

## Module 4: Coping with Multiple Predictors

# Multidimensional Kernel Methods

STAT/BIOSTAT 527, University of Washington

Emily Fox

May 13<sup>th</sup>, 2014


©Emily Fox 2014

1


## Kernel Density Estimation

- Kernel methods are often used for density estimation (actually, classical origin)

- Assume random sample  $x_1, \dots, x_n \stackrel{\text{iid}}{\sim} P$

- Choice #1: empirical estimate?  $\hat{p} = \frac{1}{n} \sum \delta_{x_i}$  

- Choice #2: as before, maybe we should use an estimator


$$\hat{p}(x_0) = \frac{\#x_i \in \text{Nbhd}(x_0)}{n \lambda}$$

width of nbhd

- Choice #3: again, consider kernel weightings instead

$$\hat{p}(x_0) = \frac{1}{n\lambda} \sum K_\lambda(x_0, x_i)$$

Parzen est.

©Emily Fox 2014

2

# Kernel Density Estimation

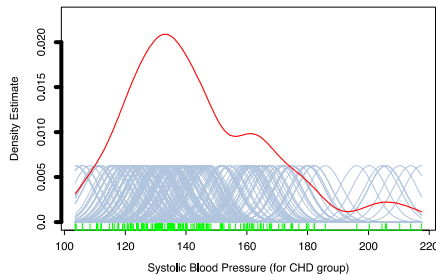
- Popular choice = Gaussian kernel → **Gaussian KDE**

$$\hat{p} = \frac{1}{n} \sum_{i=1}^n \phi_{\lambda}(x - x_i)$$

$\phi_{\lambda}$

$$= (\hat{p} * \phi_{\lambda})(x)$$

↑ empirical dist.



From Hastie, Tibshirani, Friedman book

©Emily Fox 2014

3

# Multivariate KDE

- In 1d 
$$\hat{p}(x_0) = \frac{1}{n\lambda} \sum_{i=1}^n K_{\lambda}(x_0, x_i)$$

- In  $\mathbb{R}^d$ , assuming a product kernel,

$$\hat{p}(x_0) = \frac{1}{n\lambda_1 \cdots \lambda_d} \sum_{i=1}^n \left\{ \prod_{j=1}^d K_{\lambda_j}(x_{0j}, x_{ij}) \right\}$$

- Typical choice = Gaussian RBF

©Emily Fox 2014

4

# Multivariate KDE

$$\hat{p}(x_0) = \frac{1}{n\lambda_1 \cdots \lambda_d} \sum_{i=1}^n \left\{ \prod_{j=1}^d K_{\lambda_j}(x_{0j}, x_{ij}) \right\}$$

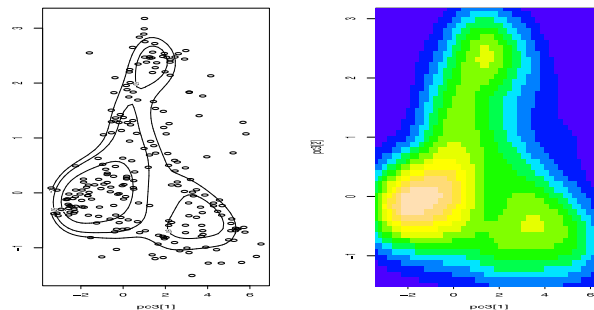
- Risk grows as  $O(n^{4/(4+d)})$
- Example: To ensure relative MSE  $< 0.1$  at 0 when the density is a multivariate norm and optimal bandwidth is chosen
- Always report confidence bands, which get wide with  $d$

©Emily Fox 2014

5

## Multivariate KDE Example

- Data on 6 characteristics of aircraft (Bowman and Azzalini 1998)
- Examine first 2 principle components of the data
- Perform KDE with independent kernels

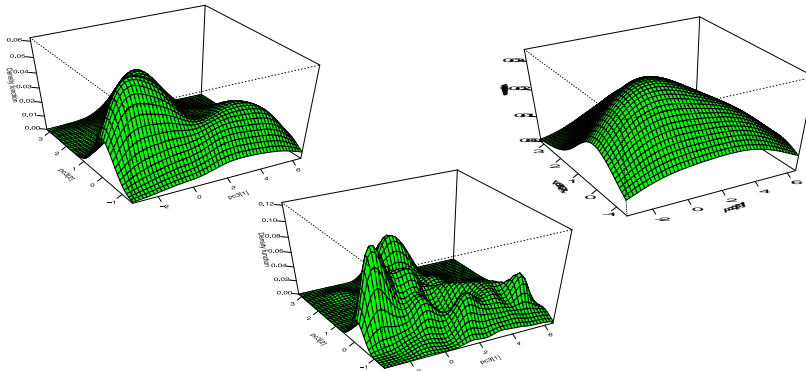


©Emily Fox 2014

6

# Multivariate KDE Example

- Data on 6 characteristics of aircraft (Bowman and Azzalini 1998)
- Examine first 2 principle components of the data
- Perform KDE with independent kernels



©Emily Fox 2014

7

## Module 4: Coping with Multiple Predictors

### Regression Trees

STAT/BIOSTAT 527, University of Washington

Emily Fox

May 13<sup>th</sup>, 2014

©Emily Fox 2014

8

# Regression Trees Overview $y \in \mathbb{R}$

- An alternative adaptive regression technique

- Conceptually simple
- Powerful

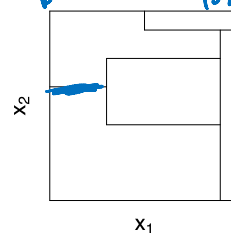
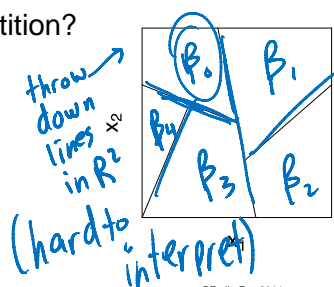
(interpretable)

- Partition the covariate space into regions and then fit a simple model in each (e.g., constant)

$x \in \mathbb{R}^2$

(axis-aligned cuts) (still hard to interpret)

- How to partition?



©Emily Fox 2014

9

## Recursive Binary Partitions

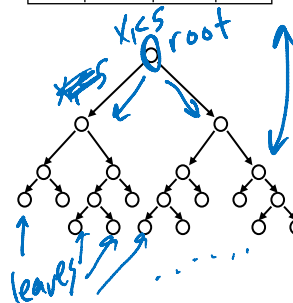
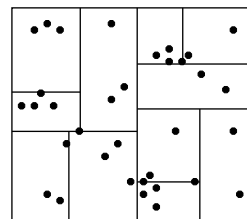
- To simplify the process and interpretability, consider recursive binary partitions

- 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

$(x_1 < 5)$

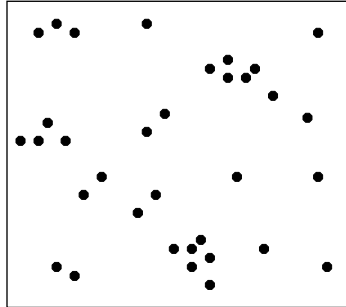
- all conditions on path from root to leaf  
- think of a pinball falling to leaf



©Emily Fox 2014

10

# Recursive Binary Partitions



Pt	$x_1$	$x_2$
1	0.00	0.00
2	1.00	4.31
3	0.13	2.85
...	...	...

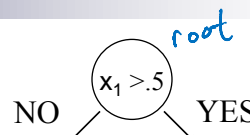
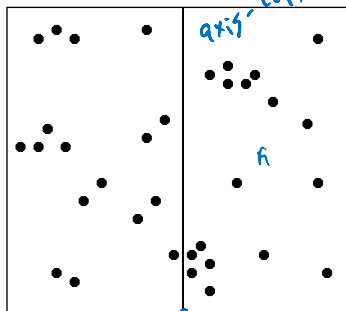
$(x_{i1}, x_{i2})$   
 $(x_1, y_1)$   
 $\vdots$

- Start with a list of  $d$ -dimensional points.

©Emily Fox 2014

11

# Recursive Binary Partitions



Pt	$x_1$	$x_2$
1	0.00	0.00
3	0.13	2.85
...	...	...

Pt	$x_1$	$x_2$
2	1.00	4.31
...	...	...

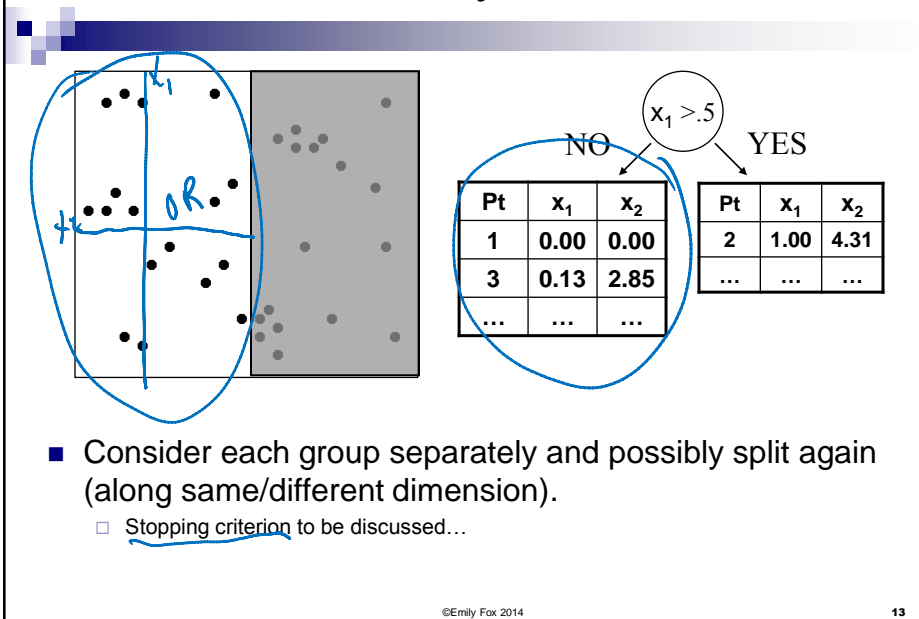
- Split the points into 2 groups by:
  - Choosing dimension  $d_j$  and value  $t_j$  (methods to be discussed...)
  - Separating the points into  $x_{id_j} > t_j$  and  $x_{id_j} \leq t_j$ .

here:  $d_j = 1$  ( $x_1$ )  
 $t_j = 0.5$

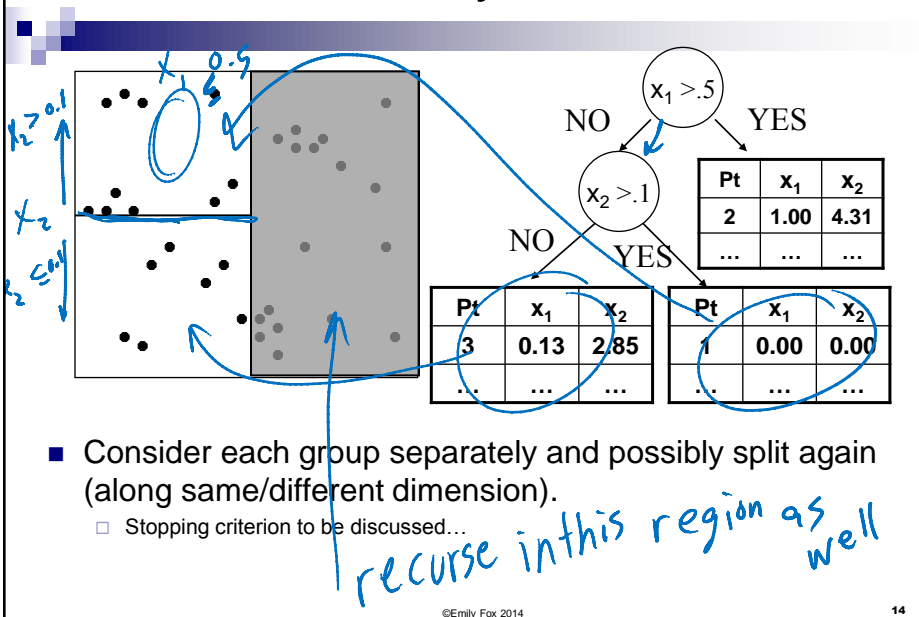
©Emily Fox 2014

12

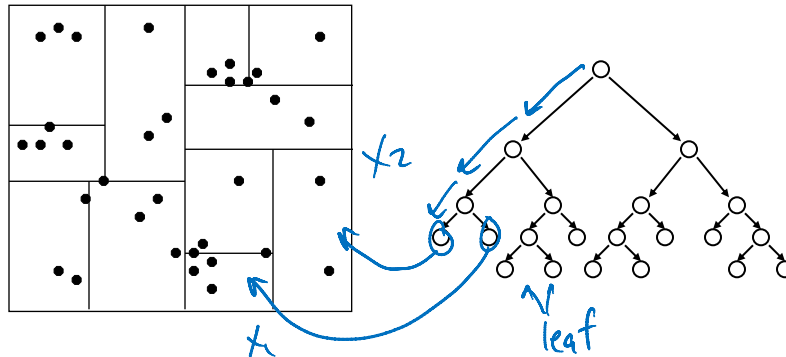
# Recursive Binary Partitions



# Recursive Binary Partitions



# Recursive Binary Partitions



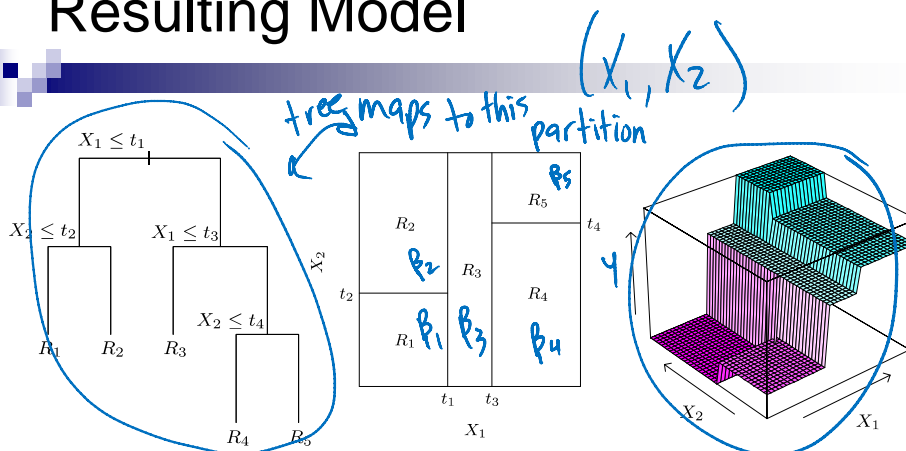
- Continue splitting points in each set
  - creates a binary tree structure
- Each leaf node contains a list of points

*that satisfy all the conditions down the tree to that point*

©Emily Fox 2014

15

# Resulting Model



- Model the response as constant within each region

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

©Emily Fox 2014

16



# Basis Expansion Interpretation

- Equivalent to a basis expansion

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

$I(x \in R_m)$   
↑ indicators on each region

- In this example:

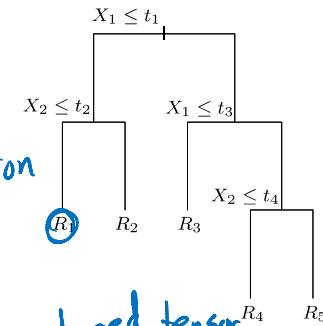
$$h_1(x_1, x_2) = I(x_1 \leq t_1)I(x_2 \leq t_2)$$

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

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

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

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



reduced tensor product spline w/ step fn. basis

©Emily Fox 2014

17

# Questions on Building the Tree

- Which variable should we split on?
- What threshold value should we consider?
- When should we stop the process?

could run until 1 obs. per leaf,  
but its prone  
to overfitting

©Emily Fox 2014

18

# Building the Tree

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

- Assume the partition  $(R_1, \dots, R_M)$  is given
- If criterion is to minimize RSS, then

$$\hat{\beta}_m = \text{avg}(y_i | x_i \in R_m) \quad \sum (y_i - \beta)^2 \rightarrow \hat{\beta} = \bar{y}$$

- How do we find the partition  $(R_1, \dots, R_M)$ ?
  - Finding the optimal tree that minimizes RSS is generally computationally infeasible
  - Consider a greedy algorithm instead

©Emily Fox 2014

19

# Choosing a Split Decision

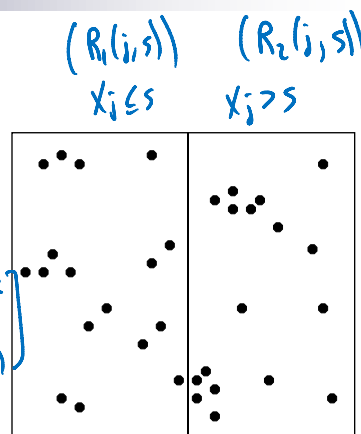
- Starting with all of the data, consider splitting on variable  $j$  at point  $s$
- Define

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

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

- 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

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

©Emily Fox 2014

20

# Choosing a Split Decision

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

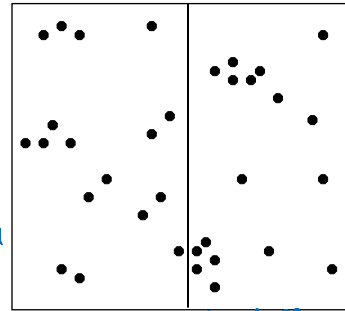
$$\hat{\beta}_1 = \text{avg}(y_i \mid x_i \in R_1(j,s))$$

$$\hat{\beta}_2 = \text{avg}(y_i \mid x_i \in R_2(j,s))$$

- For each splitting variable  $j$ , finding the optimal  $s$  can be done efficiently

□ Why?

*Start at one end*  
*obj. only changes when "s" passes an observation*  
*update to  $\hat{\beta}_1, \hat{\beta}_2$  is  $O(1)$  ... obs. diff*



- Max of  $d(n-1)$  partitions to consider
- So, determining  $(j,s)$  is feasible

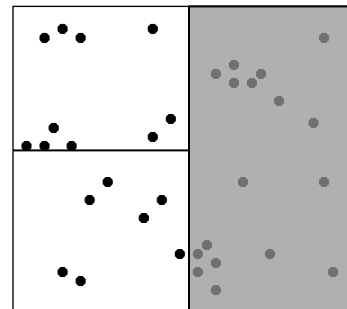
©Emily Fox 2014

21

# Choosing a Split Decision

- Conditioning on the best split just found, we recurse on each of the two regions
- Repeat on all resulting regions
- When do we stop recursing?

*←*



©Emily Fox 2014

22

# How Large of a Tree?

- Large tree, like partitioning until each node has one observation  
→ *overfit (↑ variance, ↓ bias)*
- Small tree → *simple, miss key features (↓ variance, ↑ bias)*
- Tree size is a tuning parameter that governs model complexity
  - Optimal tree size should be chosen adaptively from the data
- Stopping criterion
  - Stop when decrease in RSS due to a split falls below some threshold  
*shortsighted b/c splits later on could be very good*
  - Stop when a minimum node size (e.g., 5) is reached. Go back and prune.  
*easy how?*

©Emily Fox 2014

23

# 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*

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

and  $|T| =$  *# of leaf nodes*

$n_m =$  *# of pts. in leaf node*

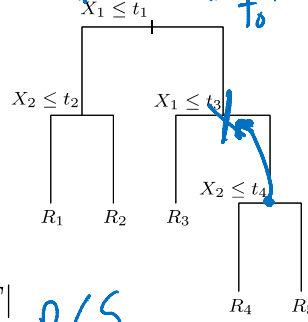
$\hat{\beta}_m =$   *$\bar{y}_m$  i.e.  $\frac{1}{n_m} \sum y_i$*

$Q_m(T) =$   *$\frac{1}{n_m} \sum (y_i - \hat{\beta}_m)^2$*

- We examine a complexity criterion

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

*total RSS*



©Emily Fox 2014

24

# Cost-Complexity Pruning

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

penalty

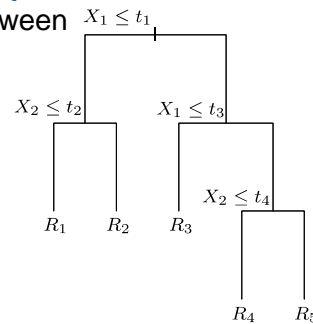
RSS over regions

- For a given  $\lambda$ , want to find  $T_\lambda \subset T_0$  to minimize  $C_\lambda(T)$

- Tuning parameter  $\lambda$  governs tradeoff between tree size and goodness of fit to the data

- Large  $\lambda \rightarrow$  small trees
- $\lambda = 0 \rightarrow$  full tree

- For each  $\lambda$ , can show that there is a unique smallest subtree  $T_\lambda$



©Emily Fox 2014

25

# Cost-Complexity Pruning

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

- Can find using *weakest link pruning*

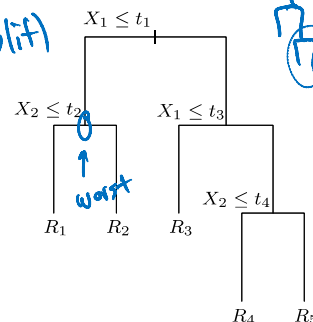
- Successively collapse the internal node that produces smallest increase in RSS

(worst internal split)

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

- Choose  $\hat{\lambda}$  via 5- or 10-fold CV

- Final tree:



sequence



©Emily Fox 2014

26

# Comments on Regression Trees

- Partition is not specified apriori, so regression trees provide a ***locally adaptive*** technique
- Effectively performs variable selection by discovering the relevant interaction terms
  - Implicit in the process
- In the construction, we are assuming that
  - Error terms are uncorrelated
  - Constant variance

recall tensor product basis.

→ RSS is right minim. metric

©Emily Fox 2014

27

## Example: Prostate Cancer

- Fit binary regression tree to log PSA with splits based on eight covariates
- Grow tree with condition of at least 3 observation per leaf
- Results in a tree with 27 splits
- Run weakest-link pruning for each candidate  $\lambda$ , with  $\lambda$  chosen according to CV

©Emily Fox 2014

28

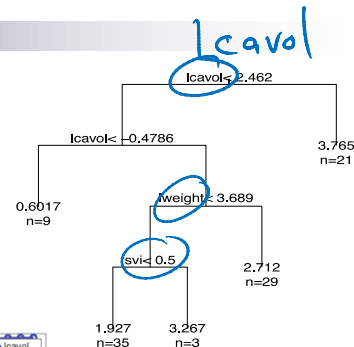
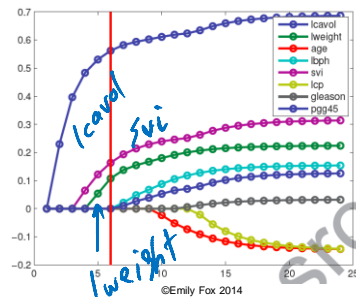
# Example: Prostate Cancer

## ■ Compare results to LASSO

- ☐ lccavol most “important”
- ☐ Then lweight and svi

$$\begin{aligned}
 h_1(x) &= I(\text{lccavol} < -0.4786) \\
 h_2(x) &= I(\text{lccavol} < -0.4786) \quad I(\text{lweight} < 3.689) \quad I(\text{svi} < 0.5) \\
 h_3(x) &= I(\text{lccavol} < -0.4786) \quad I(\text{lweight} < 3.689) \quad I(\text{svi} > 0.5) \\
 h_4(x) &= I(\text{lccavol} < -0.4786) \quad I(\text{lweight} > 3.689) \\
 h_5(x) &= I(\text{lccavol} > 2.462).
 \end{aligned}$$

↑  
basis  
fns



29

## 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

30

# 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

31

# 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

32



## 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

33

## Readings

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

©Emily Fox 2014

34