Lecture 3: Evaluating ML Systems and Decision Tree

Richa Singh

Google classroom code: wgzuohn

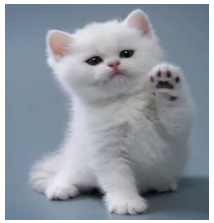
Slides are prepared from several information sources on the web and books

Recap of Lecture 2

- ROC
- CMC
- No free lunch
- Decision Trees

Recap: CMC

Kitten A



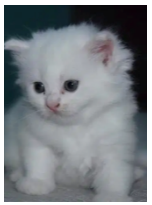
Kitten B

Kitten C

And four test queries:

1

2



3

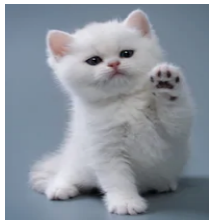
4

Query	Top result	Result 2	Result 3
1	A	B	C
2	B	C	A
3	A	B	C
4	C	B	A

What are the rank accuracies?

Recap: CMC


Kitten A

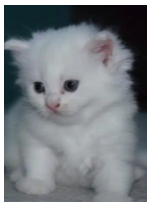

Kitten B


Kitten C

And four test queries:


1


2


3


4

Query	Top result	Result 2	Result 3
1	A	B	C
2	B	C	A
3	A	B	C
4	C	B	A

What are the rank accuracies?

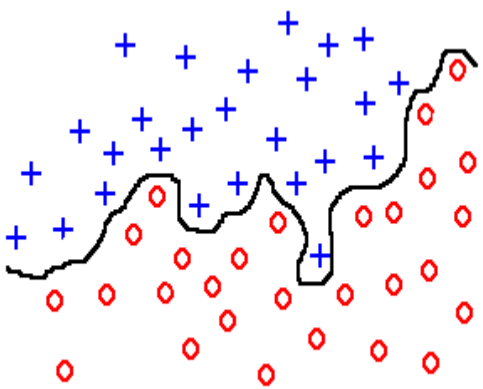
Rank 1: 1/4 predicted correctly: 25%
Rank 2: 3/4 : 75%
Rank 3: 4/4 : 100%

Evaluating ML Systems

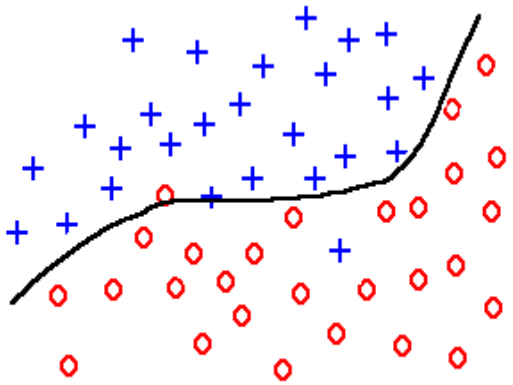
- Assumption: Building a model for population
- Reality: Population is not available
- We work with a sample database − not necessarily true representation of the population

- What to do?
  - Should we use the entire available database for training the model?
    - High accuracy on the training data
    - Lower accuracy on the testing data
    - Called as overfitting

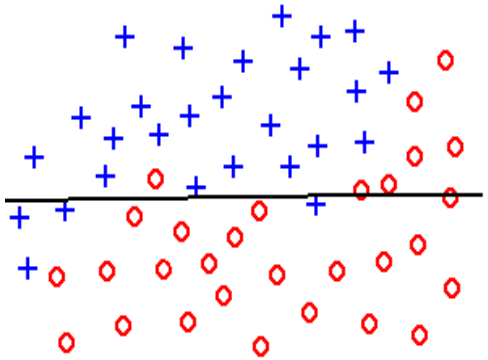
Evaluating ML Systems



Overfitting



Good fit

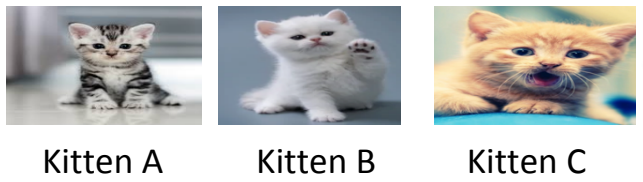


Underfitting

- Underfitting: Learning algorithm had the opportunity to learn more from training data, but didn't
- Overfitting: Learning algorithm paid too much attention to idiosyncracies of the training data; the resulting tree doesn't generalize

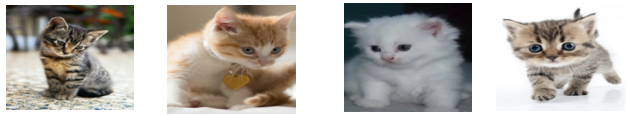
Cross Validation

- "Cross-Validation is a statistical method of evaluating and comparing learning algorithms."
- The data is divided into two parts:
  - Training: to learn or train a model
  - Testing: to validate the model



Kitten A Kitten B Kitten C

Training database



1 2 3 4

Testing database

Cross Validation

- It is used for
  - Performance evaluation: Evaluate the performance of a classifier using the given data
  - Model Selection: Compare the performance of two or more algorithms (DT classifier and neural network) to determine the best algorithm for the given data
  - Tuning model parameters: Compare the performance of two variants of a parametric model

Type of Cross Validation

- Resubstitution Validation
- Hold-Out Validation
- K-Fold Cross-Validation
- Leave-One-Out Cross-Validation
- Repeated K-Fold Cross-Validation

Type of Cross Validation…

- Resubstitution Validation
  - All the available data is used for training and the same data is used for testing
    - Does not provide any information about generalizability

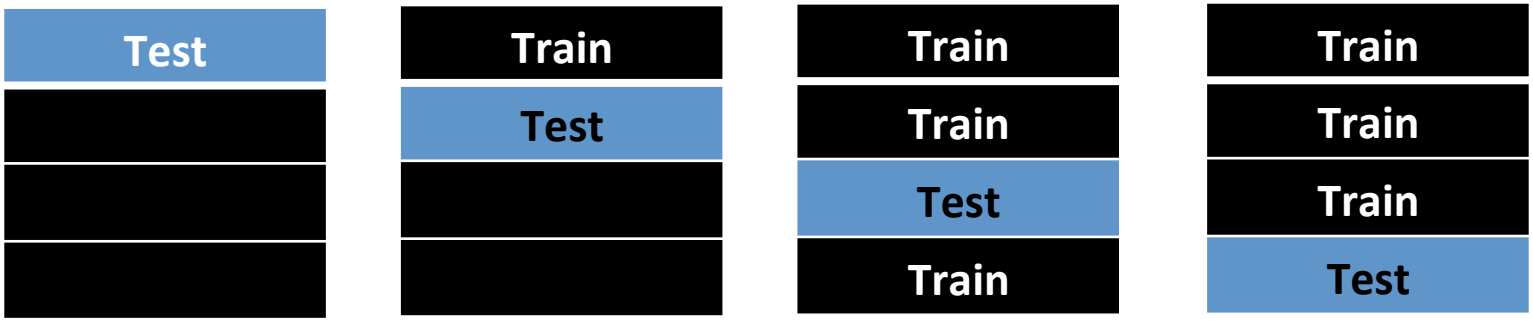
Type of Cross Validation…

- Hold-Out Validation
  - The database is partitioned into two non-overlapping parts, one for training and other for testing
    - The results depend a lot on the partition, may be skewed if the test set is too easy or too difficult

Type of Cross Validation…

- K-Fold Cross-Validation
  - Data is partitioned into k equal folds (partitions). k-1 folds are used for training and 1 fold for testing
  - The procedure is repeated k times

- Across multiple folds, report:
  - Average error or accuracy
  - Standard deviation or variance



4-fold cross validation

Type of Cross Validation…

- Repeated K-Fold Cross-Validation
  - Repeat k-fold cross validation multiple times
- Leave-One-Out Cross-Validation
  - Special case of k-fold cross validation where k=number of instances in the data
  - Testing is performed on a single instance and the remaining are used for training

- Across multiple folds, report:
  - Average error or accuracy
  - Standard deviation or variance

Comparing Cross-Validation Methods

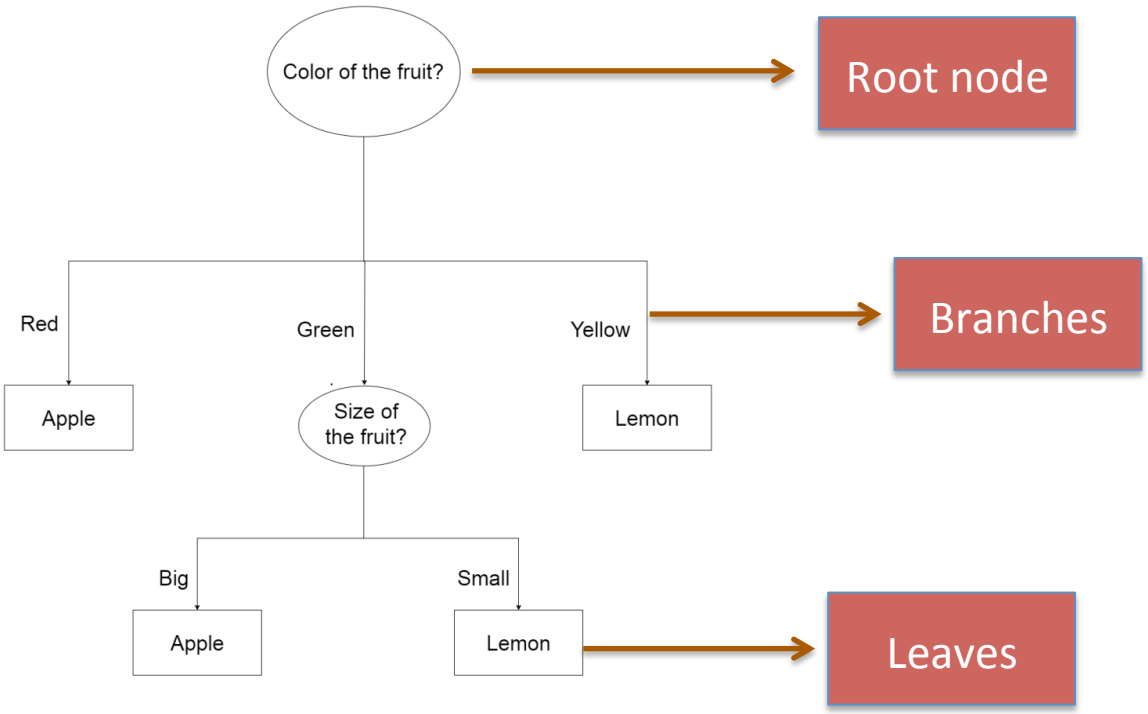
Validation Method	Advantages	Disadvantages
Resubstitution	Simple	Overfitting
Hold-out validation	Independent training and testing sets	Reduced data for training and testing
K-fold cross validation	Accurate performance estimation	Small sample for performance estimation, underestimated performance variance or overestimated degree of freedom for comparison
Leave-one-out cross validation	Unbiased performance estimation	Very large variance
Repeated k-fold cross-validation	Large number of performance estimates	Overlapped training and test data between each round, underestimated performance variance or overestimated degree of freedom for comparison

Questions?

Solve the no free lunch trivia shared on WebEx

Decision Trees

Decision Tree



Each internal node: tests one feature Xi

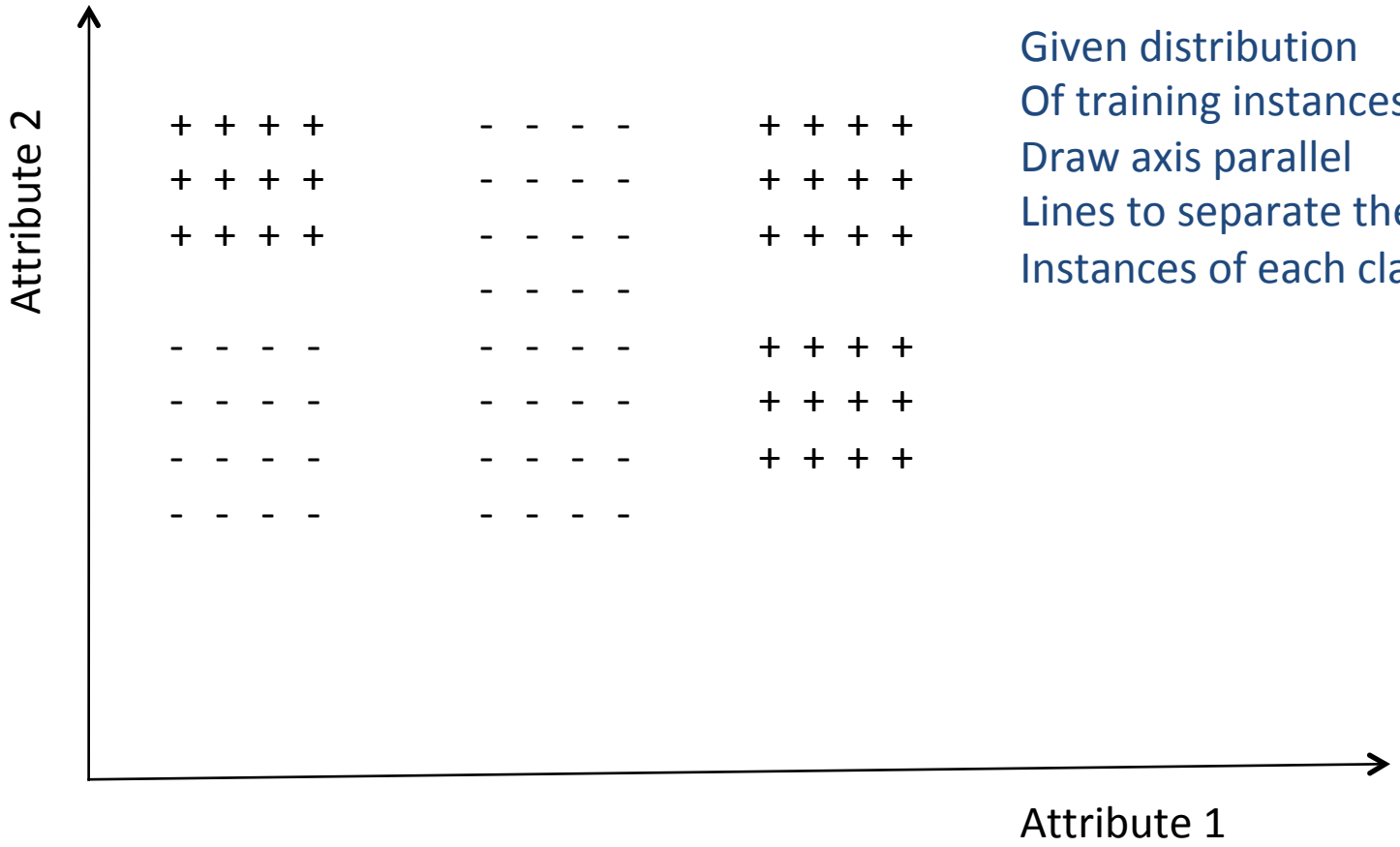
Each branch from a node: selects one value for Xi

Each leaf node: prediction for Y

Features can be discrete, continuous or categorical

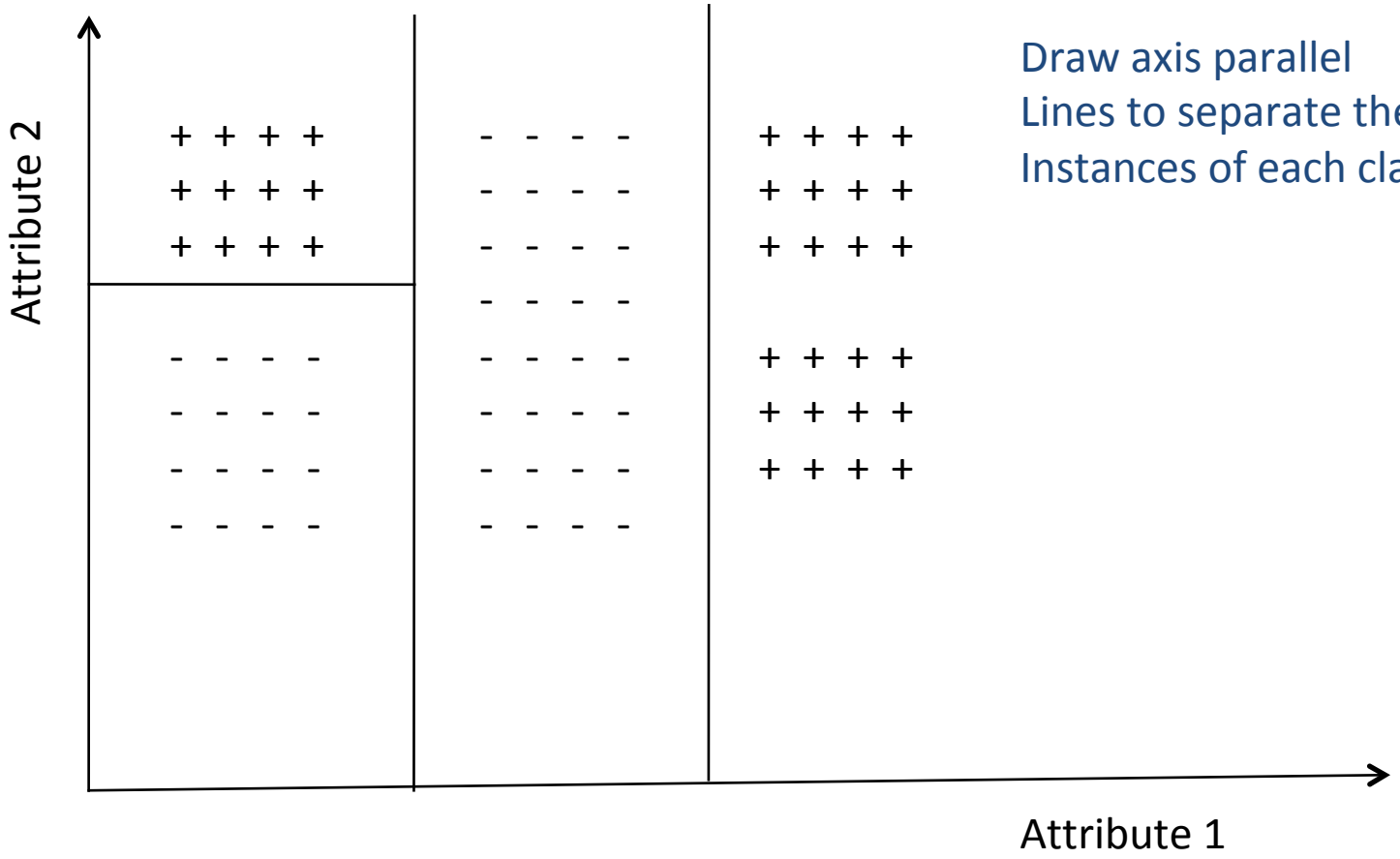
Images from https://machinelearning.school.blog/2017/01/12/introduction-to-decision-trees/

Decision Trees



Given distribution
Of training instances
Draw axis parallel
Lines to separate the
Instances of each class

Decision Tree Structure

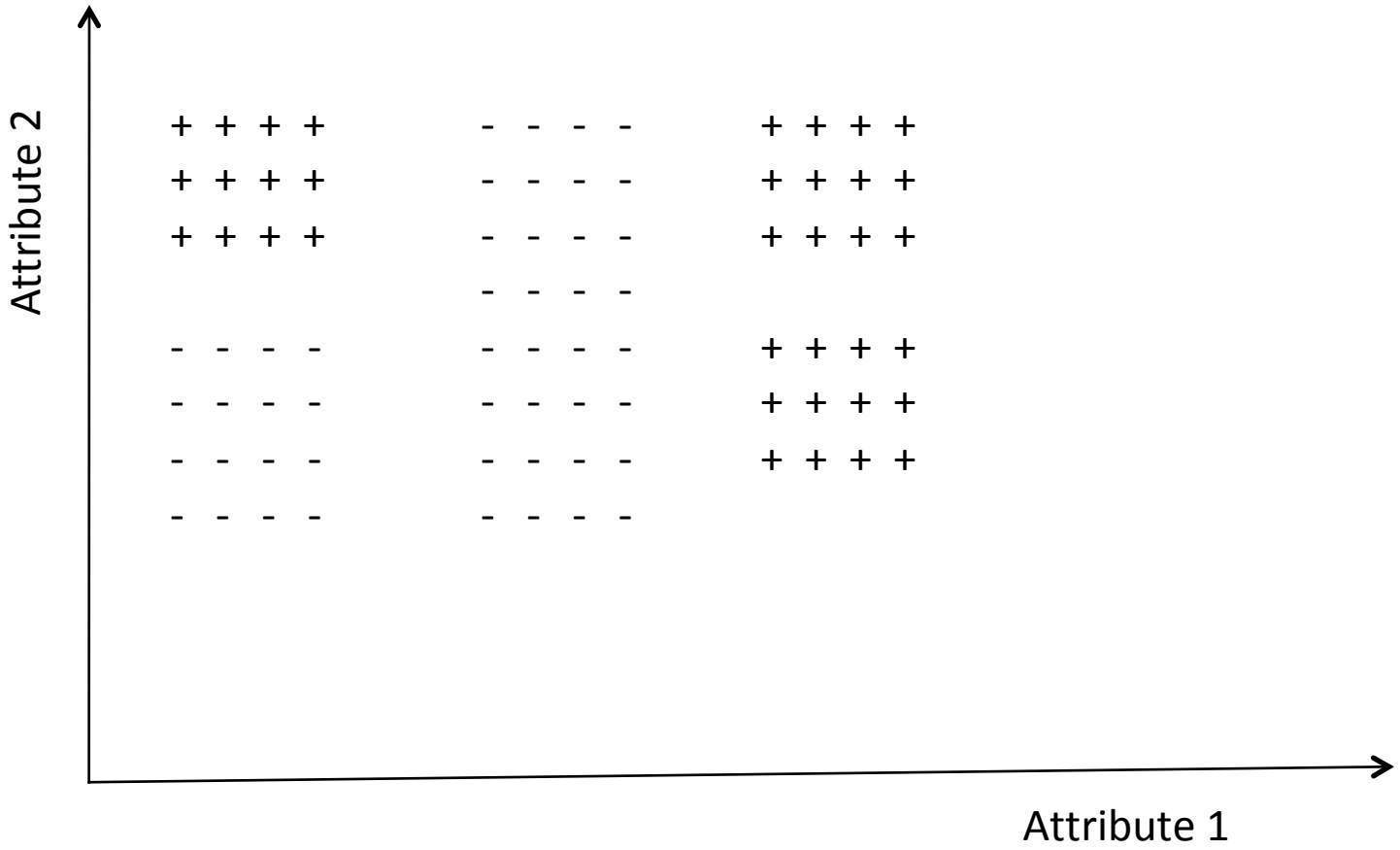


Draw axis parallel
Lines to separate the
Instances of each class

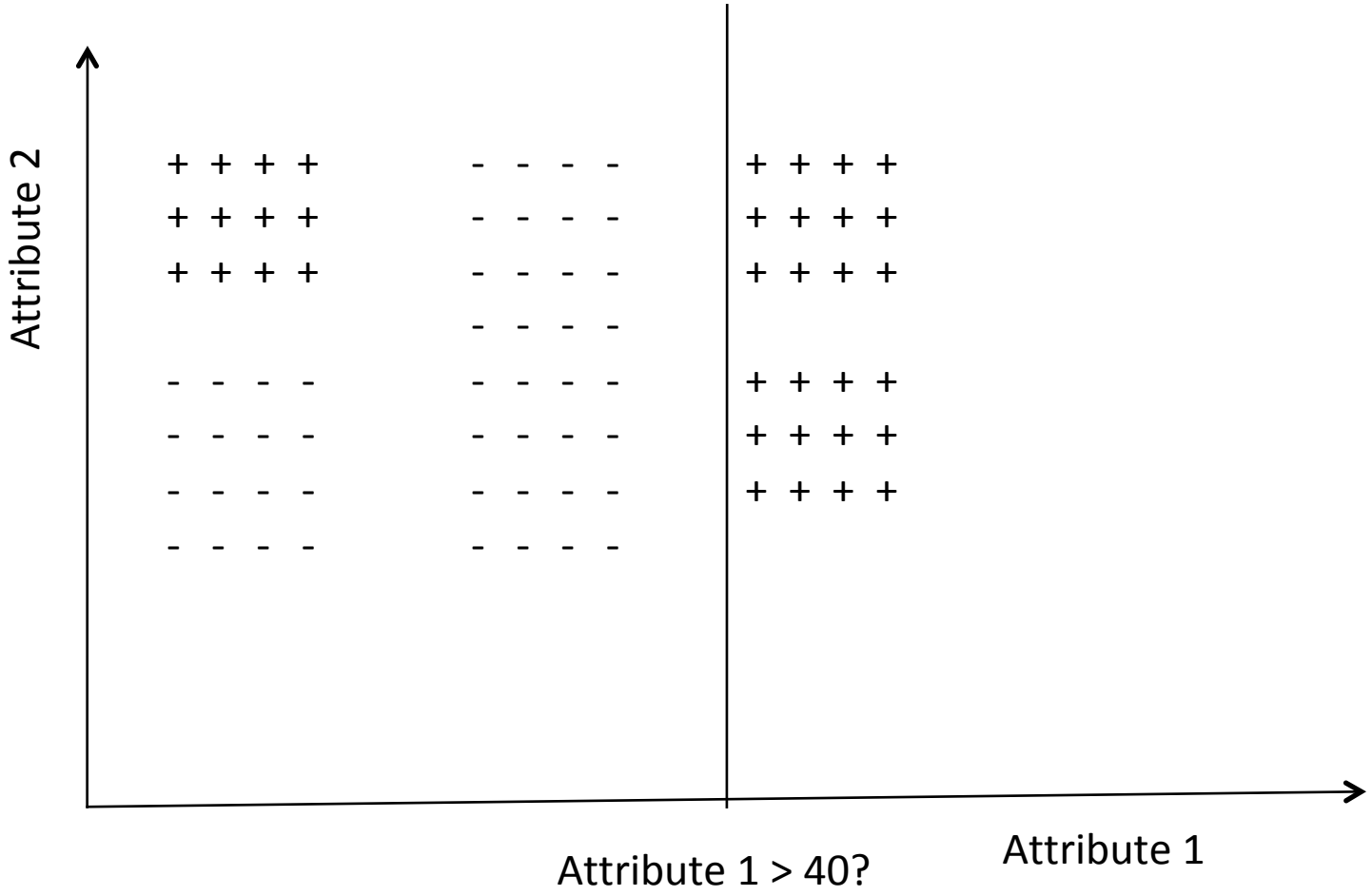
Decision Tree Construction

- Find the best structure
- Given a training data set

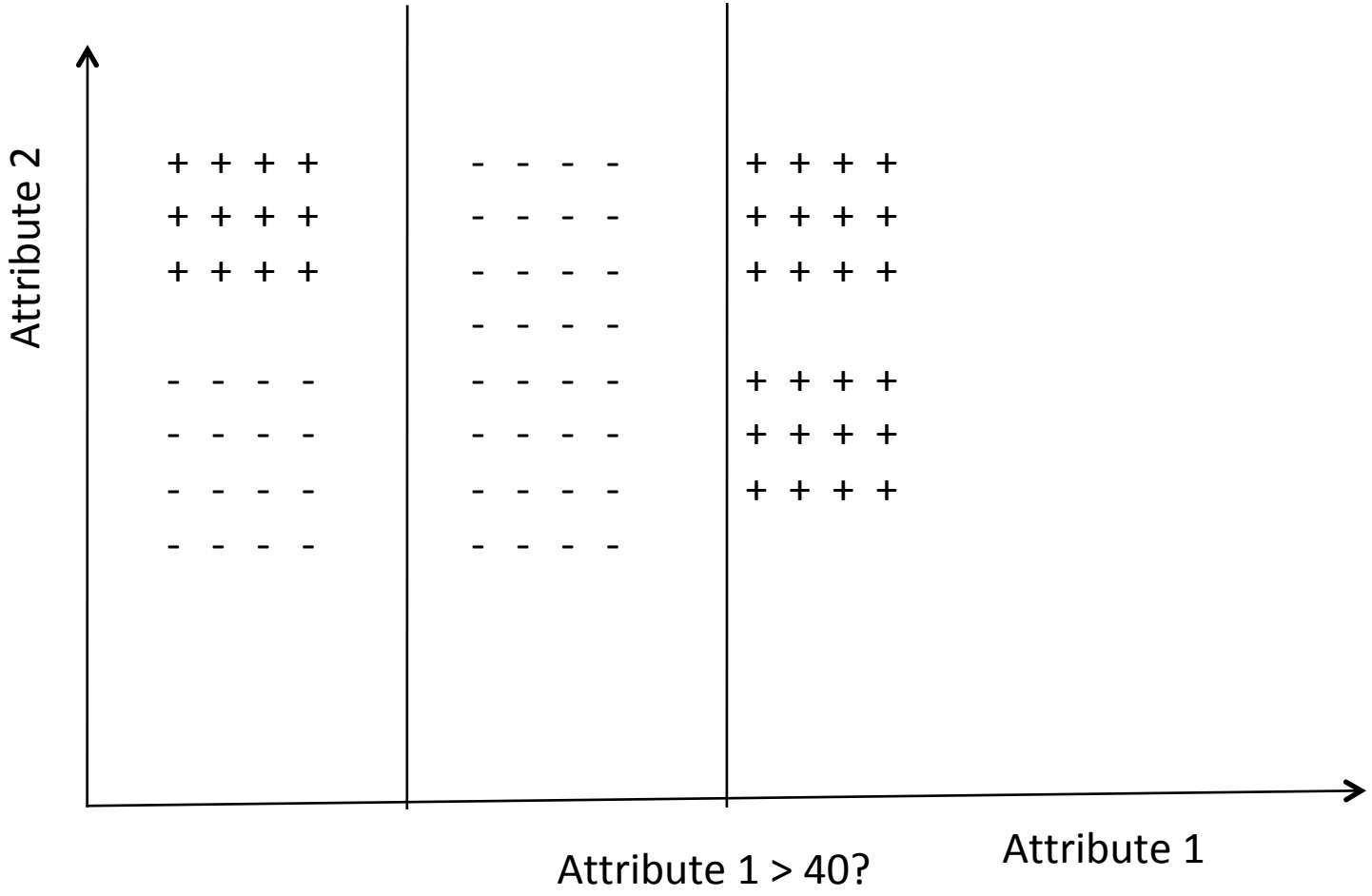
Best attribute to split?



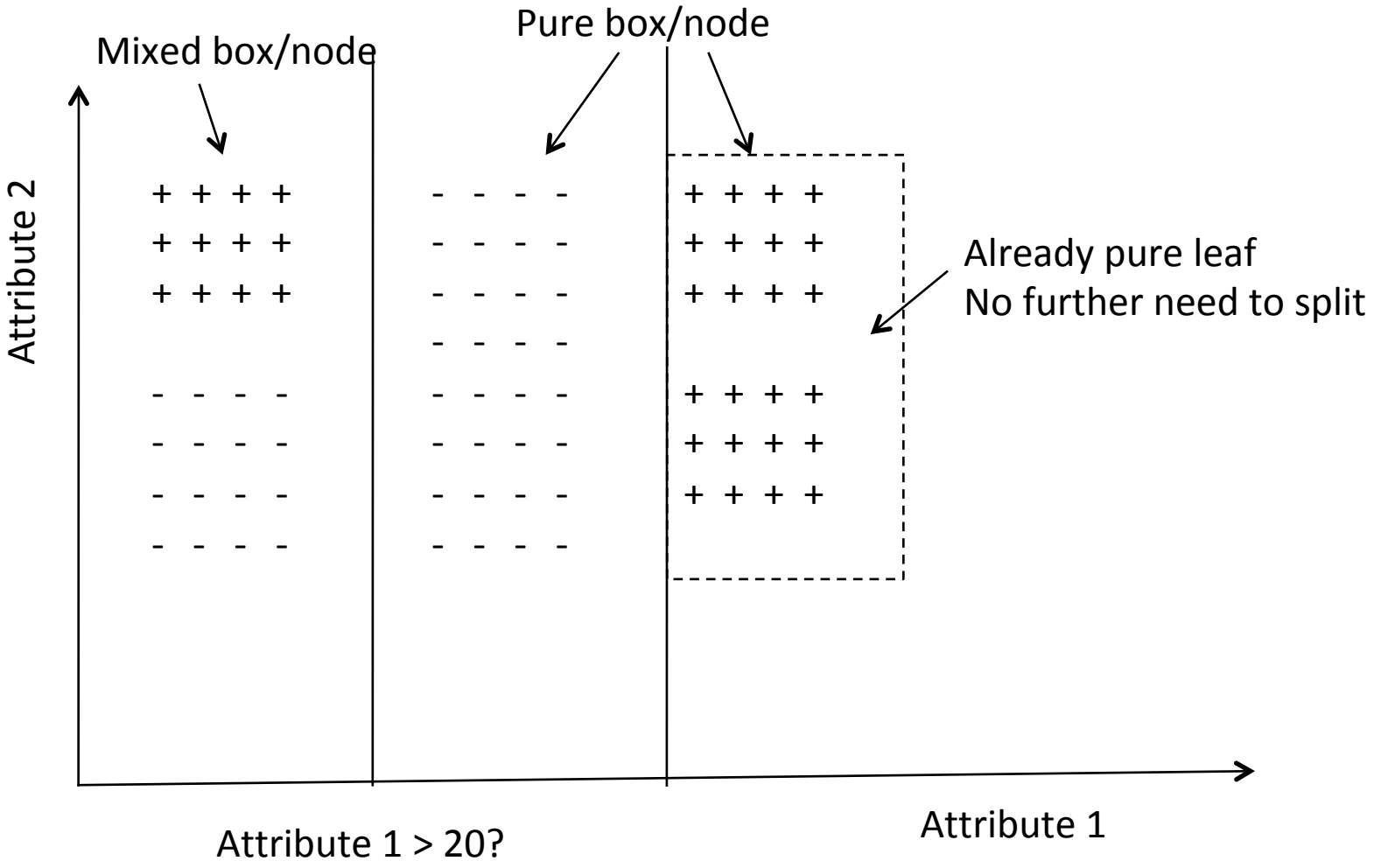
Best attribute to split?



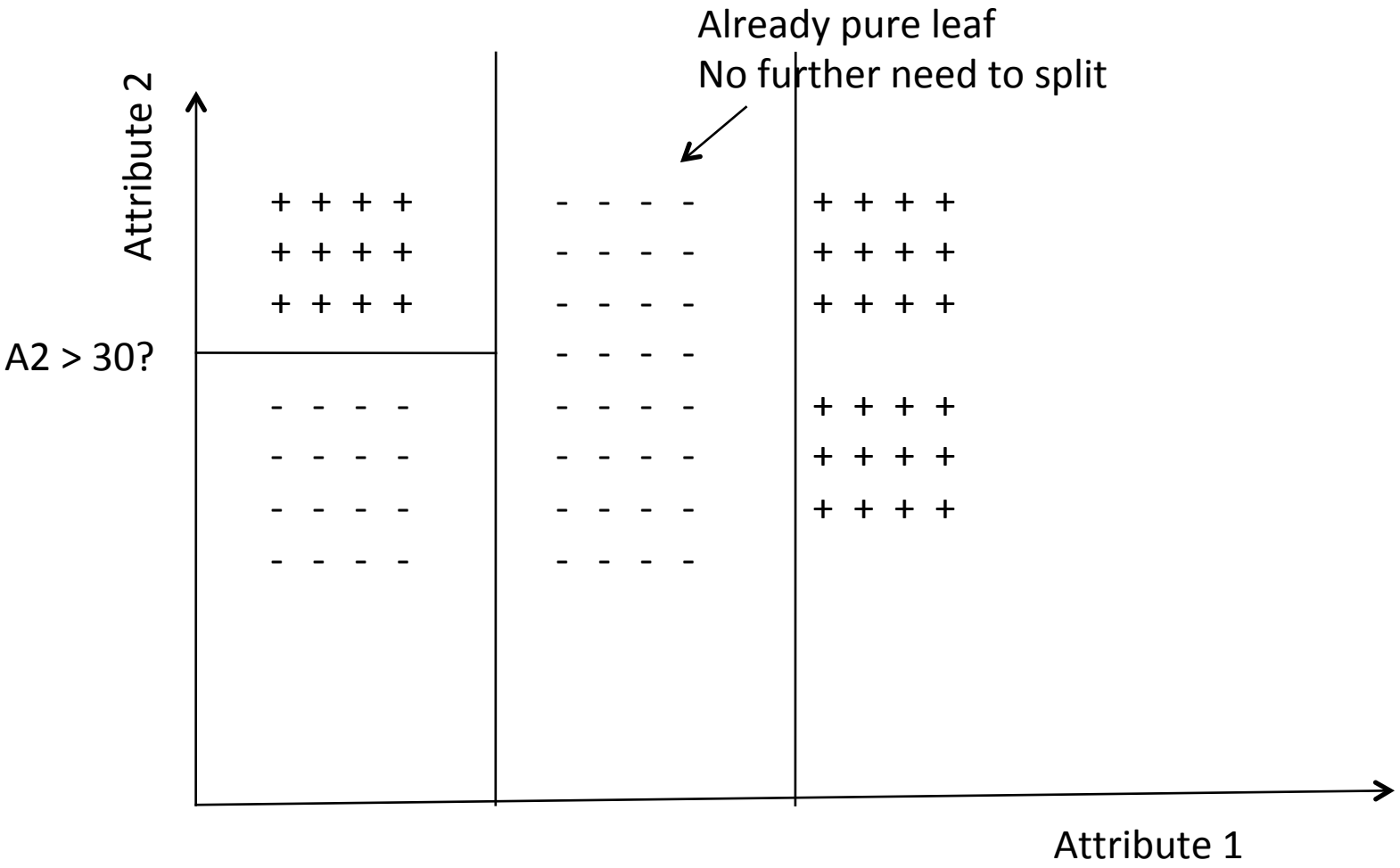
Best attribute to split?



Which split to make next?



Which split to make next?



Choosing the Best Attribute

Key problem: choosing which attribute to split a given set of examples

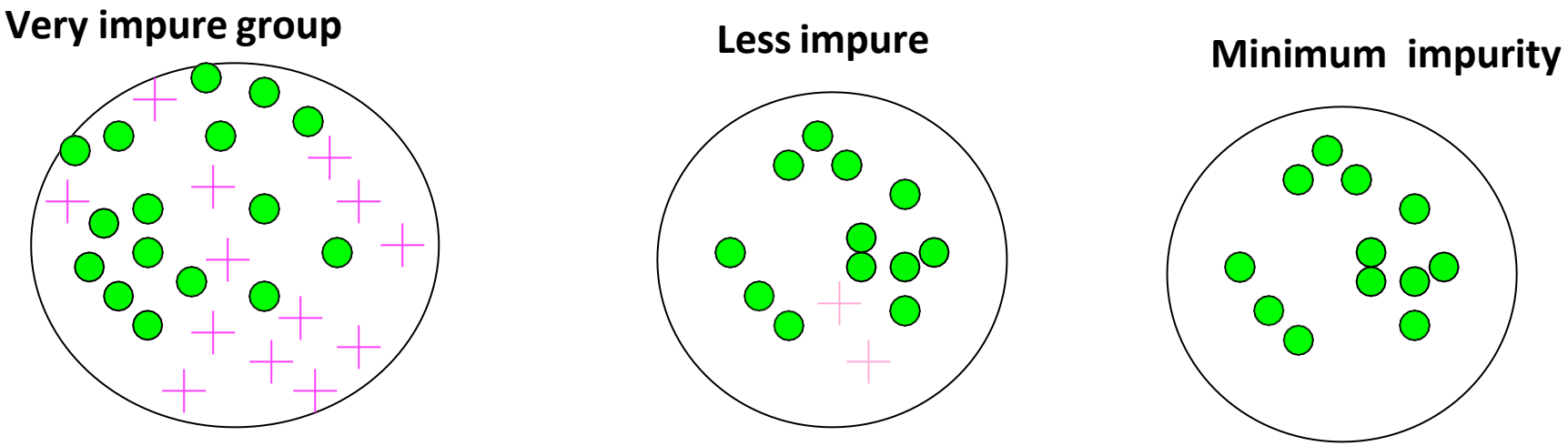
- Some possibilities are:
  - **Random:** Select any attribute at random
  - **Least-Values:** Choose the attribute with the smallest number of possible values
  - **Most-Values:** Choose the attribute with the largest number of possible values
  - **Max-Gain:** Choose the attribute that has the largest expected *information gain*
    - i.e., attribute that results in smallest expected size of subtrees rooted at its children
- The ID3 algorithm uses the Max-Gain method of selecting the best attribute

Based on slide by Byron Boots

Information Gain

Impurity/Entropy (informal)

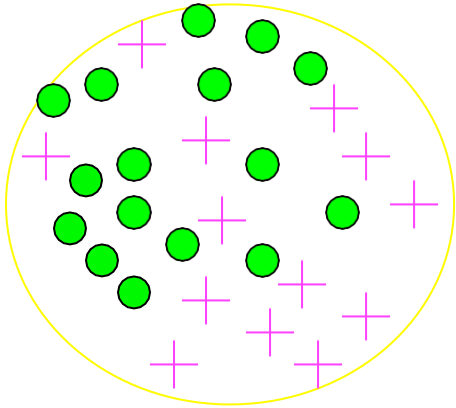
- Measures the level of **impurity** in a group of examples



Based on slide by Pedro Domingos

Entropy: a common way to measure impurity

- Entropy = $\sum_i - p_i \log_2 p_i$



p_i is the probability of class i

Compute it as the proportion of class i in the set.

- Entropy comes from information theory. The higher the entropy the more the information content.

What does that mean for learning from examples?

Based on slide by Pedro Domingos

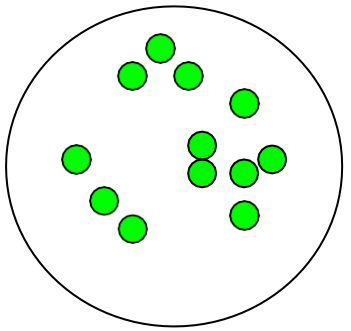
2-Class Cases

Entropy $H(x) = -\sum_{i=1}^{n} P(x=i)\log_2 P(x=i)$

- What is the entropy of a group in which all examples belong to the same class?
  - entropy = - 1 $\log_2 1$ = 0

  not a good training set for learning

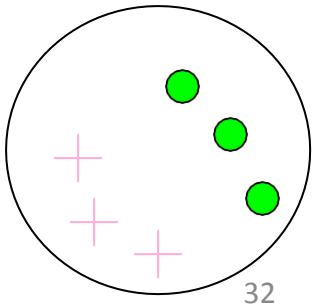
Minimum impurity



- What is the entropy of a group with 50% in either class?
  - entropy = -0.5 $\log_2 0.5$ – 0.5 $\log_2 0.5$ =1

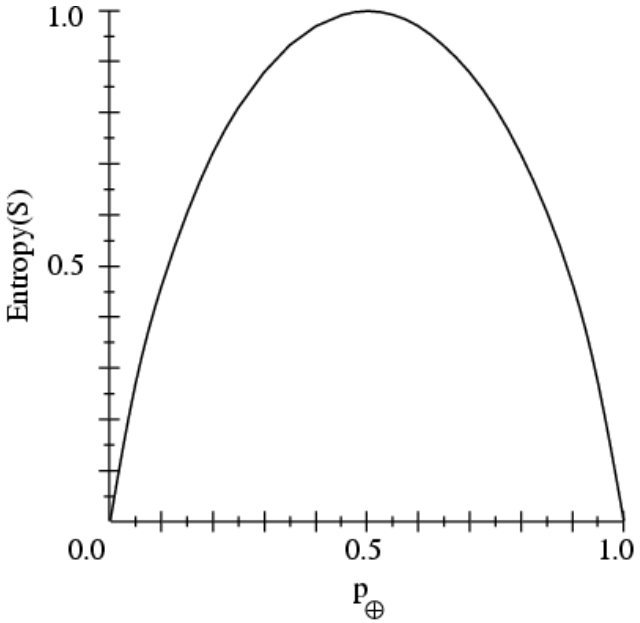
  good training set for learning

Maximum impurity



Based on slide by Pedro Domingos

Sample Entropy



- S is a sample of training examples
- $p_\oplus$ is the proportion of positive examples in S
- $p_\ominus$ is the proportion of negative examples in S
- Entropy measures the impurity of S

$$H(S) \equiv -p_\oplus \log_2 p_\oplus - p_\ominus \log_2 p_\ominus$$

Slide by Tom Mitchell

Entropy Example

PlayTennis: training examples

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Based on example by Tom Mitchell

Entropy

Entropy of the database:

Entropy(S) =

$$- (9/14) \log_2 (9/14) - (5/14) \log_2 (5/14)$$

$$= 0.940$$

Based on example by Tom Mitchell

Entropy

Entropy of the database:

Entropy(S) =

$$- (9/14) \log_2 (9/14) - (5/14) \log_2 (5/14)$$

$$= 0.940$$

What is the next step?

Based on example by Tom Mitchell

Selecting the Next Attribute

PlayTennis: training examples

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Which attribute creates the most homogeneous branches?

Information Gain

- We want to determine which attribute in a given set of training feature vectors is most useful for discriminating between the classes to be learned.
- Information gain tells us how important a given attribute of the feature vectors is.
- We will use it to decide the ordering of attributes in the nodes of a decision tree.

Based on slide by Pedro Domingos

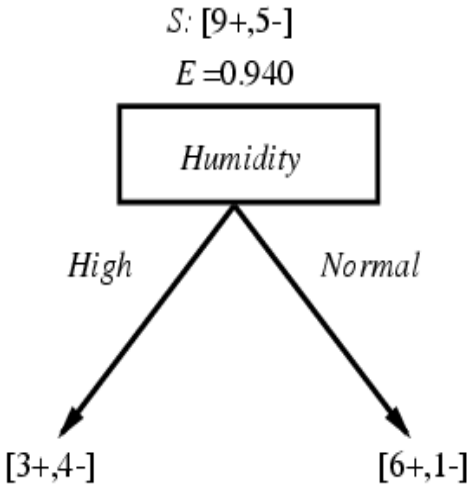
Information Gain

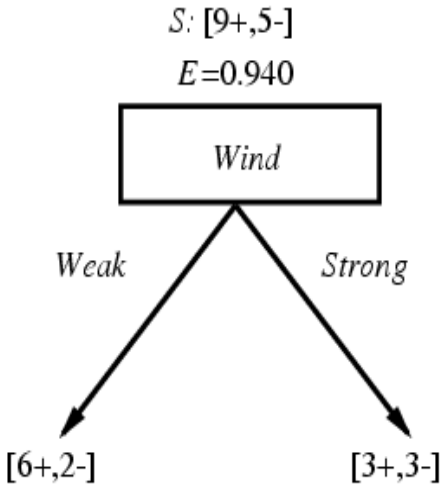
- Calculate Gain(S, A): Estimate the reduction in entropy we obtain if we know the value of attribute A for the examples in S

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

Choose the attribute that gives the maximum information gain

Selecting the Next Attribute

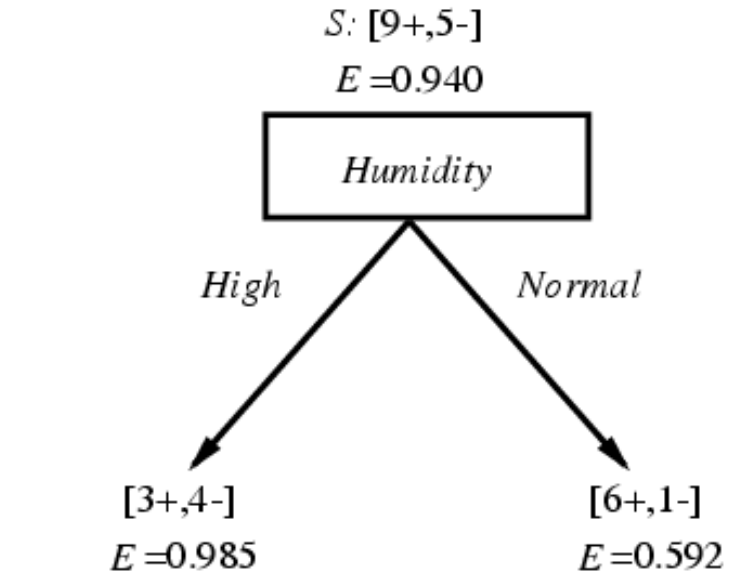




PlayTennis: training examples

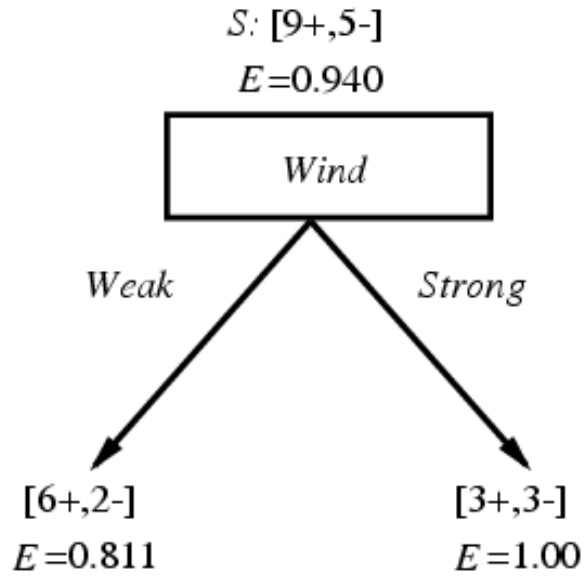
Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Selecting the next attribute



Gain (S, Humidity)

= .940 - (7/14).985 - (7/14).592
= .151



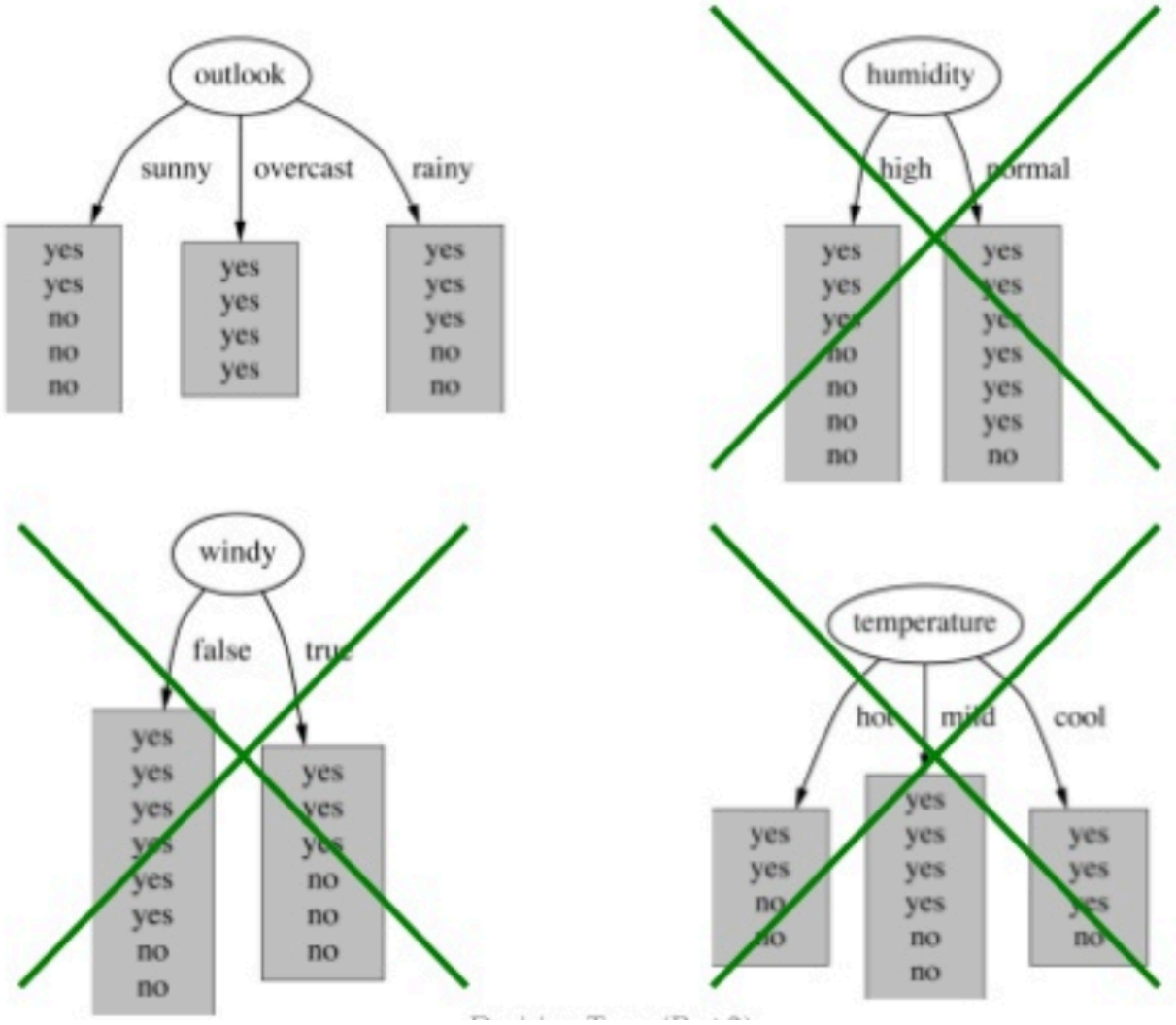
Gain (S, Wind)

= .940 - (8/14).811 - (6/14)1.0
= .048

Gain (S, Outlook) = 0.246

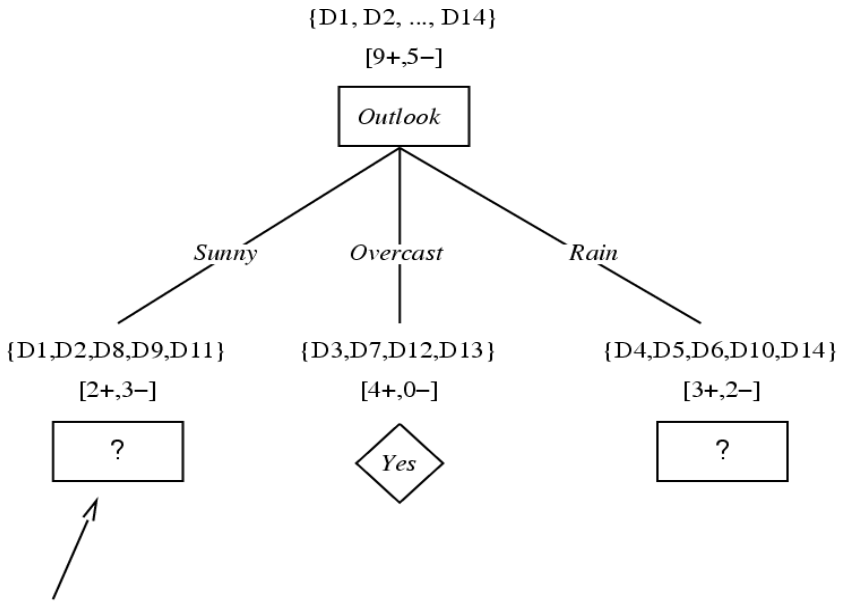
Gain (S, Temp) = 0.029

Selecting the next attribute



Based on example by Tom Mitchell

What is the next step??



Sort the training examples according to the selected attribute and the values it takes

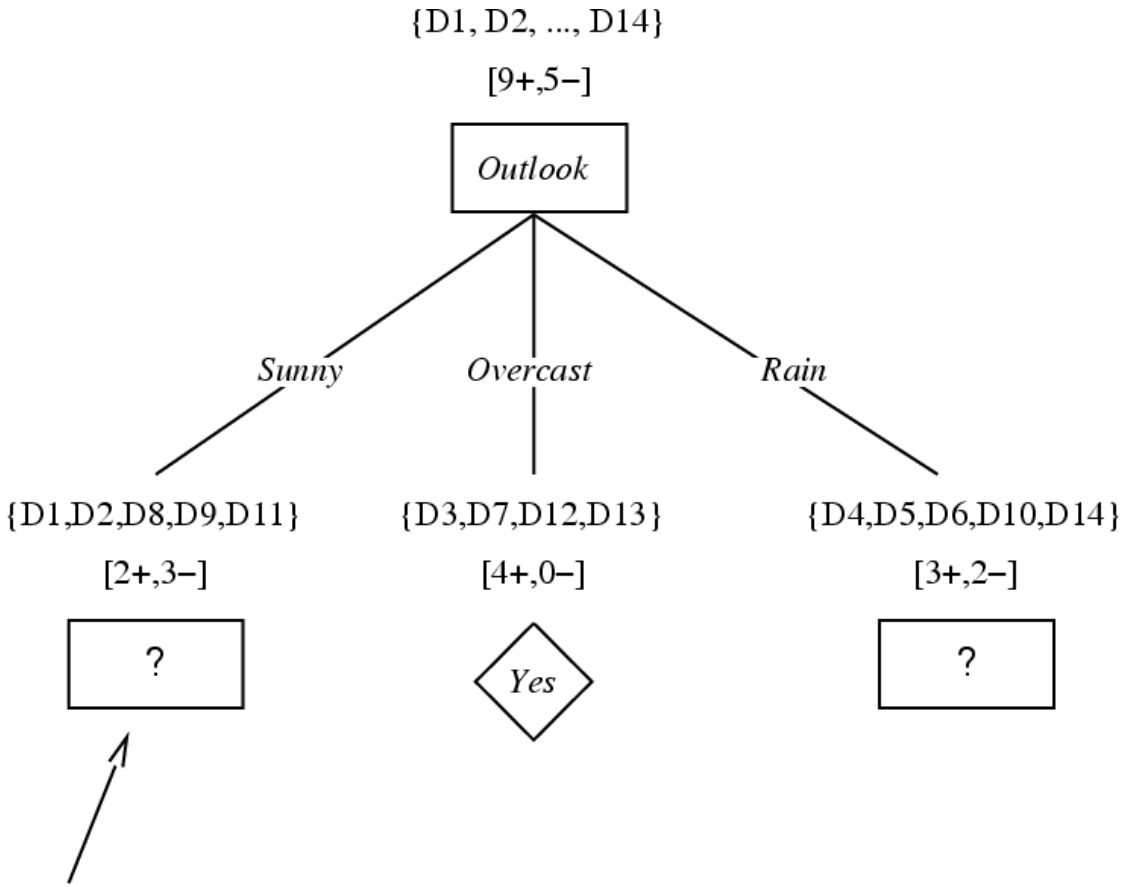
Find the next attribute

PlayTennis: training examples

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Based on example by Tom Mitchell

What is the next step??



Sort the training examples according to the selected attribute and the values it takes

Find the next attribute

S_{sunny} = {D1,D2,D8,D9,D11}

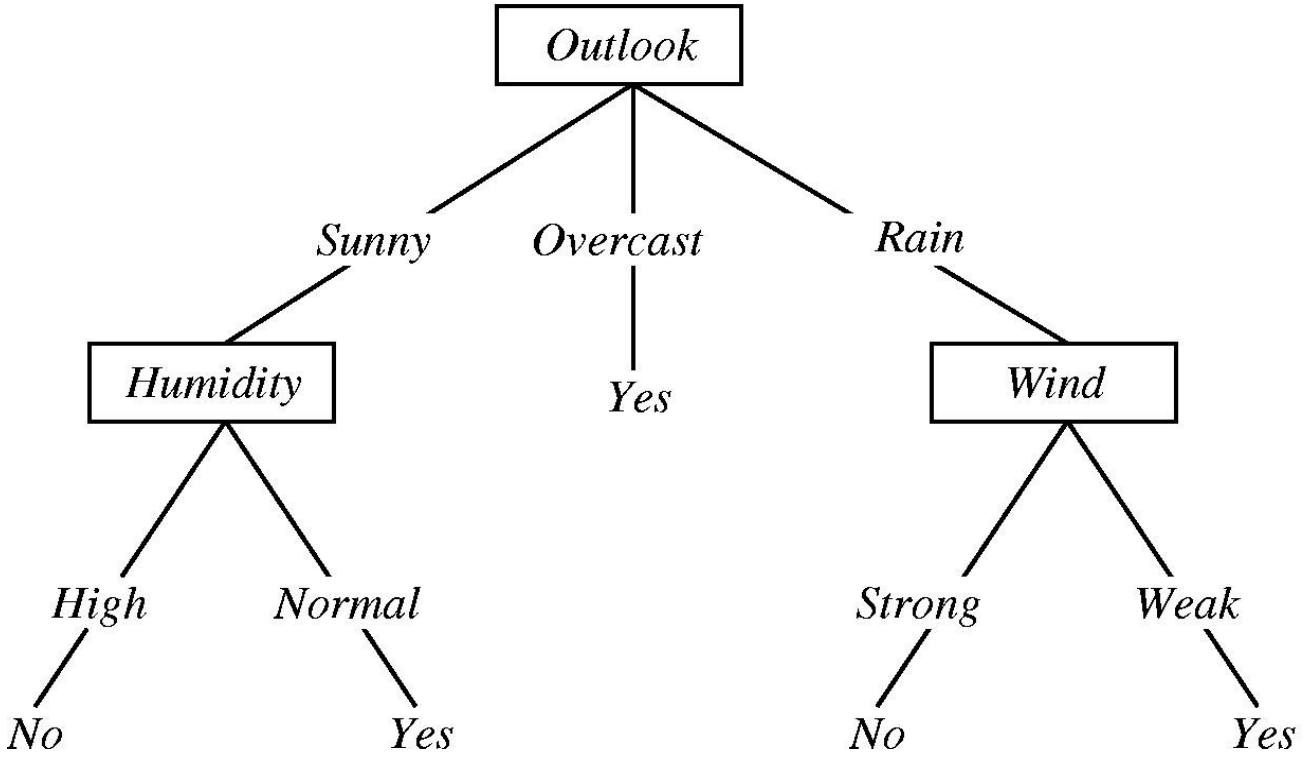
$Gain(S_{sunny}, Humidity) = .970 - (3/5)\,0.0 - (2/5)\,0.0 = .970$

$Gain(S_{sunny}, Temperature) = .970 - (2/5)\,0.0 - (2/5)\,1.0 - (1/5)\,0.0 = .570$

$Gain(S_{sunny}, Wind) = .970 - (2/5)\,1.0 - (3/5)\,.918 = .019$

Based on example by Tom Mitchell

Final Decision Tree



Based on example by Tom Mitchell

Algorithm Steps

- First the entropy of the total dataset is calculated
- The dataset is then split on different attributes
- The entropy for each branch is calculated. Then it is added proportionally, to get total entropy for the split
- The resulting entropy is subtracted from the entropy before the split

- The result is the Information Gain, or decrease in entropy
- The attribute that yields the largest IG is chosen for the decision node

The ID3 Algorithm

- Given a set of examples, S
  - Described by a set of attributes A_i
  - Categorised into categories c_j

1. Choose the root node to be attribute A
   - Such that A scores highest for information gain
     - Relative to S, i.e., gain(S,A) is the highest over all attributes
2. For each value v that A can take
   - Draw a branch and label each with corresponding v

The ID3 Algorithm

- For each branch you've just drawn (for value v)
  - If S_v only contains examples in category c
    - Then put that category as a leaf node in the tree
  - If S_v is empty
    - Then find the default category (which contains the most examples from S)
      - Put this default category as a leaf node in the tree
  - Otherwise
    - Remove A from attributes which can be put into nodes
    - Replace S with S_v
    - Find new attribute A scoring best for Gain(S, A)
    - Start again at part 2
- Make sure you replace S with S_v

Questions?

Example

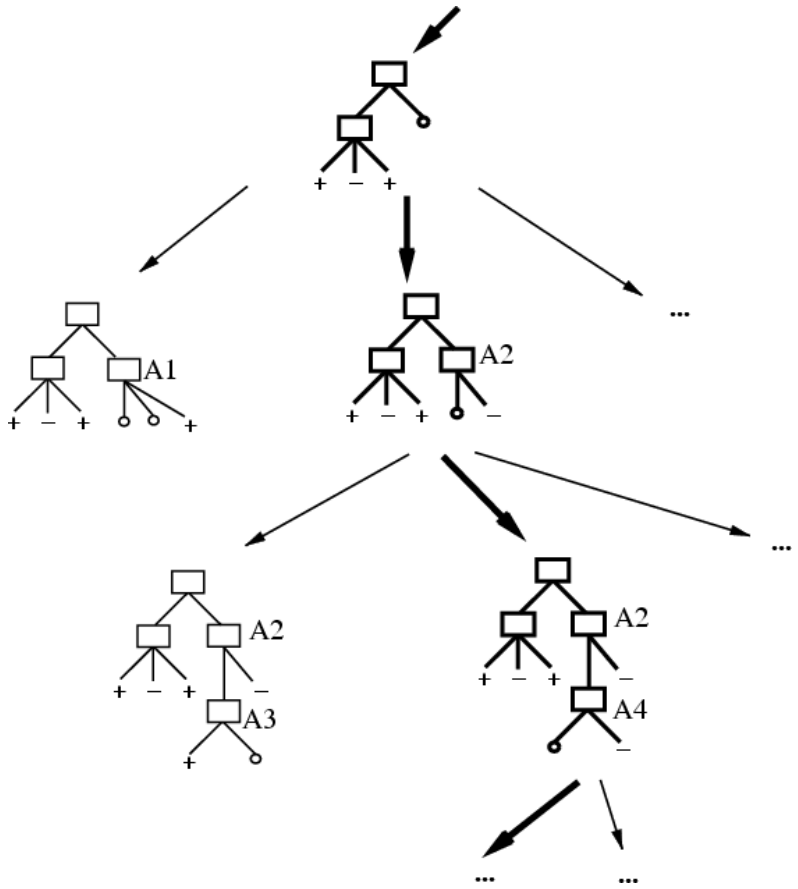
Consider the following data: -

Age	Stipend	Category	Academic Rating	Buys Laptop
<= 25	high	student	fair	no
<=25	high	student	excellent	no
26-30	high	student	fair	yes
>30	medium	student	fair	yes
>30	low	Project associate	fair	yes
>30	low	Project associate	excellent	no
26-30	low	Project associate	excellent	yes
<=25	medium	student	fair	no
<=25	low	Project associate	fair	yes
>30	medium	Project associate	fair	yes
<=25	medium	Project associate	excellent	yes
26-30	medium	student	excellent	yes
26-30	high	Project associate	fair	yes
>30	medium	student	excellent	no

Last Column is the column to be predicted (y), calculate Information gain on
- Age
- Stipend

Which attribute among the above 2 can be considered for the split at level 0 and why (consider entropy gain).

Which Tree Should We Output?



- ID3 performs heuristic search through space of decision trees
- It stops at smallest acceptable tree. Why?

Slide by Tom Mitchell

Preference bias: Ockham's Razor

- Principle stated by William of Ockham (1285-1347)
  - "*non sunt multiplicanda entia praeter necessitatem*"
  - entities are not to be multiplied beyond necessity
  - AKA Occam's Razor, Law of Economy, or Law of Parsimony

Idea: The simplest consistent explanation is the best

- Therefore, the smallest decision tree that correctly classifies all of the training examples is best
  - Finding the provably smallest decision tree is NP-hard
  - ...So instead of constructing the absolute smallest tree consistent with the training examples, construct one that is pretty small

Overfitting in Decision Trees

- Many kinds of "noise" can occur in the examples:
  - Two examples have same attribute/value pairs, but different classifications
  - Some values of attributes are incorrect because of errors in the data acquisition process or the preprocessing phase
  - The instance was labeled incorrectly (+ instead of -)

- Also, some attributes are irrelevant to the decision-making process
  - e.g., color of a die is irrelevant to its outcome

Based on Slide from M. desJardins & T. Finin

Overfitting in Decision Trees

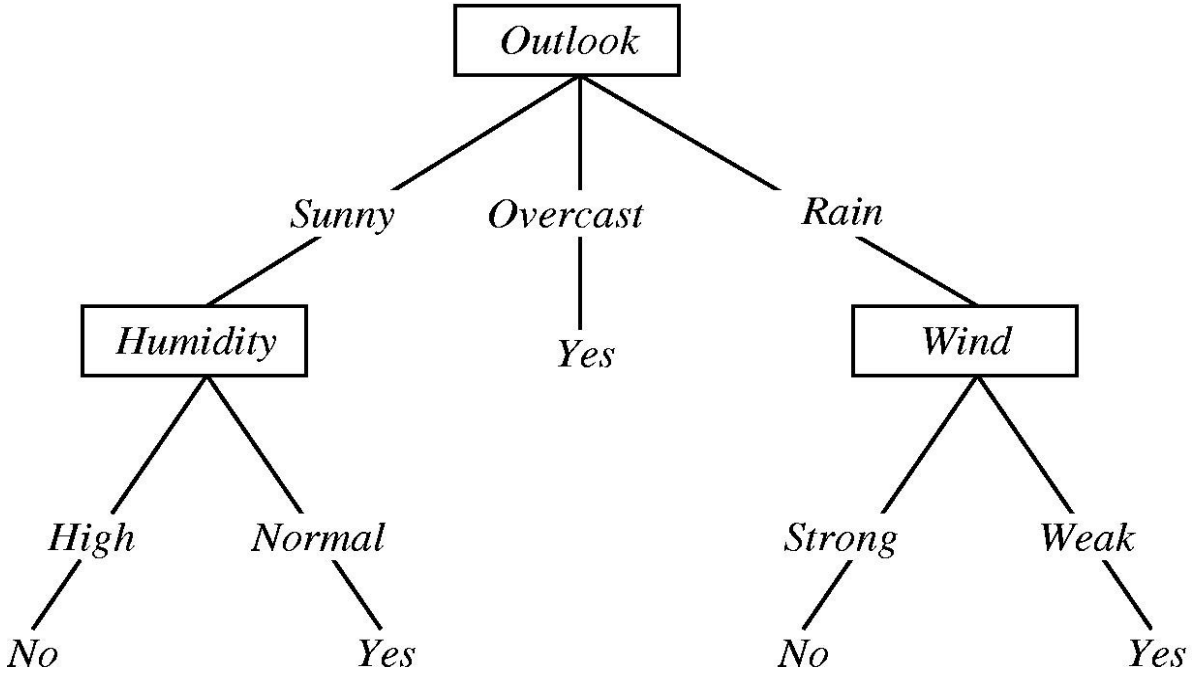
- Irrelevant attributes can result in *overfitting* the training example data
  - If hypothesis space has many dimensions (large number of attributes), we may find **meaningless regularity** in the data that is irrelevant to the true, important, distinguishing features

- If we have too little training data, even a reasonable hypothesis space will 'overfit'

Based on Slide from M. desJardins & T. Finin

Noisy Samples

Consider adding a **noisy** training example to the following tree:



What would be the effect of adding:

<outlook=sunny, temperature=hot, humidity=normal, wind=strong, playTennis=No> ?

Based on Slide by Pedro Domingos

Overfitting in Decision Trees

Consider error of hypothesis h over

- training data: $error_{train}(h)$
- entire distribution $\mathcal{D}$ of data: $error_{\mathcal{D}}(h)$

Hypothesis $h \in H$ **overfits** training data if there is an alternative hypothesis $h' \in H$ such that

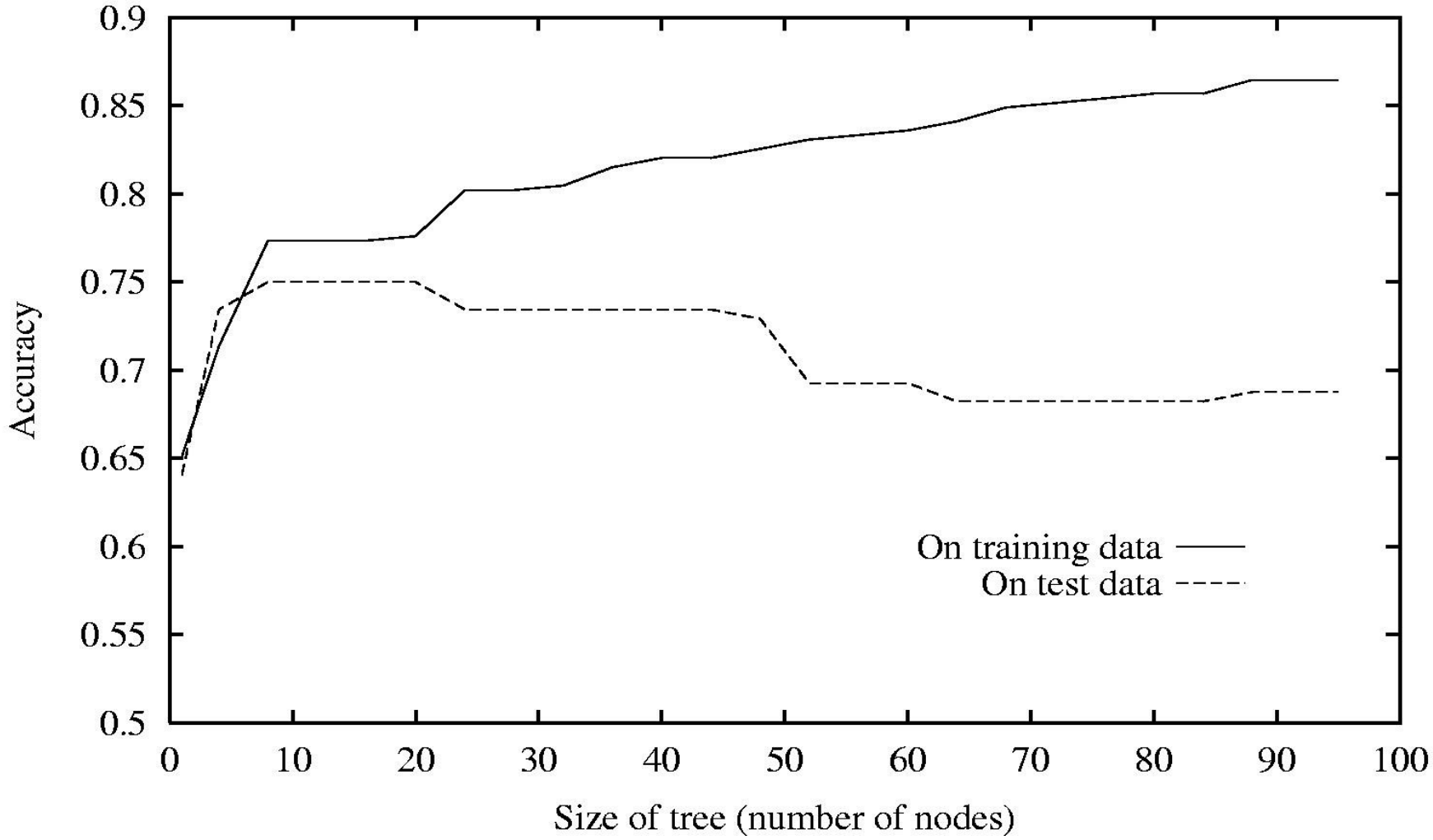
$$error_{train}(h) < error_{train}(h')$$

and

$$error_{\mathcal{D}}(h) > error_{\mathcal{D}}(h')$$

Slide by Pedro Domingos

Overfitting in Decision Trees



Slide by Pedro Domingos

Avoiding Overfitting in Decision Trees

How can we avoid overfitting?

- Stop growing when data split is not statistically significant
- Acquire more training data
- Remove irrelevant attributes (manual process – not always possible)
- **Grow full tree, then post-prune**

How to select "best" tree:

- Measure performance over training data
- Measure performance over separate validation data set
- Add complexity penalty to performance measure (heuristic: simpler is better)

Based on Slide by Pedro Domingos

Reduced-Error Pruning

Split training data further into *training* and *validation* sets

Grow tree based on *training set*

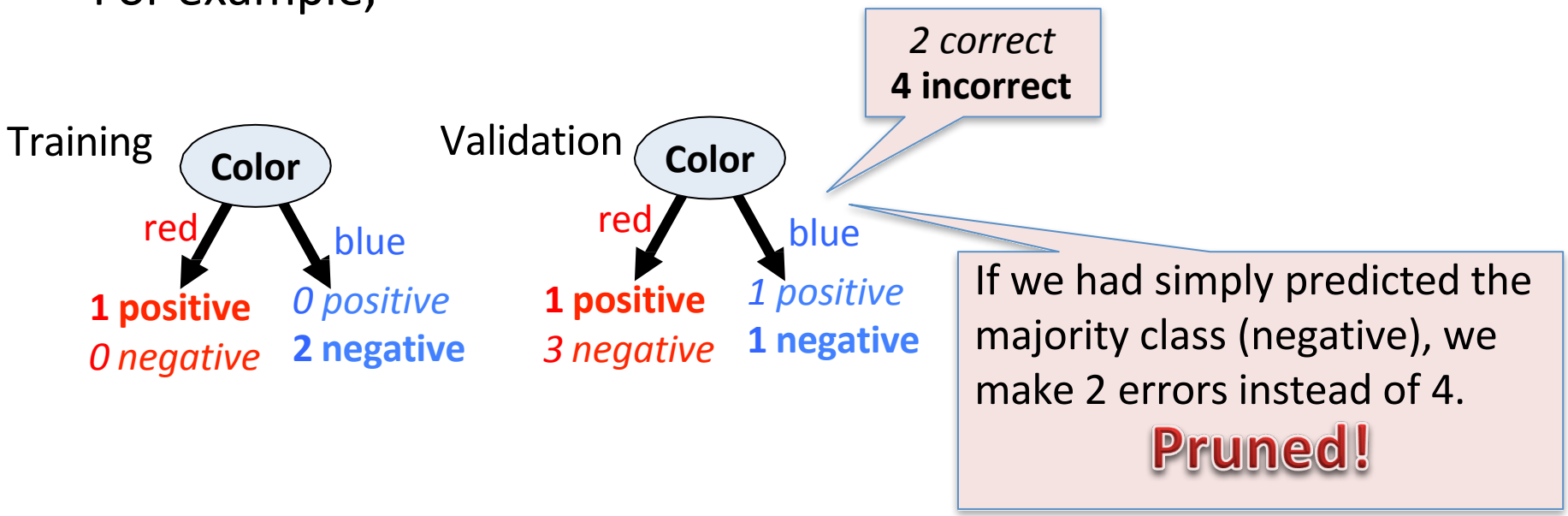
Do until further pruning is harmful:

1. Evaluate impact on validation set of pruning each possible node (plus those below it)
2. Greedily remove the node that most improves *validation set* accuracy

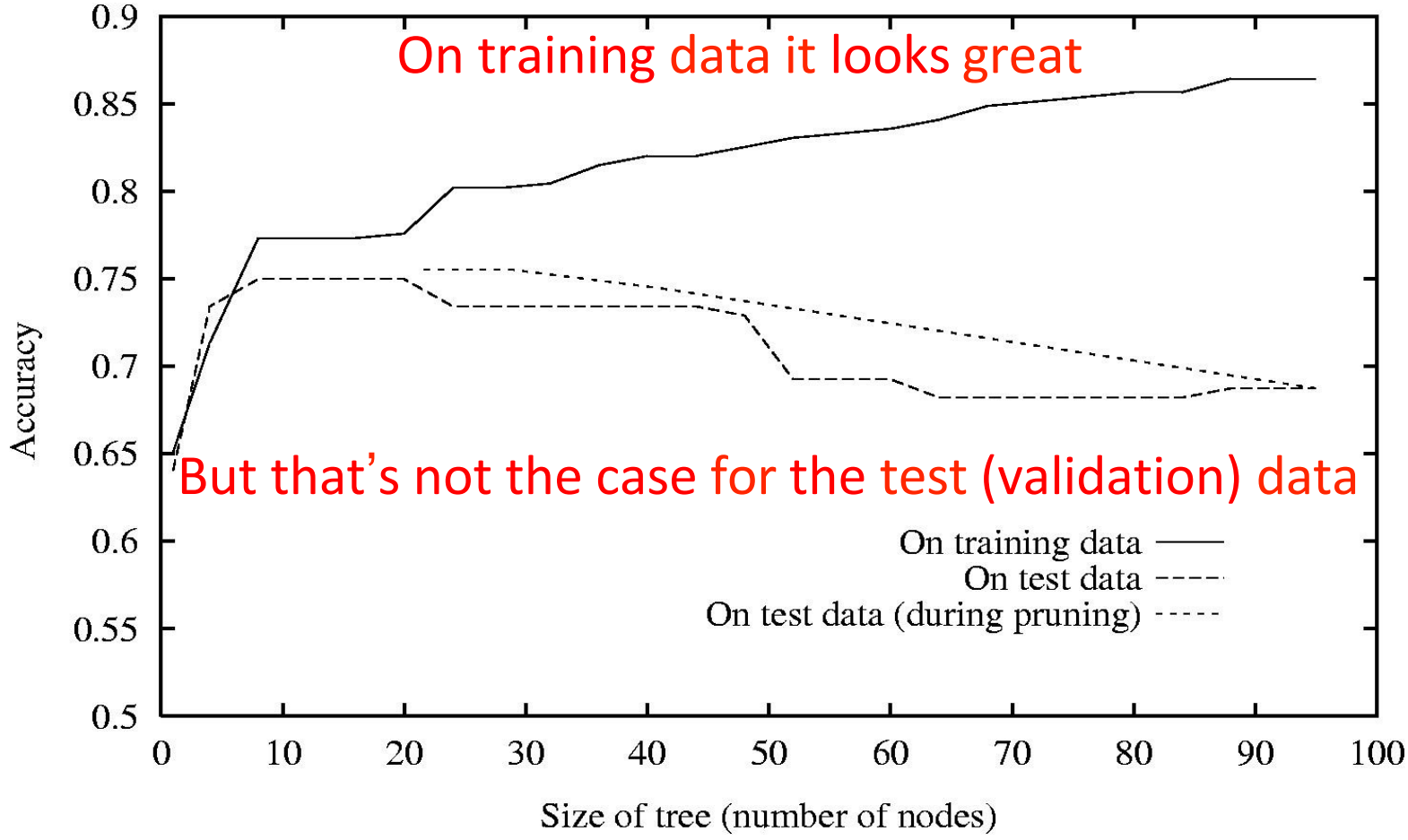
Slide by Pedro Domingos

Pruning Decision Trees

- Pruning of the decision tree is done by replacing a whole subtree by a leaf node.
- The replacement takes place if a decision rule establishes that the expected error rate in the subtree is greater than in the single leaf.
- For example,

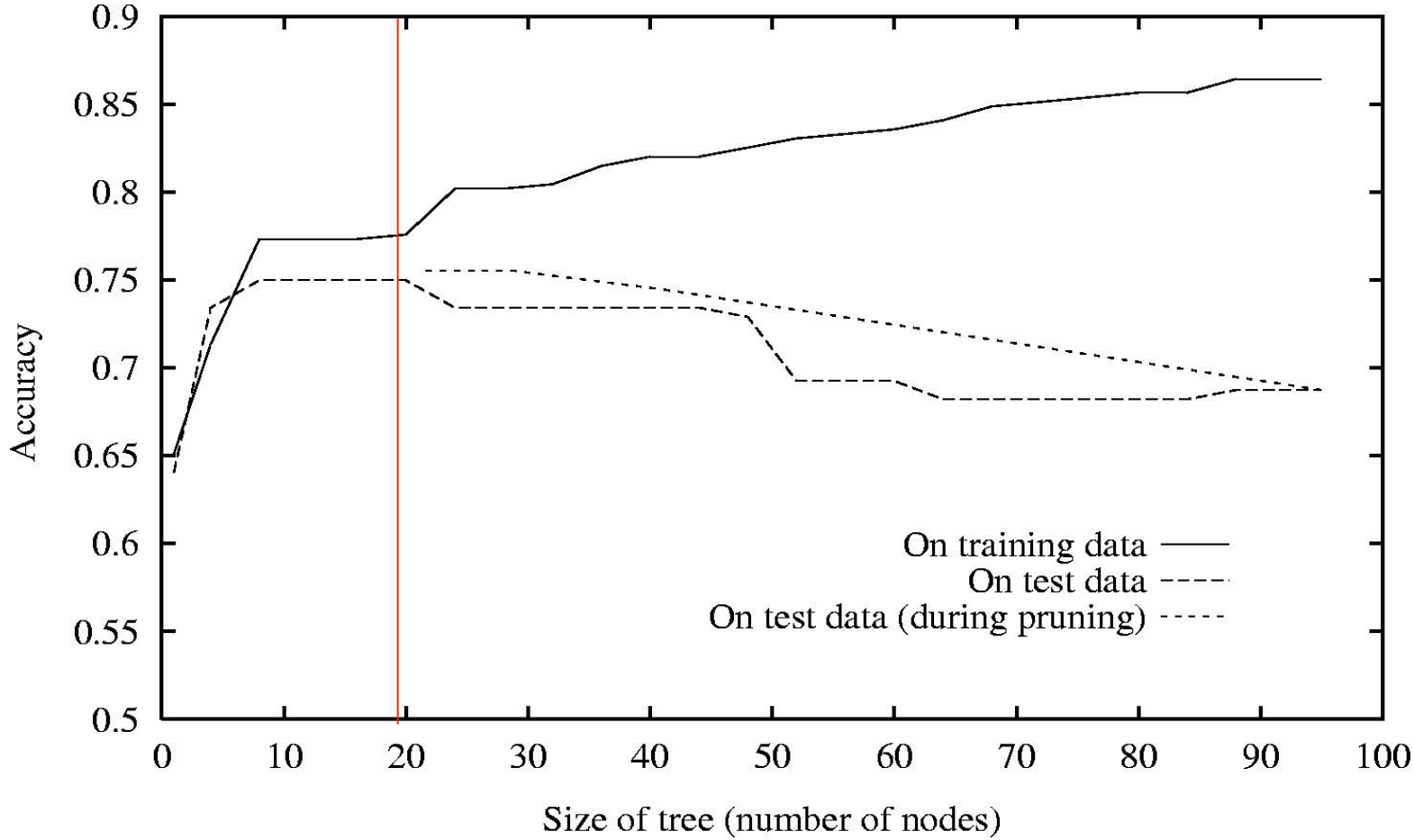


Effect of Reduced-Error Pruning



Based on Slide by Pedro Domingos

Effect of Reduced-Error Pruning



The tree is pruned back to the red line where it gives more accurate results on the test data

Based on Slide by Pedro Domingos

Summary: Decision Tree Learning

- Widely used in practice
- Strengths include
  - Fast and simple to implement
  - Can convert to rules
  - Handles noisy data
- Weaknesses include
  - Univariate splits/partitioning using only one attribute at a time --- limits types of possible trees
  - Large decision trees may be hard to understand
  - Requires fixed-length feature vectors
  - Non-incremental (i.e., batch method)

Any questions? Please ask on WebEx chat.

Questionnaire shared with you