

Machine Learning (CS 567) Lecture 7

Fall 2008

Time: T-Th 5:00pm - 6:20pm

Location: GFS 118

Instructor: Sofus A. Macskassy (macskass@usc.edu)

Office: SAL 216

Office hours: by appointment

Teaching assistant: Cheol Han (cheolhan@usc.edu)

Office: SAL 229

Office hours: M 2-3pm, W 11-12

Class web page:

<http://www-scf.usc.edu/~csci567/index.html>

Administrative - Projects

- You should all be in a team by now.
- By next class, one of the team-members should send me and Cheol Han a list of the team-members and your pre-proposal.
- Pre-proposals deliverable, 1-2 paragraphs:
 - Team members
 - Problem domain
 - Learners you are considering
 - Where you think you will get your data
 - How you think you will evaluate your results

Lecture 7 Outline

- Decision Trees, Part 2

Choosing the Best Attribute (Method 2)

- Choose the attribute x_j that has the highest mutual information with y .

$$\begin{aligned}\operatorname{argmax}_j I(x_j; y) &= H(y) - \sum_v P(x_j = v) H(y|x_j = v) \\ &= \operatorname{argmin}_j \sum_v P(x_j = v) H(y|x_j = v)\end{aligned}$$

- Define $\tilde{J}(j)$ to be the expected remaining uncertainty about y after testing x_j

$$\tilde{J}(j) = \sum_v P(x_j = v) H(y|x_j = v)$$

Choosing the Best Attribute (Method 2)

ChooseBestAttribute(S)

choose j to minimize \tilde{J}_j , computed as follows:

for $v \in \{0, 1\}$ do

$S_v :=$ all $\langle \mathbf{x}, y \rangle \in S$ with $x_j = v$;

$p_v := |S_v|/|S|$;

$n_v := |S_v|$;

$n_{v,y} :=$ number of examples in S_v with class y ;

$p_{v,y} := n_{v,y}/n_v$ probability of examples from class y in S_v ;

$H(y|x_j = v) := -\sum_y p_{v,y} \log p_{v,y}$;

done

$\tilde{J}_j := p_0 H(y|x_j = 0) + p_1 H(y|x_j = 1)$

return j

Non-Boolean Features

- Multiple discrete values
 - Method 1: Construct multiway split
 - Method 2: Test for one value versus all of the others
 - Method 3: Group the values into two disjoint sets and test one set against the other
- Real-valued variables
 - Test the variable against a threshold
- In all cases, mutual information can be computed to choose the best split

Efficient Algorithm for Real-Valued Features

- To compute the best threshold θ_j for attribute j
 - Sort the examples according to x_{ij} .
 - Let θ be the smallest observed x_{ij} value
 - Let $n_{0L} := 0$ and $n_{1L} := 0$ be the number of examples from class $y=0$ and $y=1$ such that $x_{ij} < \theta$
 - Let $n_{0R} := N_0$ and $n_{1R} := N_1$ be the number of examples from class $y=0$ and $y=1$ such that $x_{ij} \geq \theta$
 - Increase θ
 - Let y_i be the class of the next instance
 - if $y_i = 0$, then $n_{0L}++$ and $n_{0R}--$
 - else $n_{1L}++$ and $n_{1R}--$
 - Compute $J(\theta)$ from n_{0L} , n_{1L} , n_{0R} , and n_{1R} .
 - Remember the smallest value of J and the corresponding θ

Real-Valued Features

- Mutual information of $\theta = 1.2$ is 0.2294

y_i	0	0	1	0		1	1	0	1	1
x_{ij}	0.2	0.4	0.7	1.1		1.3	1.7	1.9	2.4	2.9

$n_{0,L} = 3$	$n_{0,R} = 1$
$n_{1,L} = 1$	$n_{1,R} = 4$

- Mutual information only needs to be computed at points between examples from different classes

Handling Missing Values: Proportional Distribution

- Attach a weight w_i to each example (\mathbf{x}_i, y_i) .
 - At the root of the tree, all examples have a weight of 1.0
- Modify all mutual information computations to use weights instead of counts
- When considering a test on attribute j , only consider those examples for which x_{ij} is not missing
- When splitting the examples on attribute j :
 - Let p_L be the probability that a non-missing example is sent to the left child and p_R be the probability that it is sent to the right child
 - For each example (\mathbf{x}_i, y_i) that is missing attribute j , send it to both children. Send it to the left child with weight $w_i := w_i \cdot p_L$ and to the right child with weight $w_i := w_i \cdot p_R$
- When classifying an example that is missing attribute j :
 - Send it down the left subtree. Let $P(\hat{y}_L | \mathbf{x})$ be the resulting prediction
 - Send it down the right subtree. Let $P(\hat{y}_R | \mathbf{x})$ be the resulting prediction
 - Return $p_L \cdot P(\hat{y}_L | \mathbf{x}) + p_R \cdot P(\hat{y}_R | \mathbf{x})$

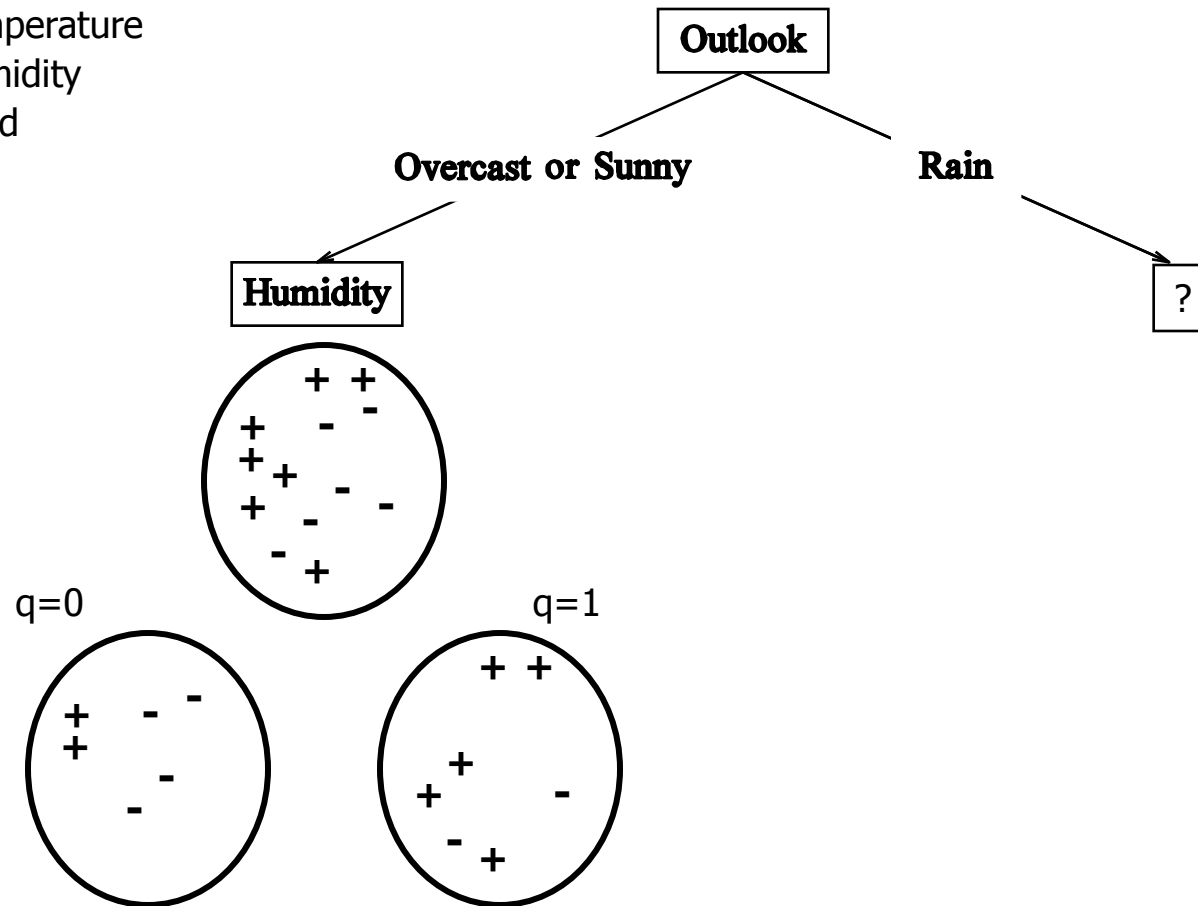
Handling Missing Values: Surrogate Splits

- Choose an attribute j and a splitting threshold θ_j using all examples for which x_{ij} is not missing
 - Let u_i be a variable that is 0 if (\mathbf{x}_i, y_i) is sent to the left subtree and 1 if (\mathbf{x}_i, y_i) is sent to the right subtree
 - For each remaining attribute q , find the splitting threshold θ_q that best predicts u_i . Sort these by their predictive power and store them in node x_j of the decision tree
- When classifying a new data point (\mathbf{x}, y) that is missing x_j , go through the list of surrogate splits until one (x_q) is found that is not missing in \mathbf{x} . Use x_q and θ_q to decide which child to send \mathbf{x} to.

Handling Missing Values: Surrogate Splits

Features:

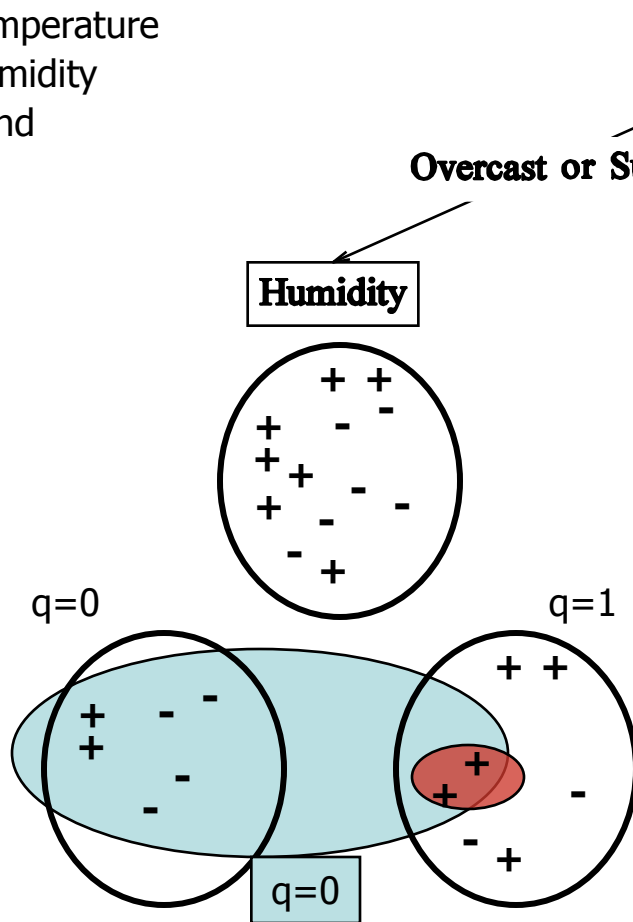
Outlook
Temperature
Humidity
Wind



Handling Missing Values: Surrogate Splits

Attribute:

- Outlook
- Temperature
- Humidity
- Wind

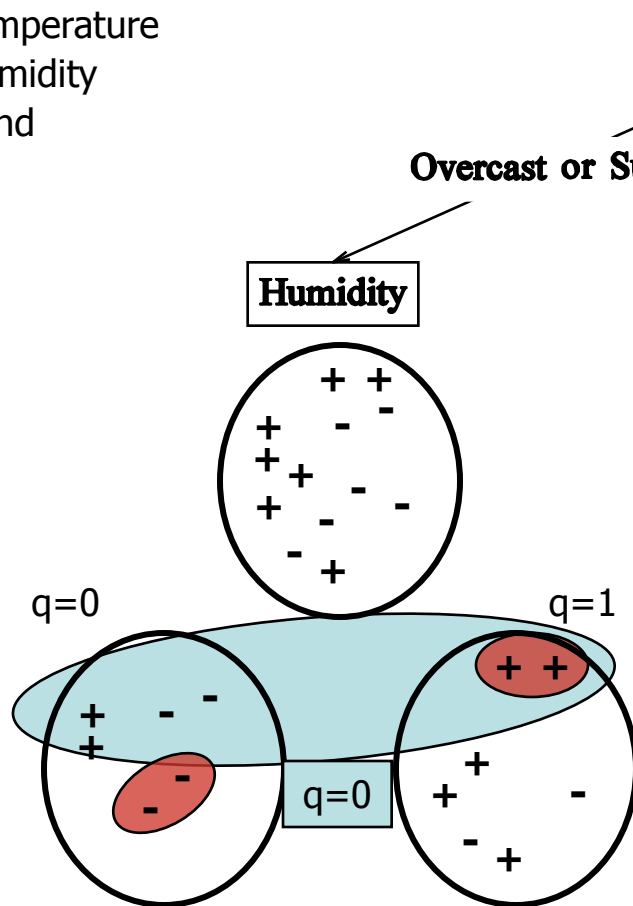


Attribute	Accuracy
Outlook	11/13

Handling Missing Values: Surrogate Splits

Attribute:

- Outlook
- Temperature
- Humidity
- Wind

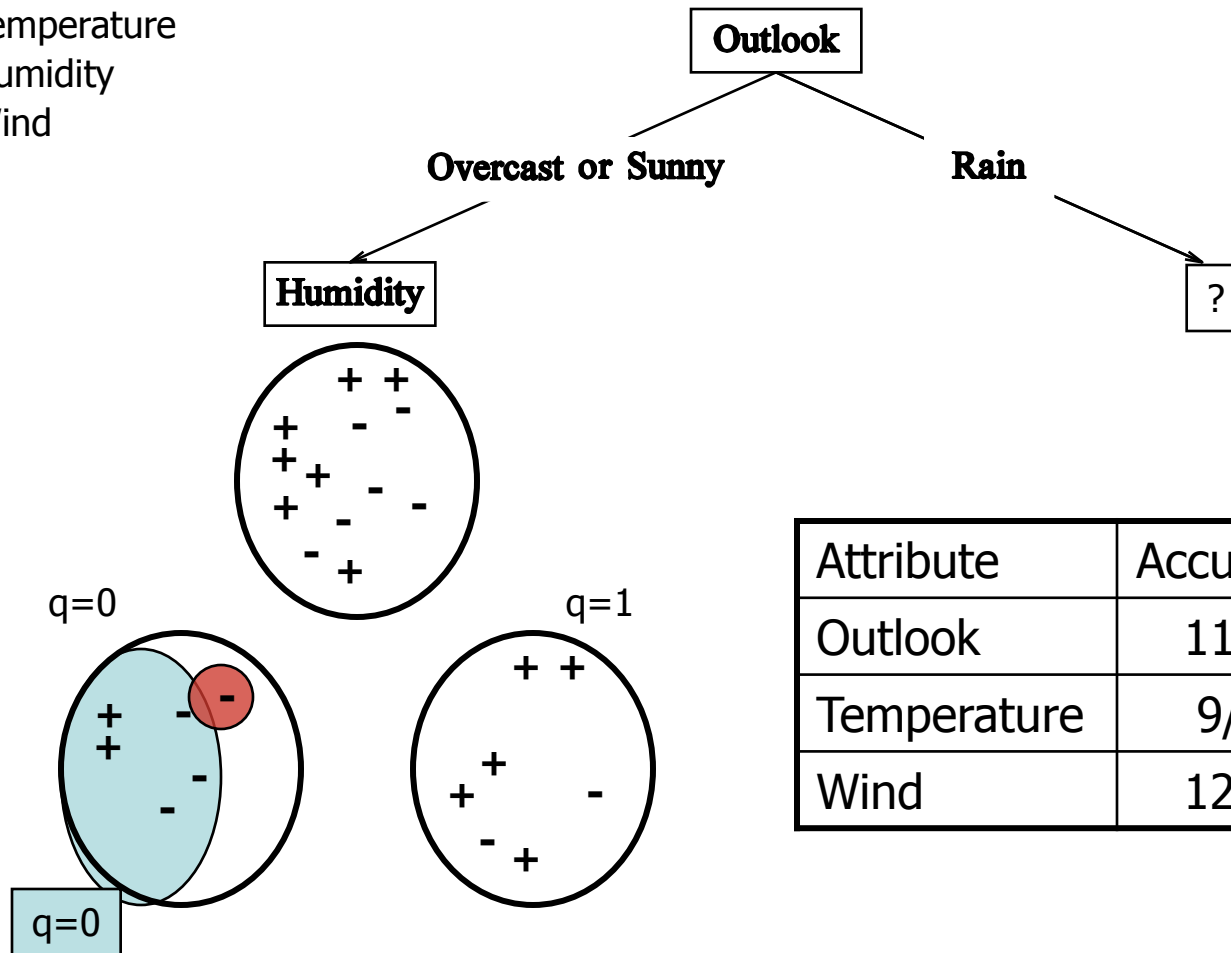


Attribute	Accuracy
Outlook	11/13
Temperature	9/13

Handling Missing Values: Surrogate Splits

Attribute:

- Outlook
- Temperature
- Humidity
- Wind

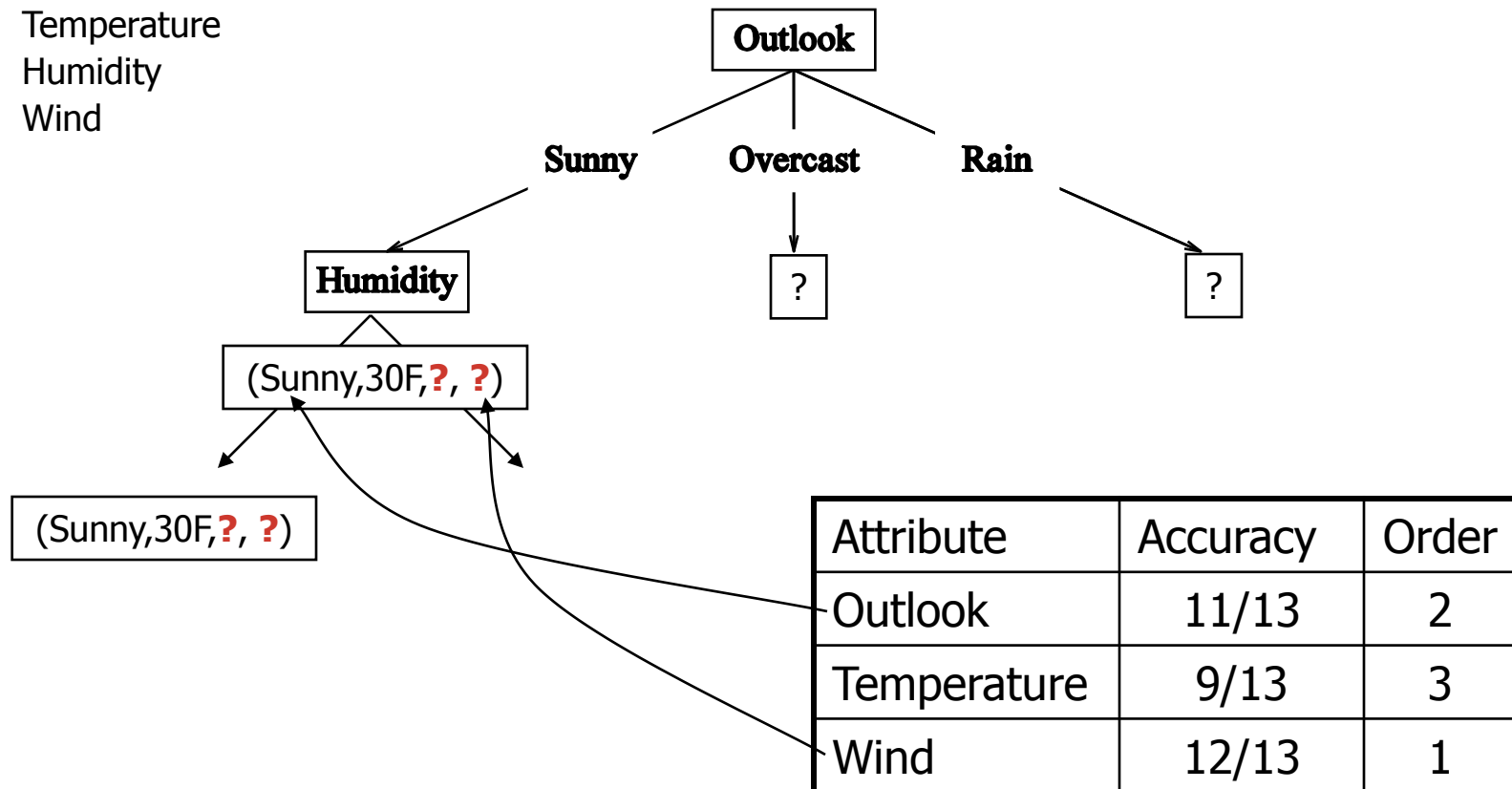


Attribute	Accuracy
Outlook	11/13
Temperature	9/13
Wind	12/13

Handling Missing Values: Surrogate Splits (Classifying new instance)

Attribute:

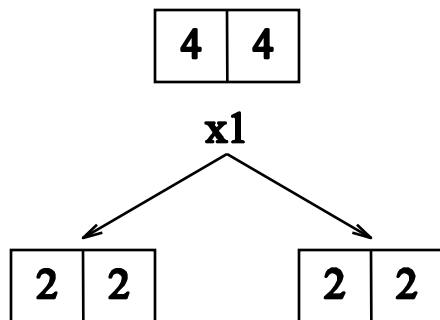
Outlook
Temperature
Humidity
Wind



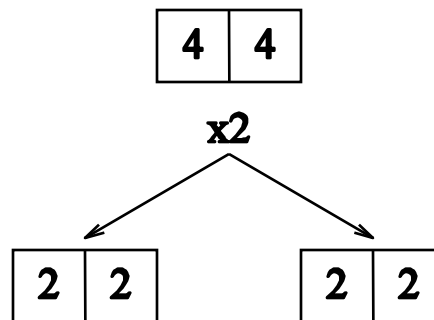
Issue(1): Failure of Greedy Approximation

- Greedy heuristics cannot distinguish random noise from XOR

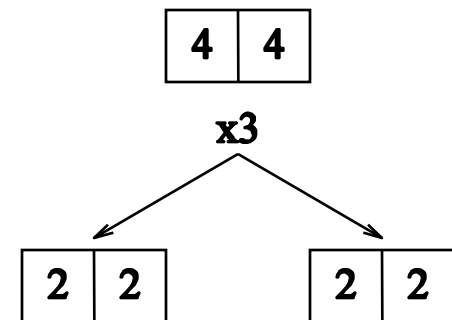
x_1	x_2	x_3	y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0



J=4



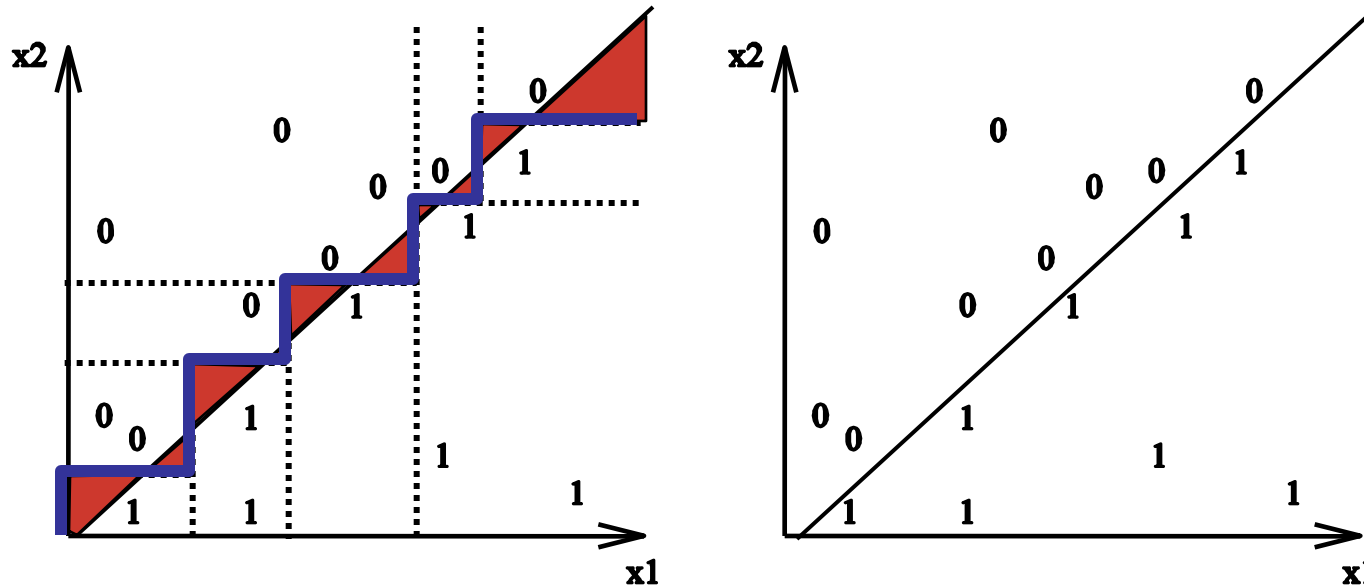
J=4



J=4

Issues (2): Accuracy

- Axis-aligned... not always the right model:



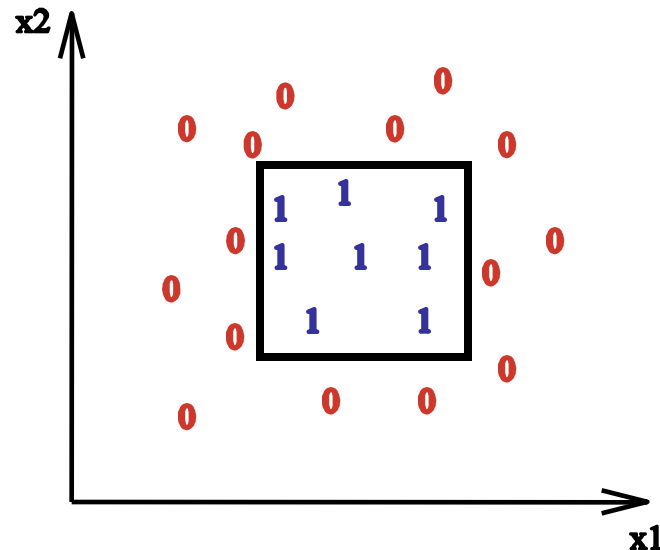
- Finding progressive splits is not guaranteed.
- Does not generalize very well.

Issues (3): Missing Values

- Missing Values: Why does “ignore attribute” work?
 - Proportional split
 - Problem: assumes independence among attributes.
 - Infer value (from other attributes in instance)
 - Problem: if we can infer value from other attributes, then this attribute is not informative, so why use it in the model?
 - Ignore attribute (learn model without it)
 - Problem: If the attribute is actually important to the concept, then we lose valuable information.

Issue(4): Revisit Attributes?

- Revisit attributes—can we avoid revisiting attributes by making “better” splits?



Issue(5): Split on multiple attributes

- Can decision split points be on multiple attributes?
- Sure, but the search space becomes so large that it quickly becomes intractable
- For example, consider n boolean features
 - For groups of k attributes, we have 2^k possible decision tests.
 - We have k -choose- n groups of k .
- So, for each node in the tree, we need to test:
 - For pairs, we get: $O(2^2 * n^2)$ possible decisions to test.
 - For triples, we get: $O(2^3 * n^3)$ possible decisions to test.
 - Each decision takes $O(m)$ to calculate $J(A)$, m =number of data points in the data set at that node.

Decision Tree Summary

- Hypothesis Space
 - variable size (contains all functions)
 - deterministic
 - discrete and continuous parameters
- Search Algorithm
 - constructive search
 - eager
 - batch

Summary so far

Criterion	Perc	Logistic	LDA	Trees
Mixed data	no	no	no	yes
Missing values	no	no	yes	yes
Outliers	no	yes	no	yes
Monotone transformations	no	no	no	yes
Scalability	yes	yes	yes	yes
Irrelevant inputs	no	no	no	somewhat
Linear combinations	yes	yes	yes	no
Interpretable	yes	yes	yes	yes
Accurate	yes	yes	yes	no