Data Mining (W4240 Section 001)
Trees

Giovanni Motta

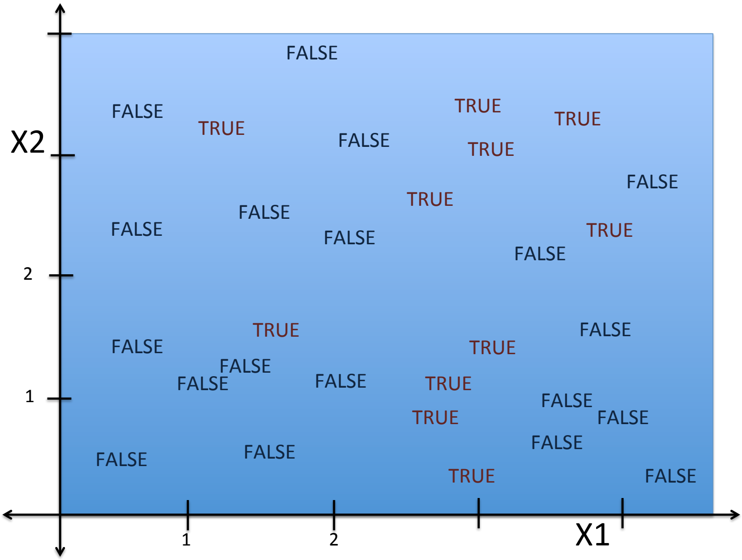
Columbia University, Department of Statistics

November 16, 2015

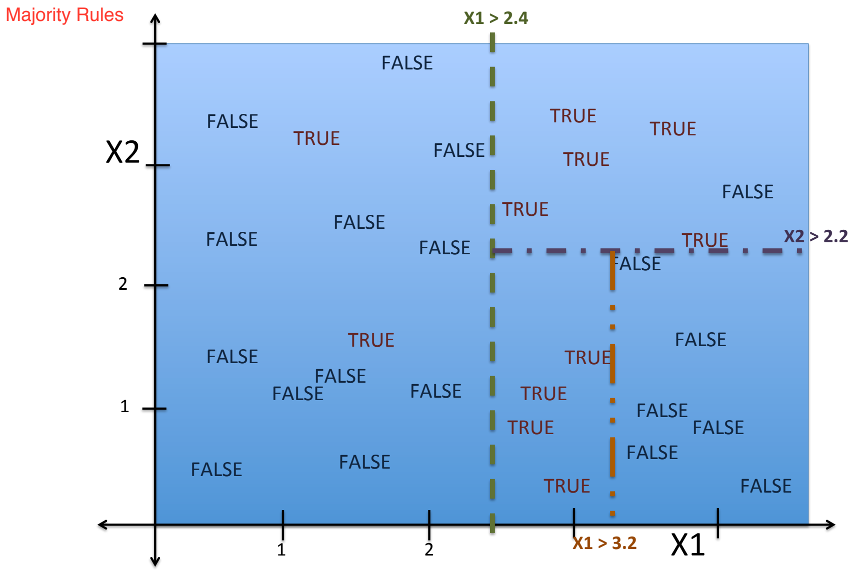
Outline

Outline

Create a classifier: LDA, QDA, kNN



Setup: another way?



Tree-based methods

- *stratifying* or *segmenting* the predictor space into a number of simple regions
- set of splitting rules used to segment the predictor space can be summarized in a tree
- Tree-based methods are simple and useful for interpretation.
- However, they typically are not competitive with the best supervised learning approaches in terms of prediction accuracy
- We will see that combining a large number of trees can often result in dramatic improvements in prediction accuracy.

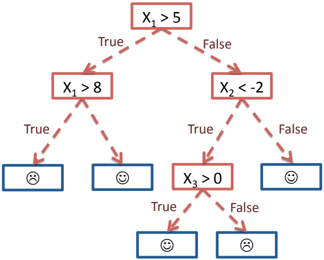
  Just like bagging in bootstraps

Outline

Trees

Suppose that we want to construct a set of rules to represent the data

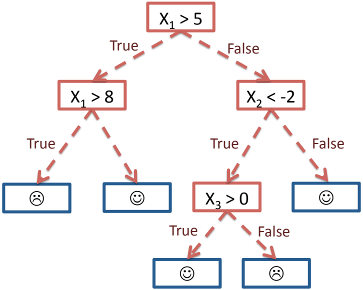
- can represent data as a series of if-then statements
- here, "if" splits inputs into two categories
- "then" assigns value
- when "if" statements are nested, structure is called a tree



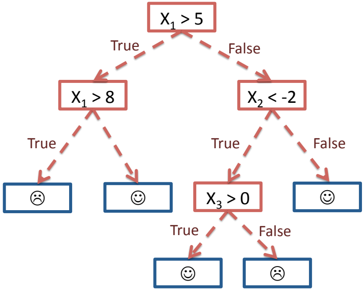
Trees

Example: data (X_1, X_2, X_3, Y) with X_1, X_2, X_3 real, $Y \in \{☺, ☹\}$

- if $X_1 > 5$:
  - if $X_1 > 8$:
    - return ☹
  - else
    - return ☺
- else
  - if $X_2 < -2$:
    - if ...
  - else
    - return ☺



Trees



Example 1: $(X_1, X_2, X_3) = (1, 1, 1)$

Example 2: $(X_1, X_2, X_3) = (10, -3, 0)$

Trees

Terminology:

- branches: one side of a split
- leaves: terminal nodes that return values

Why trees?

- trees can be used for regression or classification
  - regression: returned value is a real number
  - classification: returned value is a class
- unlike linear regression, SVMs, naive Bayes, etc, trees fit *local models*
  - in large spaces, global models may be hard to fit
  - results may be hard to interpret
- fast, interpretable predictions

Example: Predicting Electoral Results (Classification)

2008 Democratic primary:

- Hillary Clinton
- Barack Obama

Given historical data, how will a county vote?

- can extrapolate to state level data
- might give regions to focus on increasing voter turnout
- would like to know how variables interact

Example: Predicting Electoral Results

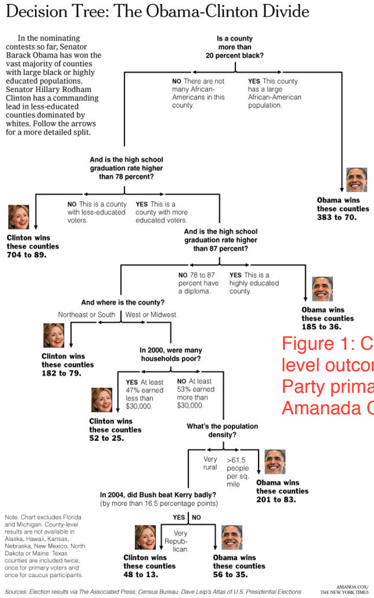


Figure 1: Classification tree for county-level outcomes in the 2008 Democratic Party primary (as of April 16), by Amanada Cox for the New York Times.

Figure 1: Classification tree for county-level outcomes in the 2008 Democratic Party primary (as of April 16), by Amanada Cox for the New York *Times*.

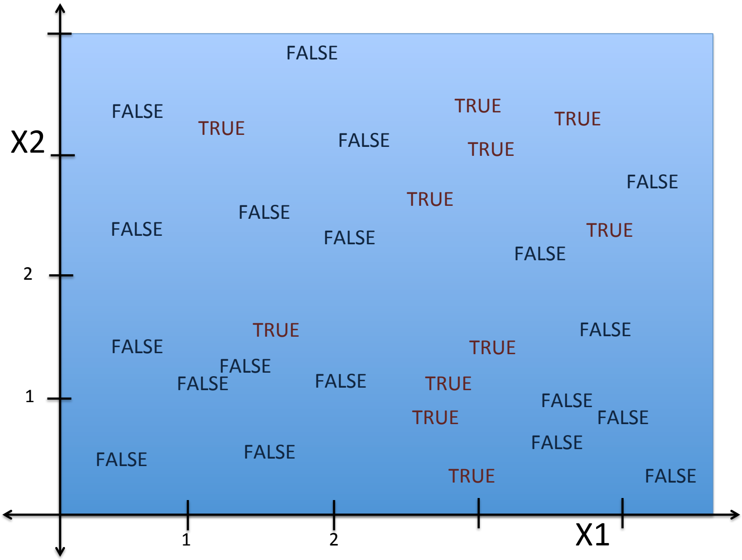
Trees

Local modeling:

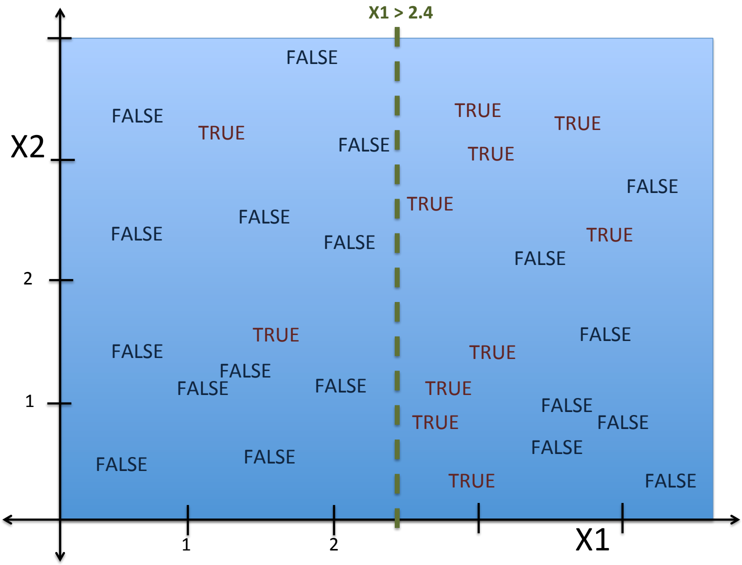
- good way to find interactions within data
- do by partitioning the data into two regions
- within each region, stop or further partition
- process is called recursive partitioning
- use some sort of greedy splitting heuristic

Outline

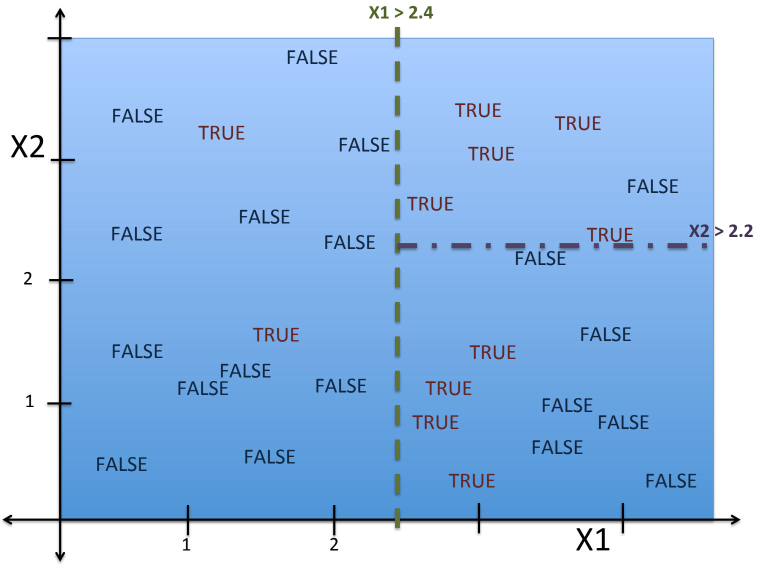
Trees: Recursive Partitioning Intuition (classification)



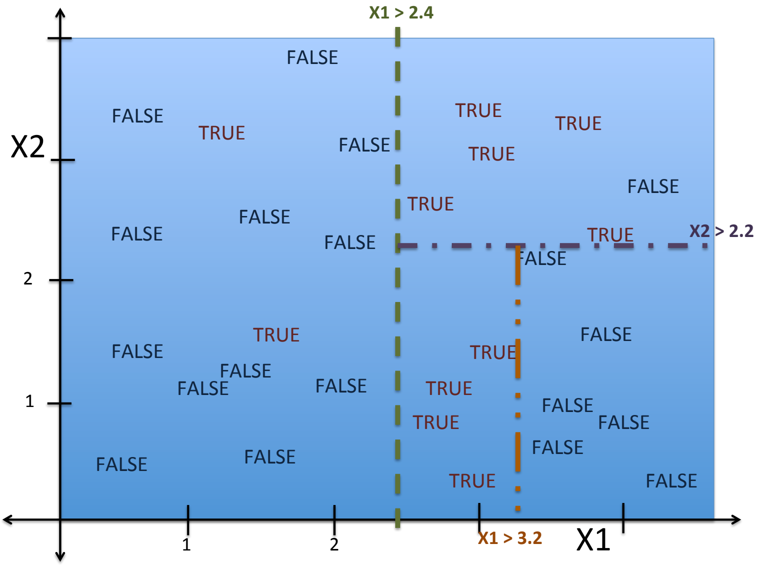
Trees: Recursive Partitioning Intuition (classification)



Trees: Recursive Partitioning Intuition (classification)

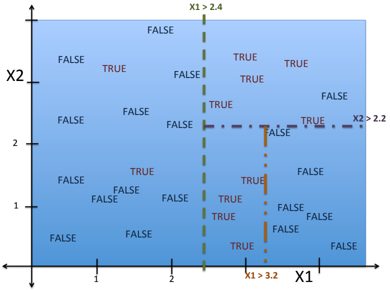


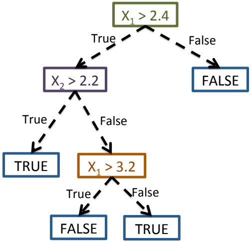
Trees: Recursive Partitioning Intuition (classification)



Trees: Recursive Partitioning Intuition (classification)

How to interpret your tree?
Layer down



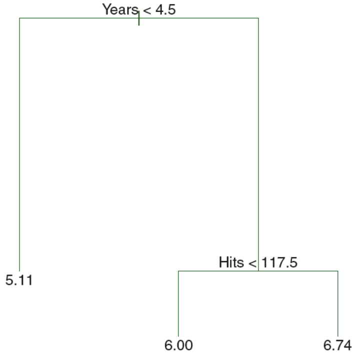


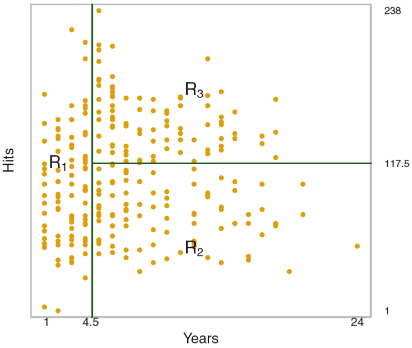
Trees: Recursive Binary Intuition

Recursive Binary Splitting is a *Top-down* and *greedy* approach

- **top-down**: it begins at the top of the tree (at which point all observations belong to a single region) and then successively splits the predictor space; each split is indicated via two new branches further down on the tree.
- **greedy**: at each step of the tree-building process, the best split is made at that particular step, rather than looking ahead and picking a split that will lead to a better tree in some future step.

Trees: Recursive Binary Intuition (regression)





Trees: Recursive Binary Intuition (regression)

- divide the predictor space $X_1, X_2, \ldots, X_p$ into J distinct and non-overlapping regions $R_1, R_2, \ldots, R_J$
- For every observation that falls into the region R_j, we make the same prediction: the mean of the response values for the training observations in R_j

Why Recursive Partitioning?

- Suppose you want to find boxes $R_1, \ldots, R_J$ that minimize

$$\sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2,$$

  where $\hat{y}_{R_j}$ is the mean response for the training observations within the jth box.
- you would have to consider *every possible* partition of the feature space into J boxes!

Trees: Recursive Binary Intuition (regression)

1. Define $\{X|X_j < s\}$ as the region of predictor space in which X_j takes on a value less than s.
2. For any j and s, we define the pair of half-planes

$$R_1(j,s) = \{X|X_j < s\} \quad \text{and} \quad R_2(j,s) = \{X|X_j \geq s\}$$

This iteration will
land us on the
minimizer area

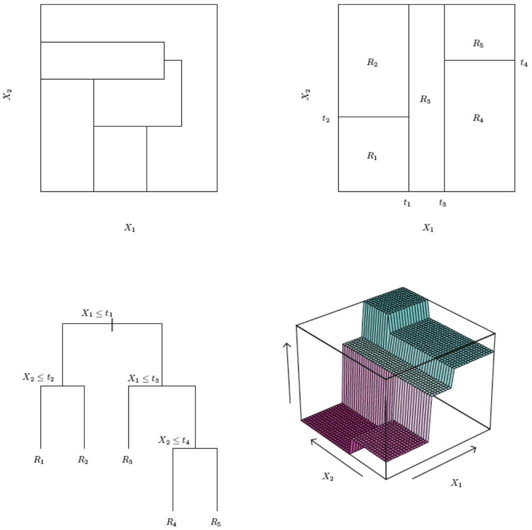
and we seek the value of j and s that minimize

This doesn't seem to have a
unique minimizer, we try
j=1:20
s=s0:lambda:sn
compute loss
order and minimize

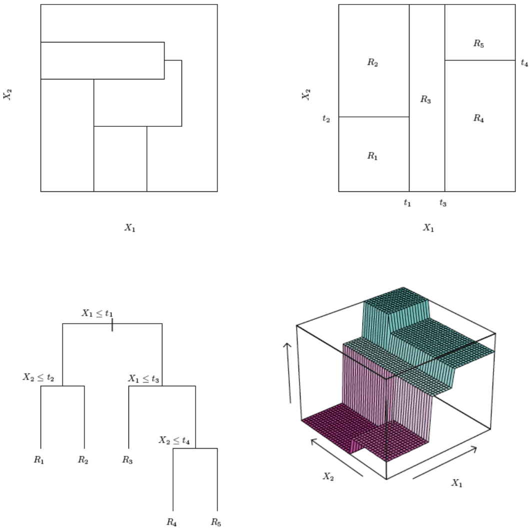
$$\sum_{i:x_i \in R_1(j,s)} (y_i - \hat{y}_{R_1})^2 + \sum_{i:x_i \in R_2(j,s)} (y_i - \hat{y}_{R_2})^2$$

3. We split the two previously identified regions. Again, we look to split these regions further, so as to minimize the RSS within each of the resulting regions.
4. The process continues until a stopping criterion is reached; for instance: continue until no region contains $> m$ observations.
5. Once the regions $R_1, \ldots, R_J$ have been created, we predict the response for a given test observation using the means $\hat{y}_{R_j}$.

Trees: Recursive Binary Intuition (regression)

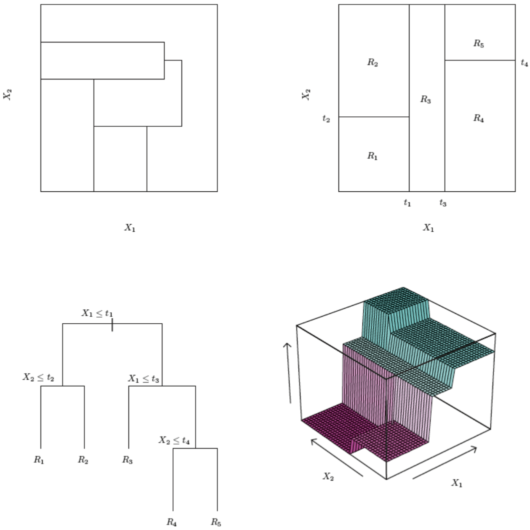


Trees: Recursive Binary Intuition (regression)



What about overfitting?

Trees: Recursive Binary Intuition (regression)



What about overfitting? Answer: pruning (tomorrow)

Trees: Recursive Partitioning

How do we learn a decision tree from training data?

Answer: iterative greedy splitting using a splitting rule based on

- information gain (ID3: Iterative Dichotomiser 3, C4.5)
- Gini impurity coefficient (CART: Classification and Regression Trees)

Note: ID3 and C4.5 are classification algorithms, CART is usable for classification and regression

We will focus on simplest version ID3 with categorical covariates

Trees: Recursive Partitioning

Idea behind ID3 algorithm:

1. Select "best feature" (X_1, X_2, or X_3) to split
2. For each value that feature takes, sort training examples to leaf nodes
3. Stop if all labels in leaf node are the same or all features have been included
4. Assign leaf with majority vote on training examples in leaf
5. Go to 1. if not yet stopped

ID3算法

通过最小化熵选择split变量；

尝试不同的分割点计算分块RSS

选取时的总RSS最小的分割点

结束，进入下一轮循环

Trees: Recursive Partitioning

So what is the "best" feature?

- good if we reduce uncertainty about label with split
- entropy is a measure of uncertainty

Recall:

$$\boxed{H(Y) = -\sum_{y} P(Y = y) \log P(Y = y)}$$

If $\log$ has base 2, $H(Y)$ is the expected number of bits required to encode a randomly drawn value of Y

Trees: Recursive Partitioning

Entropy before split:

$$H(Y) = -\sum_y P(Y = y) \log P(Y = y)$$

Entropy after split based on X_i:

$$\begin{aligned} H(Y \mid X_i) &= \sum_x P(X_i = x) H(Y \mid X_i = x) \\ &= -\sum_x P(X_i = x) \sum_y P(Y = y \mid X_i = x) \log P(Y = y \mid X_i = x) \end{aligned}$$

Information gain (or mutual information) is the difference between the two:

$$I(Y, X_i) = H(Y) - H(Y \mid X_i)$$

Maximize your information gain by minimizing conditional entropy!

$$X_i = \arg\min_{X_i} H(Y \mid X_i)$$

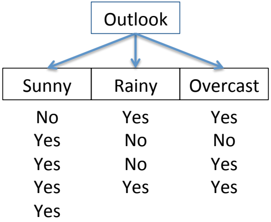
Outline

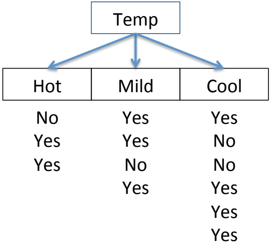
Example (classification)

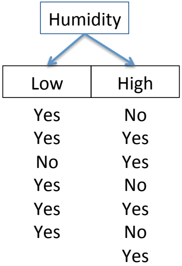
Consider a dataset on playing football:

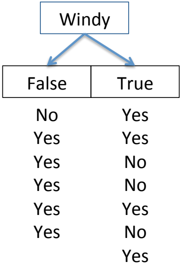
Outlook	**Temp**	**Humidity**	**Windy**	**Play**
Sunny	Hot	High	False	No
Sunny	Hot	Low	True	Yes
Sunny	Mild	Low	True	Yes
Rainy	Mild	High	False	Yes
Overcast	Cool	High	False	Yes
Rainy	Cool	High	True	No
Overcast	Cool	Low	True	No
Sunny	Cool	High	True	Yes
Rainy	Mild	High	True	No
Rainy	Cool	Low	False	Yes
Overcast	Hot	Low	True	Yes
Sunny	Cool	Low	False	Yes
Overcast	Mild	High	False	Yes

Example

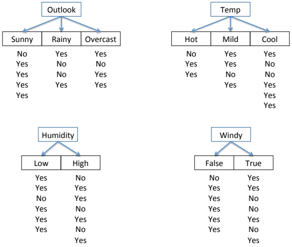

Outlook
Sunny
Rainy
Overcast
No
Yes
Yes
Yes
Yes
Yes
No
No
Yes
Yes
No
Yes
Yes


Temp
Hot
Mild
Cool
No
Yes
Yes
Yes
Yes
No
Yes
Yes
No
No
Yes
Yes
Yes


Humidity
Low
High
Yes
Yes
No
Yes
Yes
Yes
No
Yes
Yes
No
Yes
No
Yes


Windy
False
True
No
Yes
Yes
Yes
Yes
Yes
Yes
Yes
No
No
Yes
No
Yes

Example



Conditional entropy of `Outlook`:

$$H(Y \mid Outlook) = -\sum_{x \in \{S,R,O\}} P(X=x) \sum_{y=0}^{1} P(Y=y \mid X=x) \log P(Y=y \mid X=x)$$

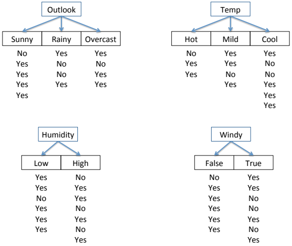
$$P(X=O) = \frac{4}{13}, \quad P(Y=1 \mid O) = \frac{3}{4}, \quad P(Y=0 \mid O) = \frac{1}{4}$$

$$P(X=R) = \frac{4}{13}, \quad P(Y=1 \mid R) = \frac{1}{2}, \quad P(Y=0 \mid R) = \frac{1}{2}$$

$$P(X=S) = \frac{5}{13}, \quad P(Y=1 \mid S) = \frac{4}{5}, \quad P(Y=0 \mid S) = \frac{1}{5}$$

$$\begin{aligned} H(Y \mid Outlook) &= -\left(\frac{4}{13}(\frac{3}{4}\log\frac{3}{4} + \frac{1}{4}\log\frac{1}{4}) + \frac{4}{13}(\frac{1}{2}\log\frac{1}{2} + \frac{1}{2}\log\frac{1}{2}) + \frac{5}{13}(\frac{4}{5}\log\frac{4}{5} + \frac{1}{5}\log\frac{1}{5})\right) \\ &= 0.5788 \end{aligned}$$

Example



Conditional entropy of `Temp`:

$$H(Y \mid Temp) = -\sum_{x \in \{H,M,C\}} P(X=x) \sum_{y=0}^{1} P(Y=y \mid X=x) \log P(Y=y \mid X=x)$$

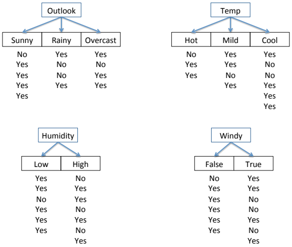
$$P(X=H) = \frac{3}{13}, \quad P(Y=1 \mid H) = \frac{2}{3}, \quad P(Y=0 \mid H) = \frac{1}{3}$$

$$P(X=M) = \frac{4}{13}, \quad P(Y=1 \mid M) = \frac{3}{4}, \quad P(Y=0 \mid M) = \frac{1}{4}$$

$$P(X=C) = \frac{6}{13}, \quad P(Y=1 \mid C) = \frac{2}{3}, \quad P(Y=0 \mid C) = \frac{1}{3}$$

$$\begin{aligned} H(Y \mid Temp) &= -\left(\frac{3}{13}\left(\frac{2}{3}\log\frac{2}{3} + \frac{1}{3}\log\frac{1}{3}\right) + \frac{4}{13}\left(\frac{3}{4}\log\frac{3}{4} + \frac{1}{4}\log\frac{1}{4}\right) + \frac{6}{13}\left(\frac{2}{3}\log\frac{2}{3} + \frac{1}{3}\log\frac{1}{3}\right)\right) \\ &= 0.6137 \end{aligned}$$

Example



Conditional entropy of `Humidity`:

$$H(Y \mid Humidity) = -\sum_{x \in \{H,L\}} P(X=x) \sum_{y=0}^{1} P(Y=y \mid X=x) \log P(Y=y \mid X=x)$$

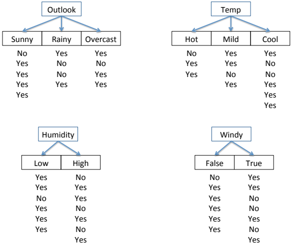
$$P(X=H) = \frac{7}{13}, \quad P(Y=1 \mid H) = \frac{4}{7}, \quad P(Y=0 \mid H) = \frac{3}{7}$$

$$P(X=L) = \frac{6}{13}, \quad P(Y=1 \mid M) = \frac{5}{6}, \quad P(Y=0 \mid M) = \frac{1}{6}$$

$$H(Y \mid Humidity) = -\left(\frac{7}{13}\left(\frac{4}{7}\log\frac{4}{7} + \frac{3}{7}\log\frac{3}{7}\right) + \frac{6}{13}\left(\frac{5}{6}\log\frac{5}{6} + \frac{1}{6}\log\frac{1}{6}\right)\right)$$

$$= 0.5757$$

Example



Conditional entropy of `Windy`:

$$H(Y \mid Windy) = -\sum_{x \in \{T,F\}} P(X = x) \sum_{y=0}^{1} P(Y = y \mid X = x) \log P(Y = y \mid X = x)$$

$$P(X = T) = \frac{7}{13}, \quad P(Y = 1 \mid H) = \frac{4}{7}, \quad P(Y = 0 \mid H) = \frac{3}{7}$$

$$P(X = F) = \frac{6}{13}, \quad P(Y = 1 \mid M) = \frac{5}{6}, \quad P(Y = 0 \mid M) = \frac{1}{6}$$

$$H(Y \mid Windy) = -\left(\frac{7}{13} (\frac{4}{7} \log \frac{4}{7} + \frac{3}{7} \log \frac{3}{7}) + \frac{6}{13} (\frac{5}{6} \log \frac{5}{6} + \frac{1}{6} \log \frac{1}{6}) \right)$$

$$= 0.5757$$

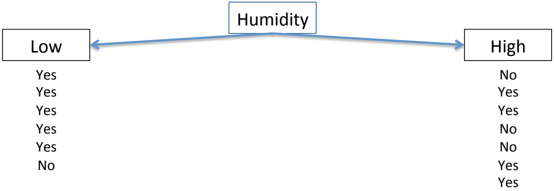
Example

Compare conditional entropies:

Feature	*Conditional Entropy*
Outlook	0.5788
Temp	0.6137
Humidity	**0.5757**
Windy	**0.5757**

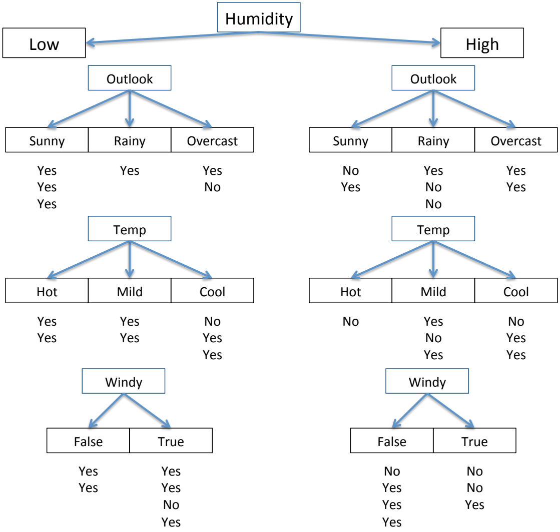
Tie between `Humidity` and `Windy`, randomly select one, say `Humidity`.

Example

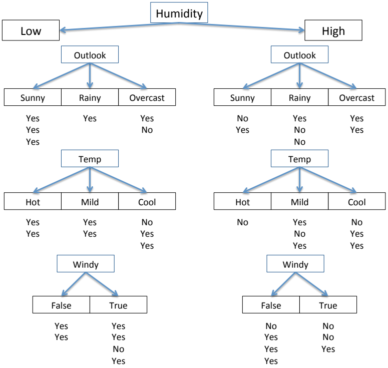

Humidity
Low
High
Yes
Yes
Yes
Yes
Yes
No
No
Yes
Yes
No
No
Yes
Yes

Example

Let's add in the next feature:

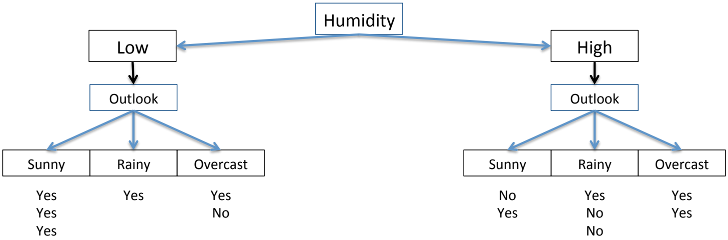


Example



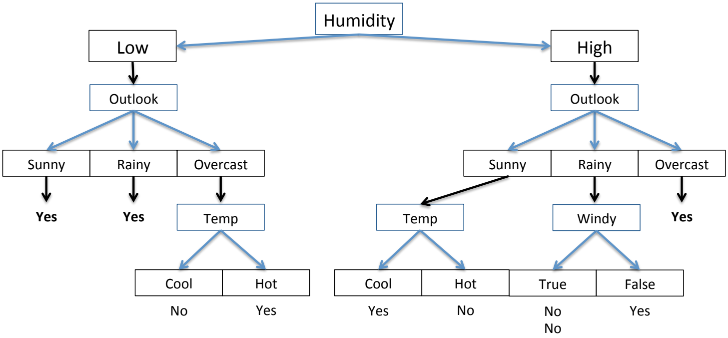
- For the low humidity side, compute conditional entropy
- For the high humidity side, compute conditional entropy
- Divide each side based on feature that minimizes its conditional entropy

Example



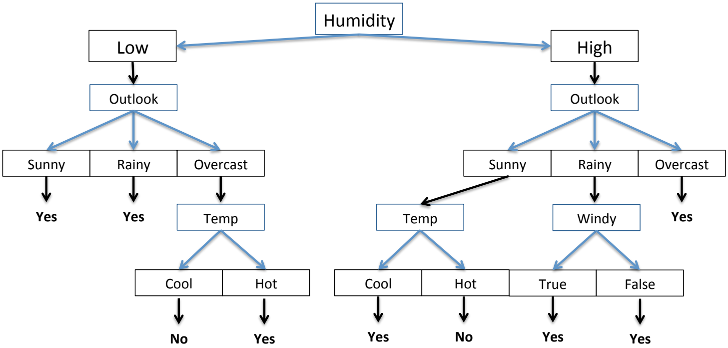
Humidity
Low
High
Outlook
Outlook
Sunny
Rainy
Overcast
Sunny
Rainy
Overcast
Yes
Yes
Yes
Yes
Yes
No
No
Yes
Yes
No
No
Yes
Yes

Example


Humidity
Low
High
Outlook
Outlook
Sunny
Rainy
Overcast
Sunny
Rainy
Overcast
Yes
Yes
Temp
Temp
Windy
Yes
Cool
Hot
Cool
Hot
True
False
No
Yes
Yes
No
No
No
Yes

Example

The Final Result



Outline

Continuous Covariates

But what happens when covariates are continuous?

- can still use entropy
- search over which feature to split
- if we split feature j at x', we get conditional entropy

$$\begin{aligned}
&H(Y \mid X_j \text{ split at } x') \\
&\quad = -P(X_j > x') \sum_y P(Y = y \mid X_j > x') \log P(Y = y \mid X_j > x') \\
&\qquad - P(X_j \leq x') \sum_y P(Y = y \mid X_j \leq x') \log P(Y = y \mid X_j \leq x')
\end{aligned}$$

- find conditional entropy for feature j by

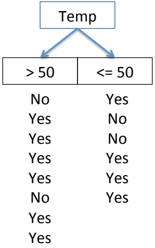
$$H(Y \mid X_j) = \min_{x'} H(Y \mid X_j \text{ split at } x')$$

Example

Use numeric value for temperature:

Outlook	**Temp**	**Humidity**	**Windy**	**Play**
Sunny	97	High	False	No
Sunny	85	Low	True	Yes
Sunny	71	Low	True	Yes
Rainy	75	High	False	Yes
Overcast	56	High	False	Yes
Rainy	42	High	True	No
Overcast	34	Low	True	No
Sunny	44	High	True	Yes
Rainy	64	High	True	No
Rainy	49	Low	False	Yes
Overcast	88	Low	True	Yes
Sunny	47	Low	False	Yes
Overcast	69	High	False	Yes

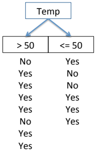
Example



Fix a value for x', say 50:

- find which records have values $>$ 50 and which have $\leq$ 50
- compute conditional entropy

Example



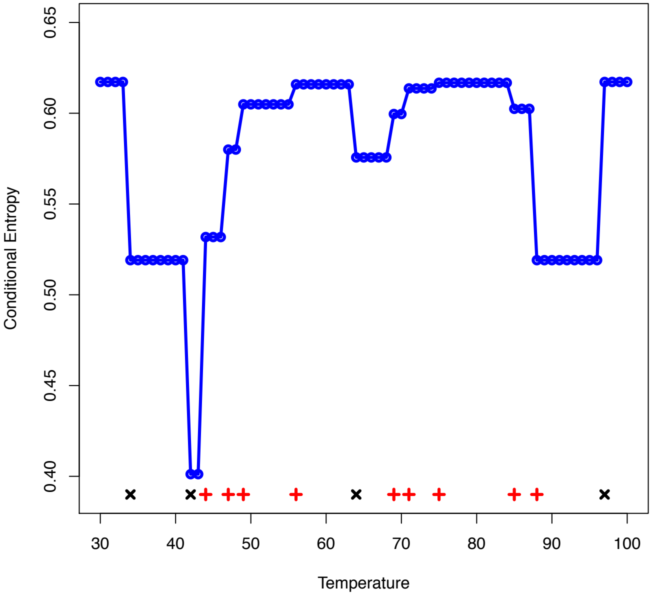
$$
\begin{aligned}
H(Y \mid x' = 50) &= -P(X > x') \sum_{y=0}^{1} P(Y = y \mid X > x') \log P(Y = y \mid X > x') \\
&\quad - P(X \le x') \sum_{y=0}^{1} P(Y = y \mid X \le x') \log P(Y = y \mid X \le x') \\
P(X > 50) &= \frac{8}{13}, \quad P(Y = 1 \mid > 50) = \frac{6}{8}, \quad P(Y = 0 \mid > 50) = \frac{2}{8} \\
P(X \le 50) &= \frac{6}{13}, \quad P(Y = 1 \mid \le 50) = \frac{4}{6}, \quad P(Y = 0 \mid \le 50) = \frac{2}{6} \\
H(Y \mid x' = 50) &= -\left(\frac{8}{13} \left(\frac{3}{4} \log \frac{3}{4} + \frac{1}{4} \log \frac{1}{4} \right) + \frac{6}{13} \left(\frac{2}{3} \log \frac{2}{3} + \frac{1}{3} \log \frac{1}{3} \right) \right) \\
&= 0.6168
\end{aligned}
$$

Example

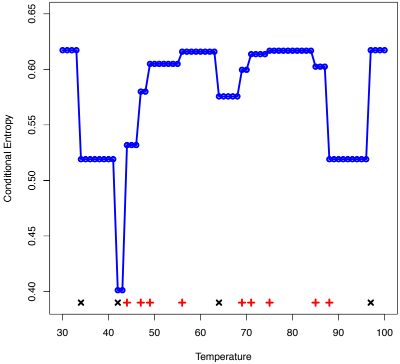
So how do we find best x'?

- do this for all values in range of X_j
- only need to search over seen values, since entropy constant in between

Example


Conditional Entropy
0.65
0.60
0.55
0.50
0.45
0.40
30
40
50
60
70
80
90
100
Temperature

Example



- conditional entropy in minimized when $42 \le x' \le 43$
- in this range, $H(Y \,|\, Temp) = 0.4012$
- for comparison,

Feature	*Conditional Entropy*
Outlook	0.5788
Temp	0.6137
Humidity	**0.5757**
Windy	**0.5757**

Outline

Regression

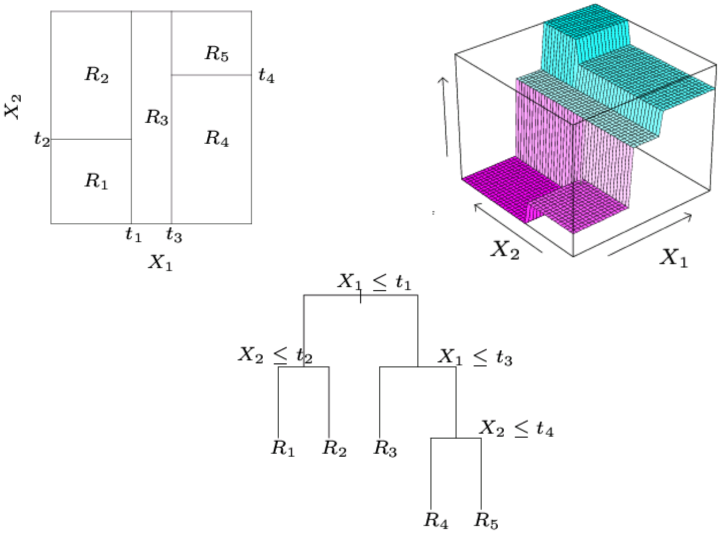
But what happens when we want to do regression?

- often mean squared error is minimized
- choose attribute which maximizes

$$\begin{aligned} X_j &= \arg\max_j MSE(Y) - MSE(Y \mid X_j) \\ &= \arg\min_j MSE(Y \mid X_j) \end{aligned}$$

- in leaf nodes, assign average value rather than majority label

Regression



Outline

Overfitting

What happens when we keep expanding the tree?

Well, let's listen to some sports announcers:

- "This team is 3 and 5 on Tuesdays when its lightly raining" (Susan Waldman WCBS Yankees radio announcer)
- "Since Joe Morgan joined the Cincinnati Reds in 1972, he is batting 385 against Bob Gibson, but while he was with the Astros he only hit 176" (Tony Kubeck on NBC Game of the Week)

Overfitting

Political pundits do it as well:

- "Both BuyCostumes.com and Spirit Halloween Store claim the number of Mitt Romney or Barack Obama masks sold this year will predict which man wins the election this November. BuyCostumes mask sales have correctly predicted the victor in the past three elections, while Spirit's mask sales have been right in the past four" (ABC News)
- "If the Washington Redskins win their last home game before the election, the incumbent party stays in power... The Redskins game has correctly predicted 17 of the last 18 presidential elections" (ABC News)

Overfitting

Basic tree fitting algorithm:

1. Select "best feature" to split (along "best split")
2. For each value of feature split, sort training examples to leaf nodes
3. Decide whether to stop
4. Assign leaf with majority vote (classification) or average (regression)
5. Go to 1. if not yet stopped

How can we reduce overfitting?

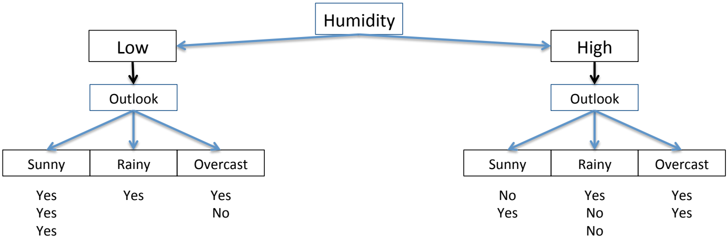
- pick good stopping rules
- "prune" the tree

Stopping Rules

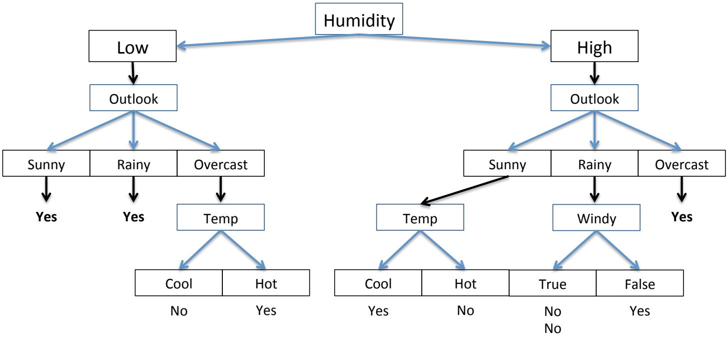
Stopping rules tell you when to stop dividing the tree

- control algorithm complexity
- control overfitting (to an extent)
- types of rules:
  - minimum number of observations (usually $> C\log(n)$)
  - threshold on information gain (if gain $< C$, stop)

Reconsider This Example: Stopping

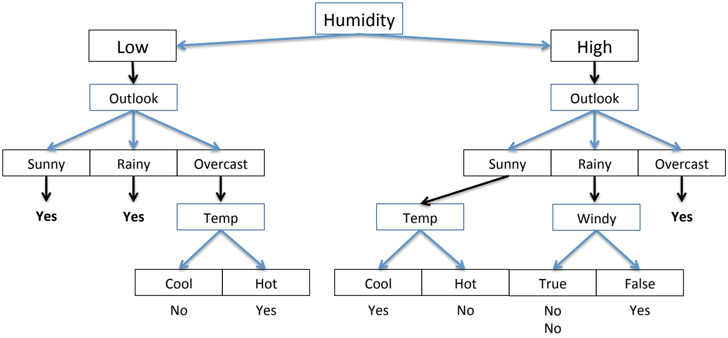


Reconsider This Example: Stopping

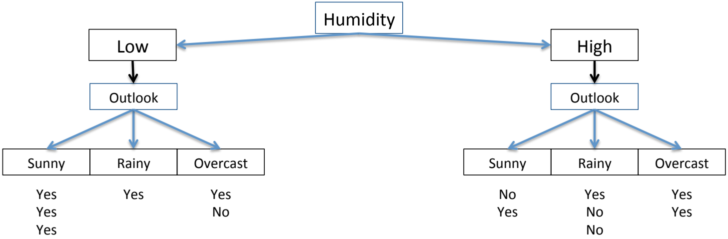


Reconsider This Example: Stopping

(What if our stopping rule is: `branch if numel > 3`)



Reconsider This Example: Stopping



Pruning

Most overfitting is reduced is by *pruning*

- idea: grow a tree that is too large and then trim back portions that "don't add much"
- it is easier to find a good fit through pruning than through aggressive stopping rules
- pruning is a way to express preference for simpler models over more complicated ones, not necessarily finding a better fit
- many ways to prune:
  - cross validation
  - error complexity (cost complexity)
  - etc

Pruning

Cost complexity pruning: add penalty for tree size

- fully expand tree
- $|T|$ is the number of terminal nodes
- want to find subtree that minimizes $Cost(\alpha)$ for a fixed α

$$Cost(\alpha) = \sum_{m=1}^{|T|} \sum_{x_i \in R_m} (y_i - \hat{y}_{R_m})^2 + \alpha|T|$$

(R_m is the set of points in the mth terminal node)

Pruning

Example: cost complexity with cross validation

1. get set of tuning parameters, $\alpha_1, \ldots, \alpha_M$
2. divide data into K folds
3. for each fold $k = 1, \ldots, K$:
    - fully expand tree for stopping rules
    - prune to find MSE on validation data as a function of α_m
4. select α_m with lowest validation MSE
5. fit tree on full data with that α_m

Outline

Trees in R

Many packages for trees in R, but 2 most useful are:

- rpart
- tree

```
> library(rpart)
> cmb <- read.csv("cmb.csv")
> names(cmb)
[1] "ell" "Cl"
> attach(cmb)
> plot(ell,Cl)
```

Trees in R

Use the function `rpart( )` to fit a tree; it takes care of all of the fitting and pruning for you

```
> tree.cmb <- rpart(Cl ~ ell)
> names(tree.cmb)
> cmb.fit <- predict(tree.cmb,cmb,type="vector")
> lines(ell,cmb.fit,col="red",lwd=3)
```

Note: `tree.cmb` is `rpart` object, `cmb` is dataframe

Trees in R

`rpart( )` also has some nice display options to show the tree

```
> plot(tree.cmb)
> text(tree.cmb,use.n=TRUE)
> # use.n displays the number in each leaf
```

(Let's spend some time digesting this)

Trees in R

And now let's use `rpart( )` for some classification:

- want to predict whether kyphosis, a type of deformation, was present or absent during spinal surgery in children
- covariates: age (in months), number (number of vertebrae involved), start (topmost vertebra operated on)
- part of `rpart` library

```
> fit <- rpart(Kyphosis ~ Age + Number + Start, data=kyphosis)
> # Default uses Gini split
> fit2 <- rpart(Kyphosis ~ Age + Number + Start, data=kyphosis,
parms=list(split='information'))
> # Now use an information gain split
> par(mfrow=c(1,2))
> plot(fit)
> text(fit, use.n=TRUE)
> plot(fit2)
> text(fit2, use.n=TRUE)
```