Entropy-Based Automatic Decision Tree Construction

Training Set X
x1=(f11,f12,...f1m)
x2=(f21,f22, f2m)
.
.
xn=(fn1,f22, f2m)

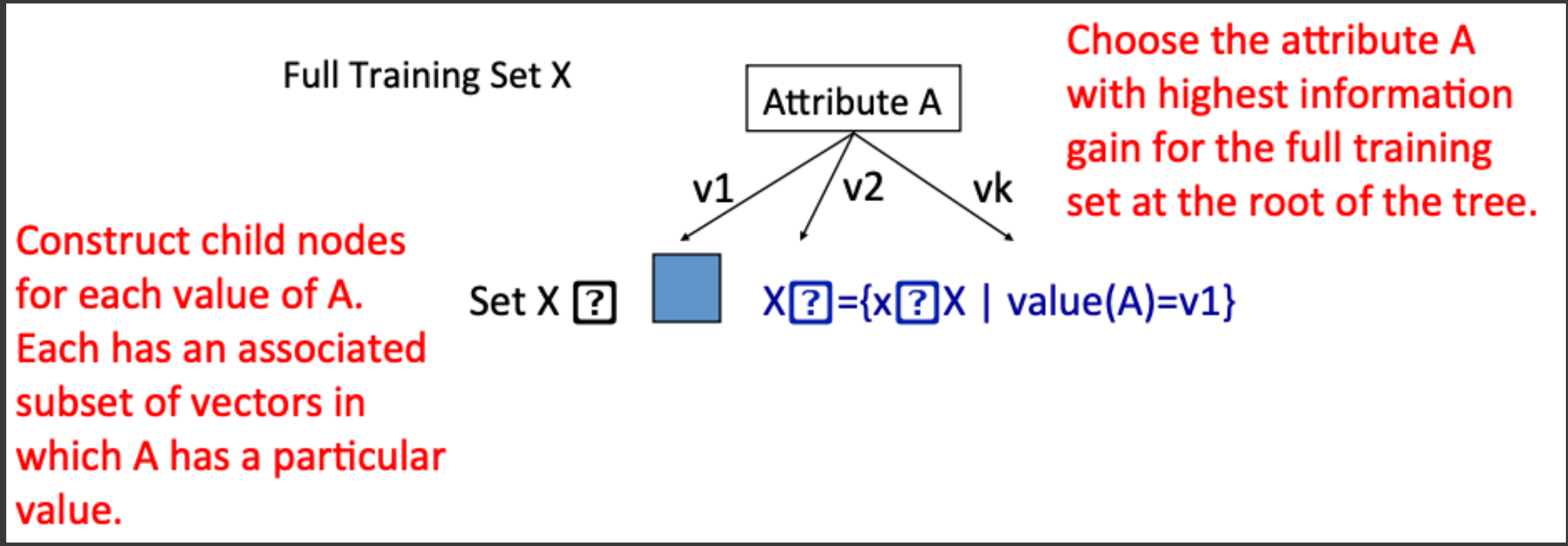
Node 1
What feature
should be used?

What values?

Quinlan suggested information gain in his ID3 system
and later the gain ratio, both based on entropy.

Based on slide by
Pedro Domingos

Using Information Gain to Construct A Decision Tree



Disadvantage of information gain:

- It prefers attributes with large number of values that split the data into small, pure subsets

Based on slide by Pedro Domingos

Attributes with Many Values

- Problem
  - If attribute has many values, InfoGain() will select it
    - e.g., image using date = 01_01_2020 as an attribute

- Alternative approach: use GainRatio() instead
  - Taking the number and size of branches into account
  - $\text{GainRatio}(X, A) = \frac{InfoGain(X,A)}{SplitInformation(X,A)}$
  - $\text{SplitInformation}(X, A) = -\sum_{v \in values(A)} \frac{|X_v|}{|X|} \log_2 \frac{|X_v|}{X}$

X_v is a subset of X for which A has value v

Slide credit:
Pedro Domingos

Drawbacks of Information Gain

- The major drawbacks of using Information Gain as the criterion for determining which feature to be used as the node is that it tends to use the feature that has more unique values.

Training Data				
Date	Weather	Temperature	Wind Level	Go out for running?
2020-01-01	Sunny	High	Low	No
2020-01-02	Sunny	Medium	Medium	Yes
2020-01-03	Cloudy	High	Medium	Yes
2020-01-04	Cloudy	Medium	High	Yes
2020-01-05	Rainy	High	Low	No
2020-01-06	Rainy	High	Medium	No
2020-01-07	Sunny	Low	High	No

Drawbacks of Information Gain

Training Data				
Date	Weather	Temperature	Wind Level	Go out for running?
2020-01-01	Sunny	High	Low	No
2020-01-02	Sunny	Medium	Medium	Yes
2020-01-03	Cloudy	High	Medium	Yes
2020-01-04	Cloudy	Medium	High	Yes
2020-01-05	Rainy	High	Low	No
2020-01-06	Rainy	High	Medium	No
2020-01-07	Sunny	Low	High	No

Entropy (Date = 2020-01-01)

$$=-\left(\frac{1}{1}\log_2\frac{1}{1}\right)=0$$

Entropy(Date)

$$=-\left(\frac{1}{7}\log_2\frac{1}{7}+\frac{1}{7}\log_2\frac{1}{7}+\cdots\right)$$

=2.807

Drawbacks of Information Gain

Training Data				
Date	Weather	Temperature	Wind Level	Go out for running?
2020-01-01	Sunny	High	Low	No
2020-01-02	Sunny	Medium	Medium	Yes
2020-01-03	Cloudy	High	Medium	Yes
2020-01-04	Cloudy	Medium	High	Yes
2020-01-05	Rainy	High	Low	No
2020-01-06	Rainy	High	Medium	No
2020-01-07	Sunny	Low	High	No

Information Gain of Weather is 0.592
Information Gain of Temperature is 0.522
Information Gain of Wind level is 0.306

Drawbacks of Information Gain

- SplitInformation(Outlook/Weather)

Outlook	Yes	No	Count of each group
Sunny	1	2	3
Cloudy	2	0	2
Rainy	0	2	2

$$\text{SplitInformation(Outlook)} = -\left(\frac{3}{7}\log_2\frac{3}{7} + \frac{2}{7}\log_2\frac{2}{7} + \frac{2}{7}\log_2\frac{2}{7}\right) = 1.040$$

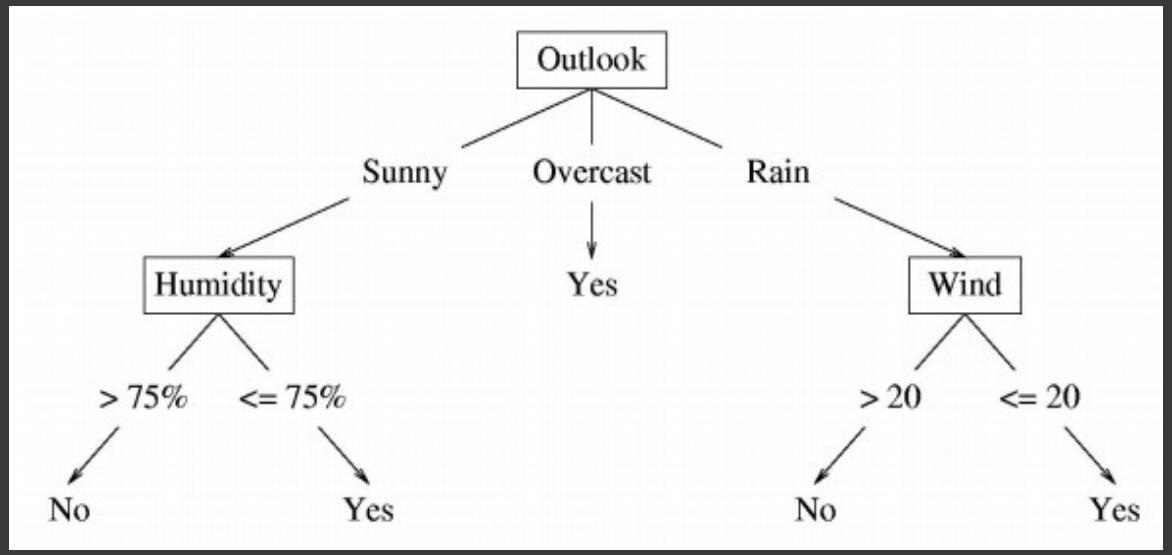
$$\text{GainRatio(Outlook)} = \frac{0.592}{1.040} = 0.57$$

Extensions of ID3

- Using gain ratios
- Real-valued data
- Noisy data and overfitting
- Setting parameters
- Cross-validation for experimental validation of performance
- Unavailable values, continuous attribute value ranges, pruning of decision trees, rule derivation, and so on

Slide credit: M. desJardins & T. Finin

Real-Valued Features



- Change to binary splits by choosing a threshold
- One method:
  - Sort instances by value, identify adjacencies with different classes

Temperature:	40	48	60	72	80	90
PlayTennis:	No	No	Yes	Yes	Yes	No

candidate splits

Choose among splits by InfoGain()

Unknown Attribute Values

Question: What if some examples are missing values of A?

Use training example anyway, sort through tree:

- If node n tests A, assign most common value of A among other examples sorted to node n
- Assign most common value of A among other examples with same class label
- Assign probability p_i to each possible value v_i of A. Assign fraction p_i of example to each descendent of tree. Classify new examples in same fashion

Slide credit: Pedro Domingos

Noisy Data

- 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

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

Slide credit: M. desJardins & T. Finin

Overfitting

- 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"

Slide credit: M. desJardins & T. Finin

Overfitting

- Consider error of hypothesis h over
  - training data: $error_{train}(h)$
  - Entire distribution D of data: $error_D(h)$

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

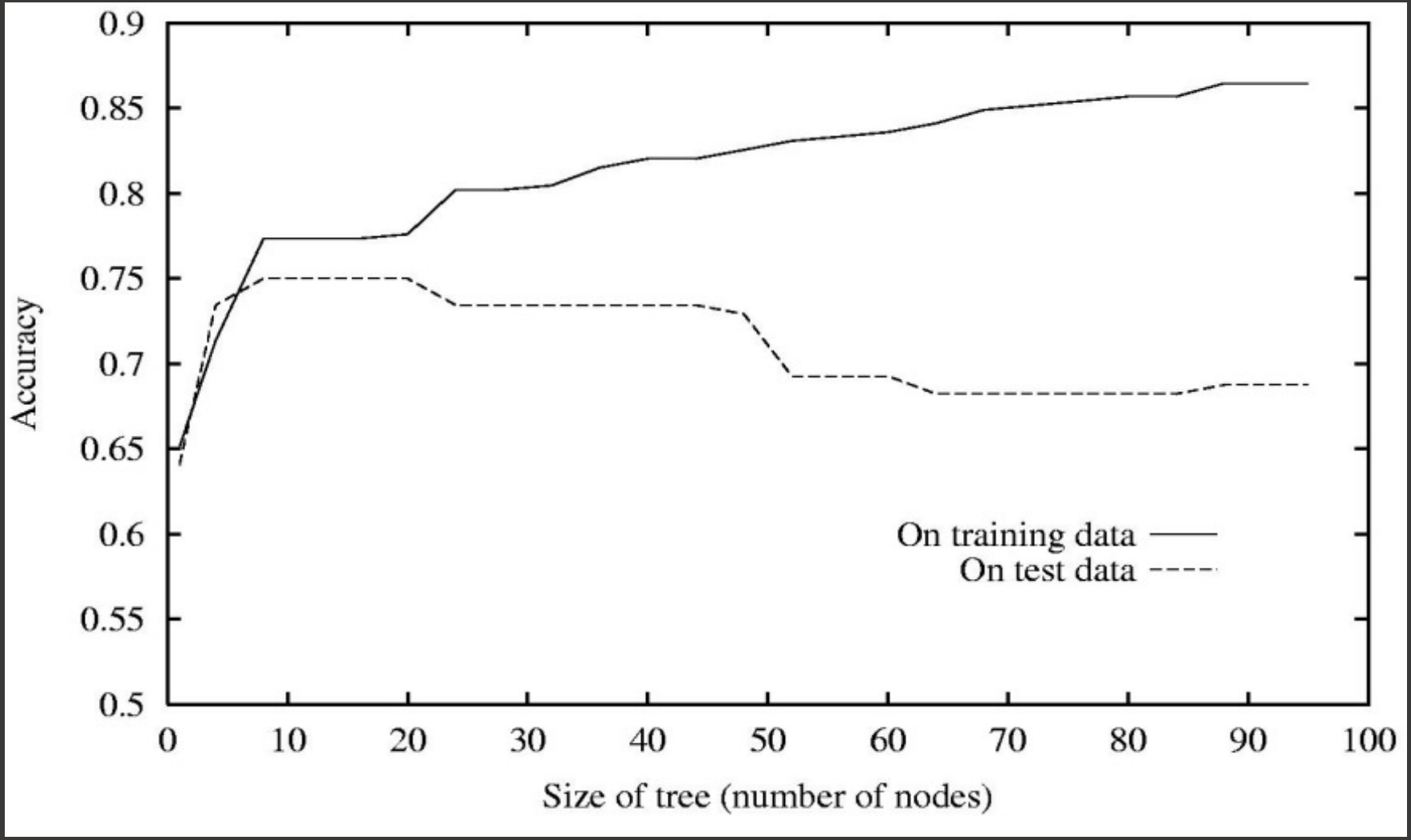
$$error_{train}(h) < error_{train}(h')$$

and

$$error_D(h) > error_D(h')$$

Slide credit: Pedro Domingos

Overfitting in Decision Trees



Slide credit: Pedro Domingos

Avoiding Overfitting

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

Slide credit: Pedro Domingos

Reduced-Error Pruning

Split data into *training* and *validation sets*

Grow tree based on *training set*

Do the following until further pruning is harmful:

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

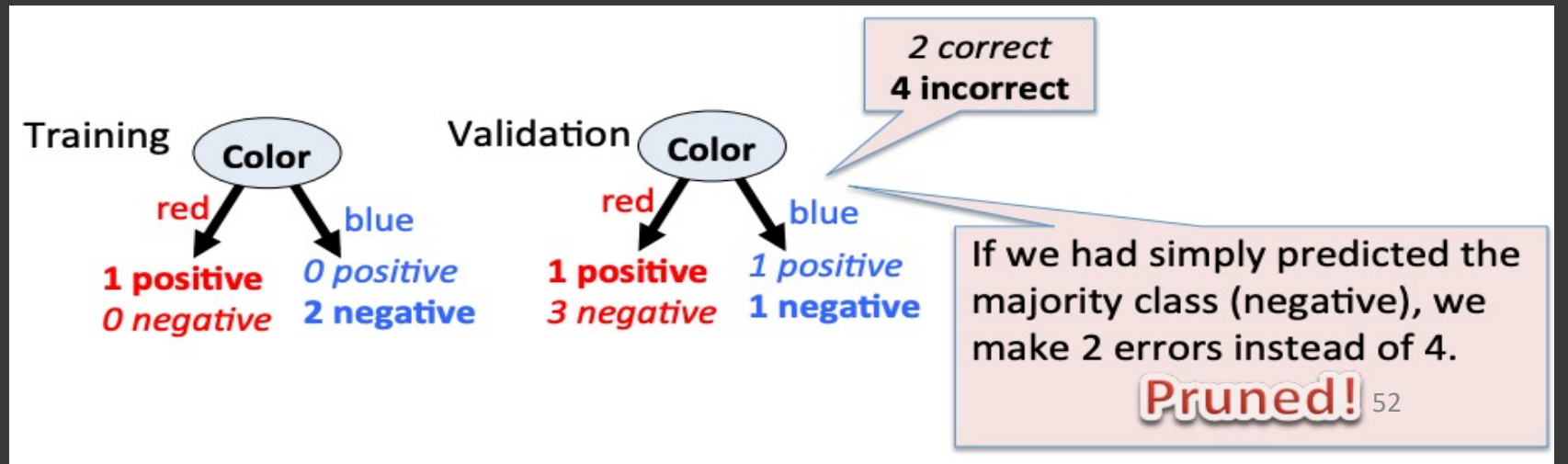
Slide credit: Pedro Domingos

Pruning Decision Trees

- Pruning of the decision tree is done *by replacing a whole subtree by a leaf node.*

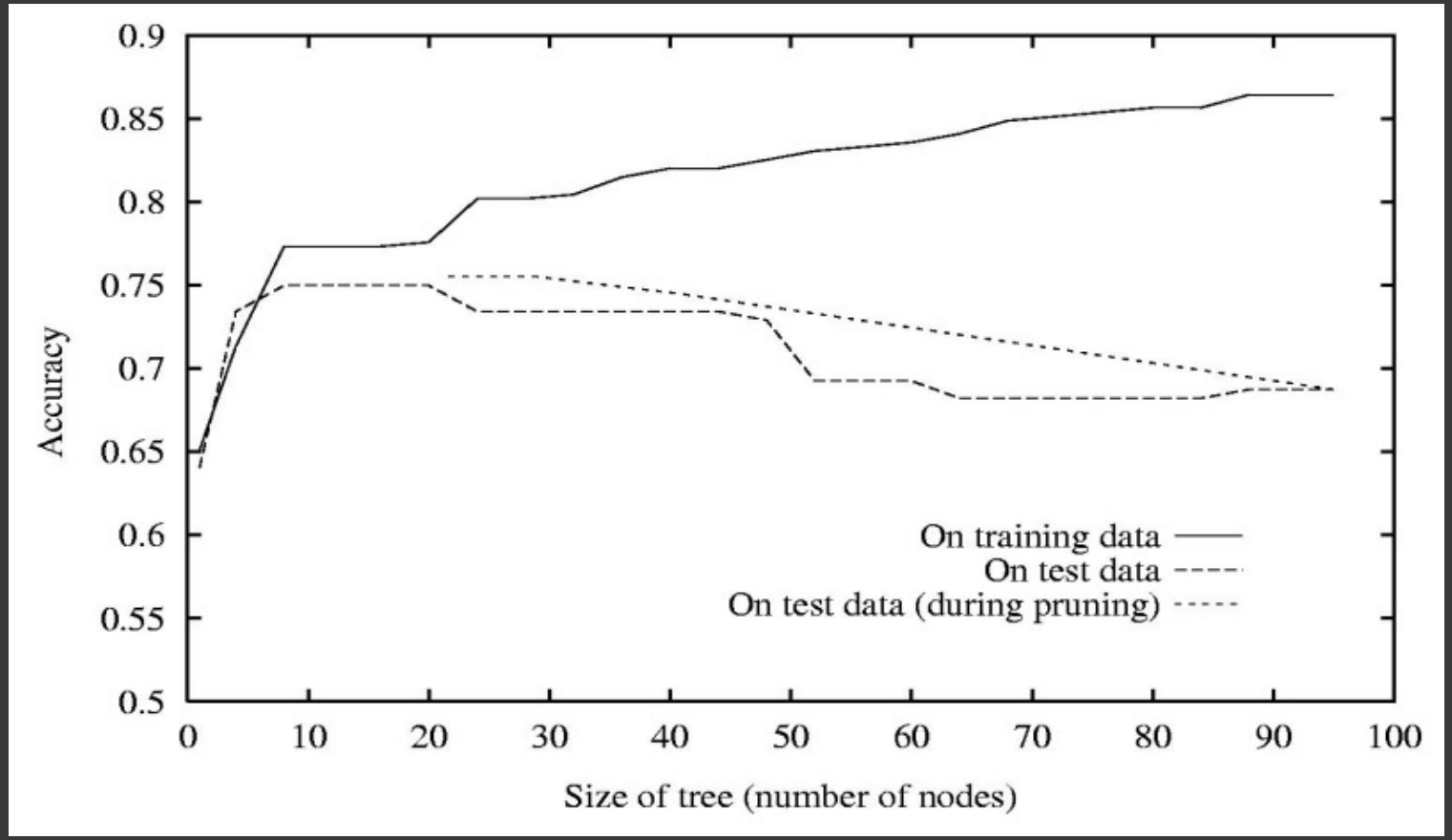
- The replacement takes place if a decision rule establishes *that the expected error rate in the subtree is greater than in the single leaf.*

- For example:



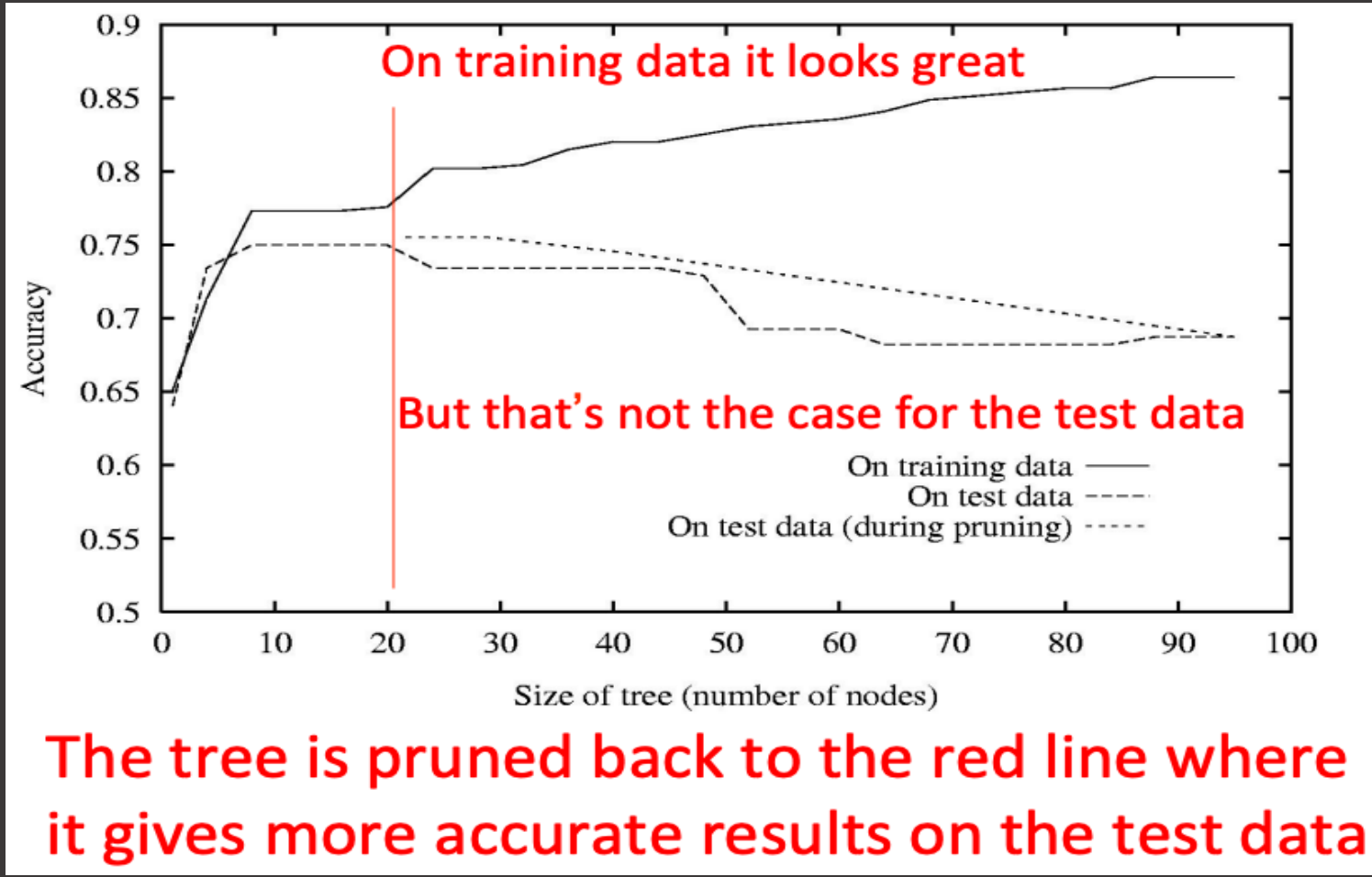
Slide credit: M. desJardins & T. Finin

Effect of Reduced-Error Pruning



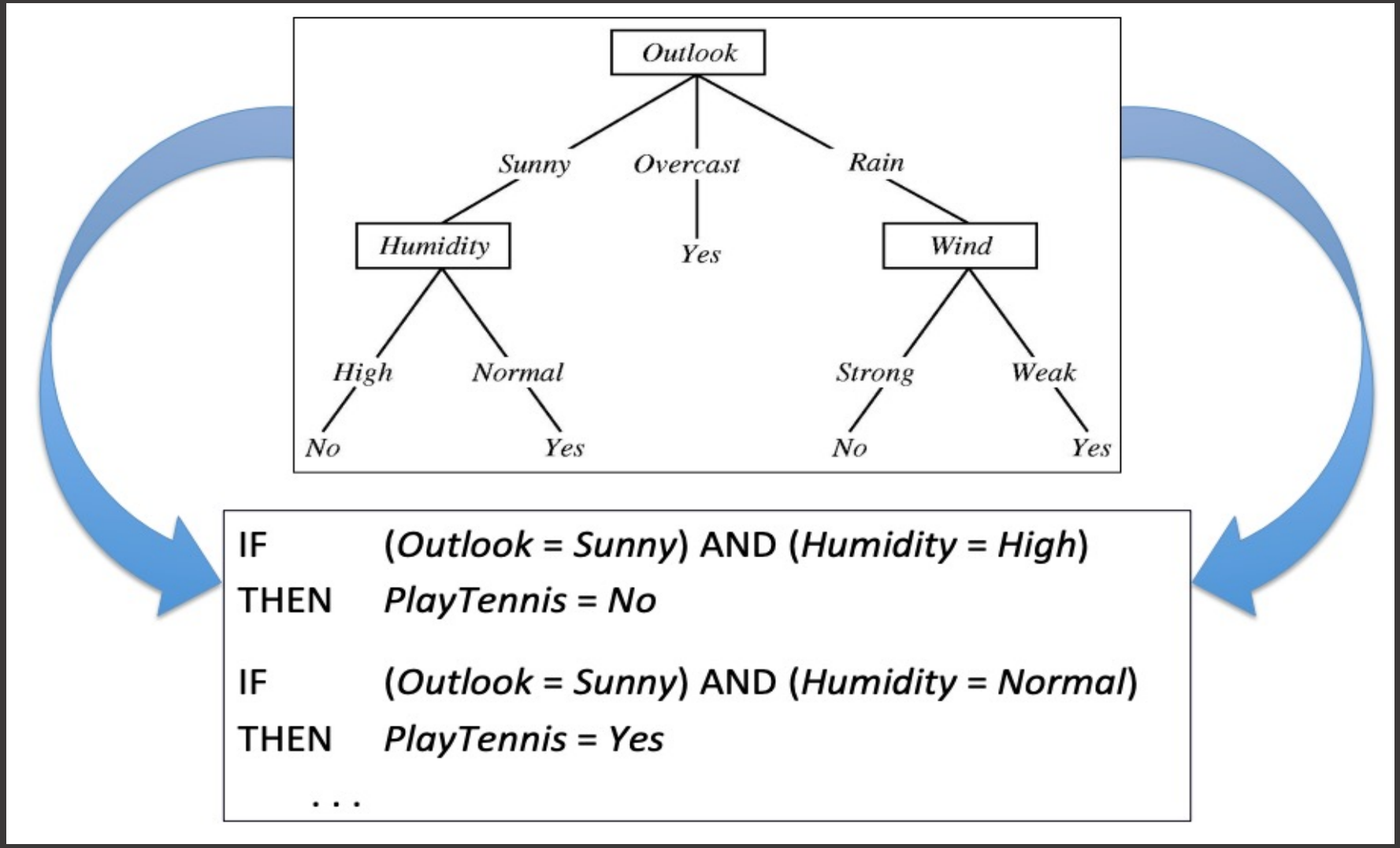
Slide credit:Pedro Domingos

Effect of Reduced-Error Pruning



Slide credit:Pedro Domingos

Converting a Tree to Rules



Scaling Up

- ID3: assumes that data fits in memory
(OK for up to hundreds of thousands of examples)

- SPRINT, SLIQ: multiple sequential scans of data
(OK for up to millions of examples)

- VFDT: at most one sequential scan
(OK for up to billions of examples)

Slide credit: Pedro Domingos

Comparison of Learning Methods

Characteristic	Neural Nets	SVM	Trees	MARS	k-NN, Kernels
Natural handling of data of "mixed" type	▼	▼	▲	▲	▼
Handling of missing values	▼	▼	▲	▲	▲
Robustness to outliers in input space	▼	▼	▲	▼	▲
Insensitive to monotone transformations of inputs	▼	▼	▲	▼	▼
Computational scalability (large N)	▼	▼	▲	▲	▼
Ability to deal with irrelevant inputs	▼	▼	▲	▲	▼
Ability to extract linear combinations of features	▲	▲	▼	▼	◆
Interpretability	▼	▼	◆	▲	▼
Predictive power	▲	▲	▼	◆	▲

Table 10.3 from Hastie, et al. Elements of Statistical Learning, 2nd Edition

Summary: Decision Tree Learning

- Representation: decision trees
- Bias: prefer small decision trees
- Search algorithm: greedy
- Heuristic function: information gain or information content or others
- Overfitting/pruning

Slide Credit: Pedro Domingos

Summary: Decision Tree Learning

- Widely used in practice
- Strengths:
  - 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)
  - Sacrifices predictive power

Slide Credit: Pedro Domingos