \beta_4h_m(x)(t - x_j)_+$$

with choices for h_m being:

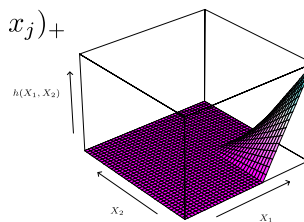


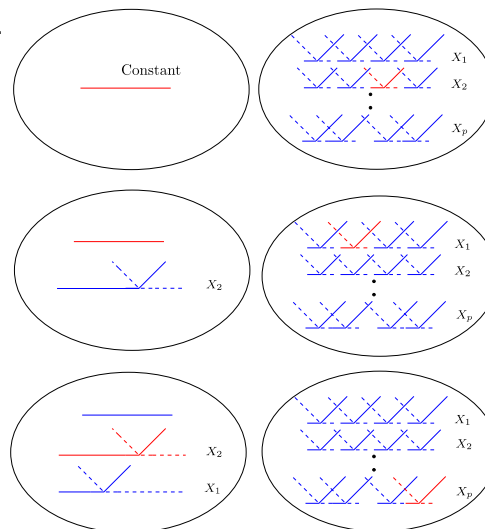
Figure from Hastie, Tibshirani, Friedman book

©Emily Fox 2014

14

MARS Forward Stepwise

■ In pictures...



From
Hastie,
Tibshirani,
Friedman
book

©Emily Fox 2014

15

Why MARS?

■ Why these piecewise linear basis functions?

- Ability to operate locally
 - When multiplied, non-zero only over small part of the input space
 - Resulting regression surface has local components and only where needed (spend parameters carefully in high dims)
- Computations with linear basis are very efficient
 - Naively, we consider fitting n reflected pairs for each input x_j
→ $O(n^2)$ operations
 - Can exploit simple form of piecewise linear function
 - Fit function with rightmost knot. As knot moves, basis functions differ by 0 over the left and by a constant over the right
→ Can try every knot in $O(n)$

©Emily Fox 2014

16

Why MARS?

■ Why forward stagewise?

- Hierarchical in that multiway products are built from terms already in model (e.g., 4-way product exists only if 3-way already existed)
- Higher order interactions tend to only exist if some of the lower order interactions exist as well
- Avoids search over exponentially large space

■ Notes:

- Each input can appear at most once in a product...Prevents formation of higher-order powers of an input
- Can place limit on order of interaction. That is, one can allow pairwise products, but not 3-way or higher.
- Limit of 1 → additive model

©Emily Fox 2014

17

Connecting MARS and CART

■ MARS and CART have lots of similarities

■ Take MARS procedure and make following modifications:

- Replace piecewise linear with step functions
- When a model term h_m is involved in a multiplication by a candidate term, replace it by the interaction and is not available for further interaction

■ Then, MARS forward procedure = CART tree-growing algorithm

- Multiplying a step function by a pair of reflected step functions
= split node at the step
- 2nd restriction → node may not be split more than once (binary tree)

■ MARS doesn't force tree structure → can capture additive effects

©Emily Fox 2014

18

What you need to know

- Regression trees provide an adaptive regression method
- Fit constants (or simple models) to each region of a partition
- Relies on estimating a binary tree partition
 - Sequence of decisions of variables to split on and where
 - Grown in a greedy, forward-wise manner
 - Pruned subsequently
- Implicitly performs variable selection
- MARS is a modification to CART allowing linear fits

©Emily Fox 2014

19

Readings

- Wakefield – 12.7
- Hastie, Tibshirani, Friedman – 9.2.1-9.2.2, 9.2.4, 9.4
- Wasserman – 5.12

©Emily Fox 2014

20

Module 4: Coping with Multiple Predictors

A Short Case Study

STAT/BIOSTAT 527, University of Washington

Emily Fox

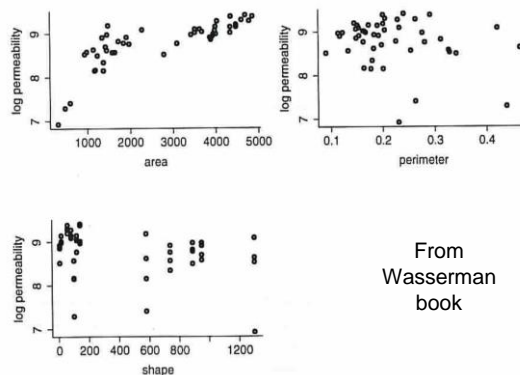
May 15th, 2014

©Emily Fox 2014

21

Rock Data

- 48 rock samples from a petroleum reservoir
- Response = permeability
- Covariates = area of pores, perimeter, and shape



From
Wasserman
book

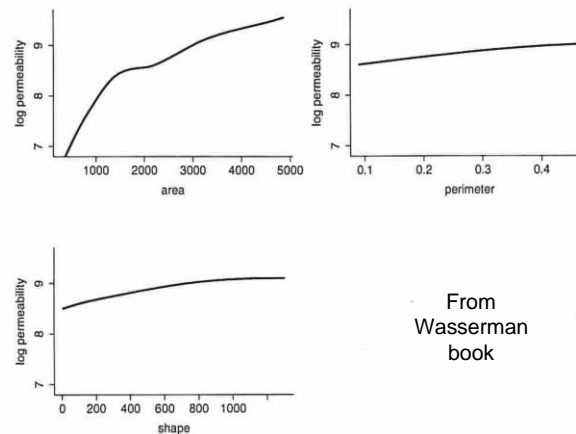
©Emily Fox 2014

22

Generalized Additive Model

- Fit a GAM:

$$\text{permeability} = f_1(\text{area}) + f_2(\text{perimeter}) + f_3(\text{shape}) + \epsilon$$



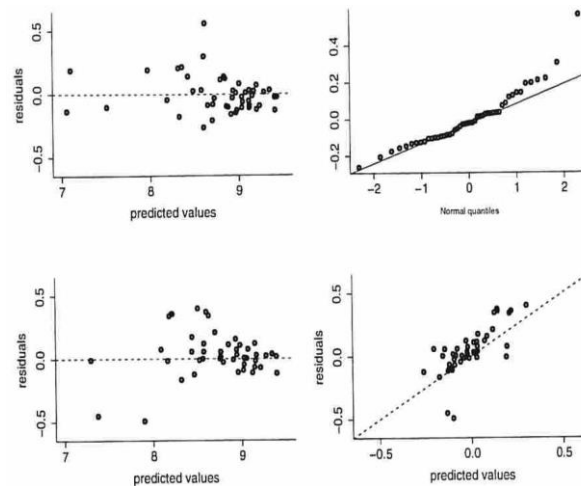
From
Wasserman
book

©Emily Fox 2014

23

GAM vs. Local Linear Fits

- Comparison to a 3-dimensional local linear fit



From
Wasserman
book

©Emily Fox 2014

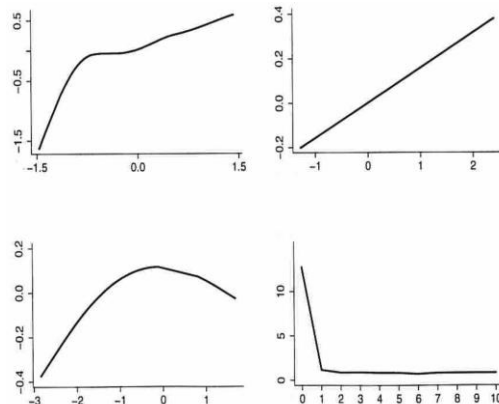
24

Projection Pursuit

$$f(x_1, \dots, x_d) = \alpha + \sum_{m=1}^M f_m(w_m^T x)$$

- Applying projection pursuit with $M = 3$ yields

$$w_1 = (.99, .07, .08)^T, \quad w_2 = (.43, .35, .83)^T, \quad w_3 = (.74, -.28, -.61)^T$$



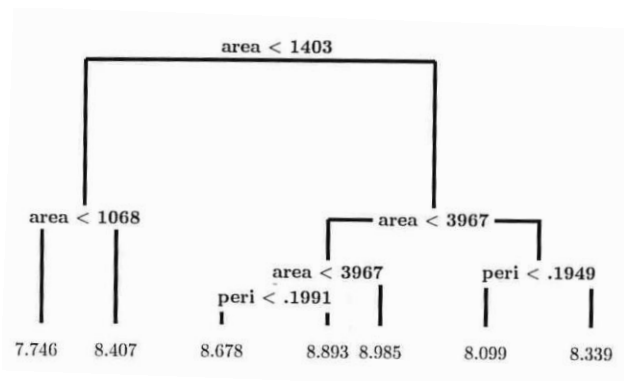
From
Wasserman
book

©Emily Fox 2014

25

Regression Trees

- Fit a regression tree to the rock data
- Note that the variable “shape” does not appear in the tree



From Wasserman book

©Emily Fox 2014

26

Module 5: Classification

A First Look at Classification: CART

STAT/BIOSTAT 527, University of Washington

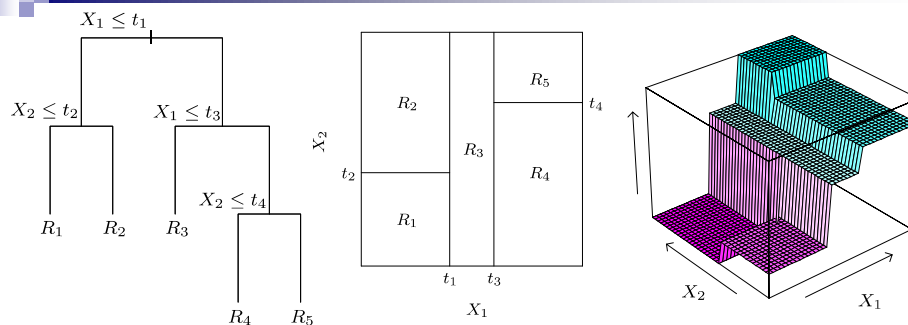
Emily Fox

May 15th, 2014

©Emily Fox 2014

27

Regression Trees



- So far, we have assumed continuous responses y and looked at regression tree models:

$$f(x) = \sum_{m=1}^M \beta_m I(x \in R_m)$$

Figures from Hastie, Tibshirani, Friedman book

©Emily Fox 2014

28

Classification Trees

- What if our response y is **categorical** and our goal is classification?

- Can we still use these tree structures?
- Recall our **node impurity** measure

$$Q_m(T) = \frac{1}{n_m} \sum_{x_i \in R_m} (y_i - \hat{\beta}_m)^2$$

- Used this for growing the tree

$$\min_{j,s} \left[\sum_{x_i \in R_1(j,s)} (y_i - \hat{\beta}_1)^2 + \sum_{x_i \in R_2(j,s)} (y_i - \hat{\beta}_2)^2 \right]$$

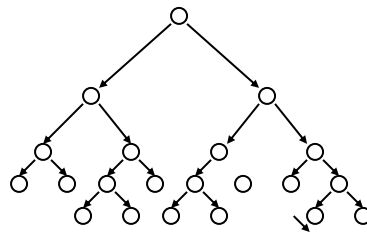
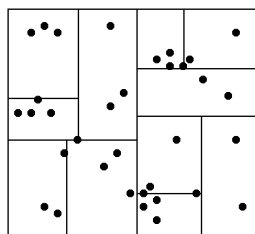
- As well as pruning $C_\lambda(T) = \sum_{m=1}^{|T|} n_m Q_m(T) + \lambda |T|$

- Clearly, squared-error is not the right metric for classification

©Emily Fox 2014

29

Classification Trees



- First, what is our decision rule at each leaf?
- Estimate probability of each class given data at leaf node:

$$\hat{p}_{mk} =$$

- Majority vote:

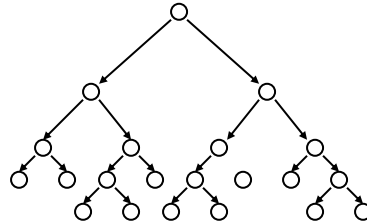
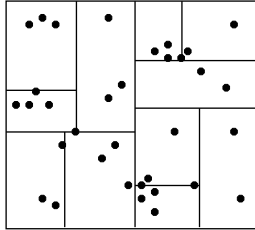
$$k(m) =$$

Figures from Andrew Moore kd-tree tutorial

©Emily Fox 2014

30

Classification Trees



- How do we measure **node impurity** for this fit/decision rule?

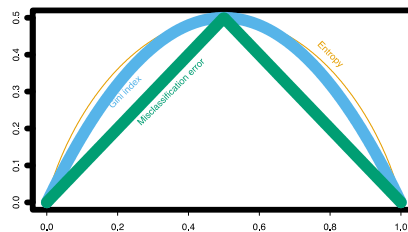
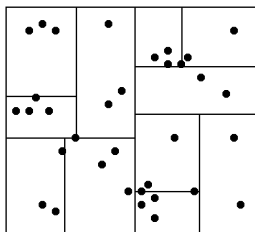
- ☐ Misclassification error:
- ☐ Gini index:
- ☐ Cross-entropy or deviance:

Figures from Andrew Moore kd-tree tutorial

©Emily Fox 2014

31

Classification Trees



From
Hastie,
Tibshirani,
Friedman
book

- How do we measure **node impurity** for this fit/decision rule?

- ☐ Misclassification error (K=2):
- ☐ Gini index (K=2):
- ☐ Cross-entropy or deviance (K=2):

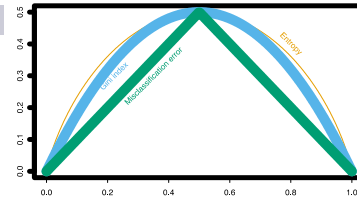
©Emily Fox 2014

32

Notes on Impurity Measures

■ Impurity measures

- Misclassification error: $1 - \hat{p}_{mk(m)}$
- Gini index: $\sum_{k=1}^K \hat{p}_{mk}(1 - \hat{p}_{mk})$
- Cross-entropy or deviance: $-\sum_{k=1}^K \hat{p}_{mk} \log \hat{p}_{mk}$



From Hastie, Tibshirani,
Friedman book

■ Comments:

- Differentiability
- Sensitivity to changes in node probabilities

- Often use Gini or cross-entropy for growing tree, and misclass. for pruning

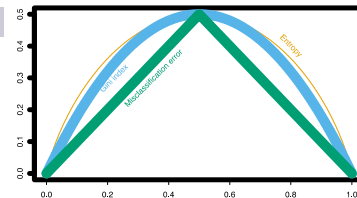
©Emily Fox 2014

33

Notes on Impurity Measures

■ Impurity measures

- Misclassification error: $1 - \hat{p}_{mk(m)}$
- Gini index: $\sum_{k=1}^K \hat{p}_{mk}(1 - \hat{p}_{mk})$
- Cross-entropy or deviance: $-\sum_{k=1}^K \hat{p}_{mk} \log \hat{p}_{mk}$



From Hastie, Tibshirani,
Friedman book

■ Other interpretations of Gini index:

- Instead of majority vote, classify observations to class k with prob. \hat{p}_{mk}

- Code each observation as 1 for class k and 0 otherwise

- Variance:

- Summing over k gives the Gini index

©Emily Fox 2014

34

Classification Tree Issues

■ Unordered categorical predictors

- With unordered categorical predictors with q possible values, there are $2^{q-1}-1$ possible choices of partition points to consider for each variable
- For binary (0-1) outcomes, can order predictor classes according to proportion falling in outcome class 1 and then treat as ordered predictor
 - Gives optimal split in terms of cross-entropy or Gini index
- Also holds for quantitative outcomes and square-error loss...order predictors by increasing mean of the outcome
- No results for multi-category outcomes

■ Loss matrix

- In some cases, certain misclassifications are worse than others
- Introduce **loss matrix** ...more on this soon
- See Tibshirani, Hastie and Friedman for how to incorporate into CART

©Emily Fox 2014

35

Classification Tree Spam Example

■ Example: *predicting spam*

■ Data from UCI repository

■ Response variable: *email* or *spam*

■ 57 predictors:

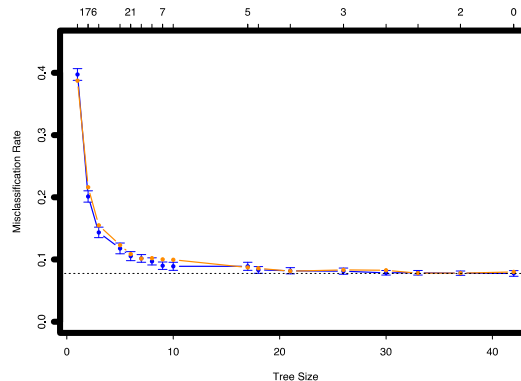
- 48 quantitative – percentage of words in email that match a give word such as “business”, “address”, “internet”,...
- 6 quantitative – percentage of characters in the email that match a given character (; , [! \$ #)
- The average length of uninterrupted capital letters: CAPAVE
- The length of the longest uninterrupted sequence of capital letters: CAPMAX
- The sum of the length of uninterrupted sequences of capital letters: CAPTOT

©Emily Fox 2014

36

Classification Tree Spam Example

- Used cross-entropy to grow tree and misclassification to prune
- 10-fold CV to choose tree size
 - CV indexed by λ
 - Sizes refer to $|T_\lambda|$
 - Error rate flattens out around a tree of size 17



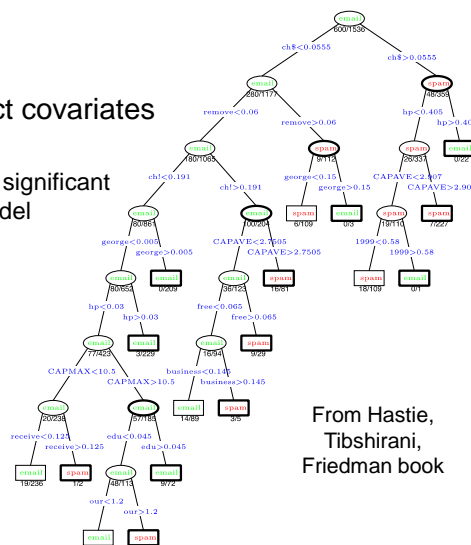
From Hastie, Tibshirani, Friedman book

©Emily Fox 2014

37

Classification Tree Spam Example

- Resulting tree of size 17
- Note that there are 13 distinct covariates split on by the tree
 - 11 of these overlap with the 16 significant predictors from the additive model previously explored



From Hastie,
Tibshirani,
Friedman book

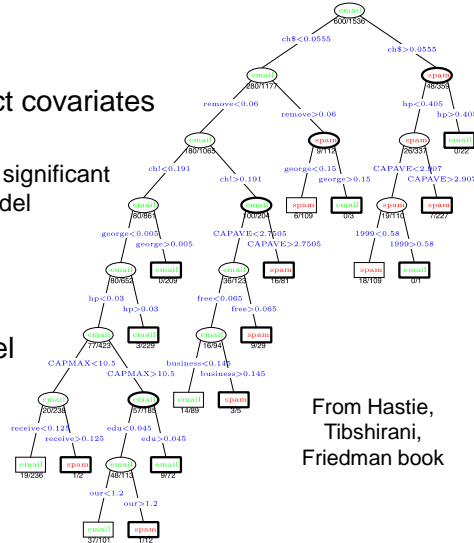
©Emily Fox 2014

38

Classification Tree Spam Example

- Resulting tree of size 17
- Note that there are 13 distinct covariates split on by the tree
 - 11 of these overlap with the 16 significant predictors from the additive model previously explored
- Overall error rate (9.3%) is higher than for additive model

True \ Predicted		
	email	spam
email	57.3%	4.0%
spam	5.3%	33.4%



©Emily Fox 2014

39

What you need to know

- Classification trees are a straightforward modification to the regression tree setup
- Just need new definition of node impurity for growing and pruning tree
- Decision at the leaves is a simple majority-vote rule

©Emily Fox 2014

40

Readings

- Wakefield – 10.3.2, 10.4.2, 12.8.4
- Hastie, Tibshirani, Friedman – 9.2.3, 9.2.5, 2.4