Data Mining:

3. Klassifikation

B) Decision Trees (cont.)

Classification by Decision Tree Induction

- Non-parametric approach: no assumptions on probability distribution of class and other attributes

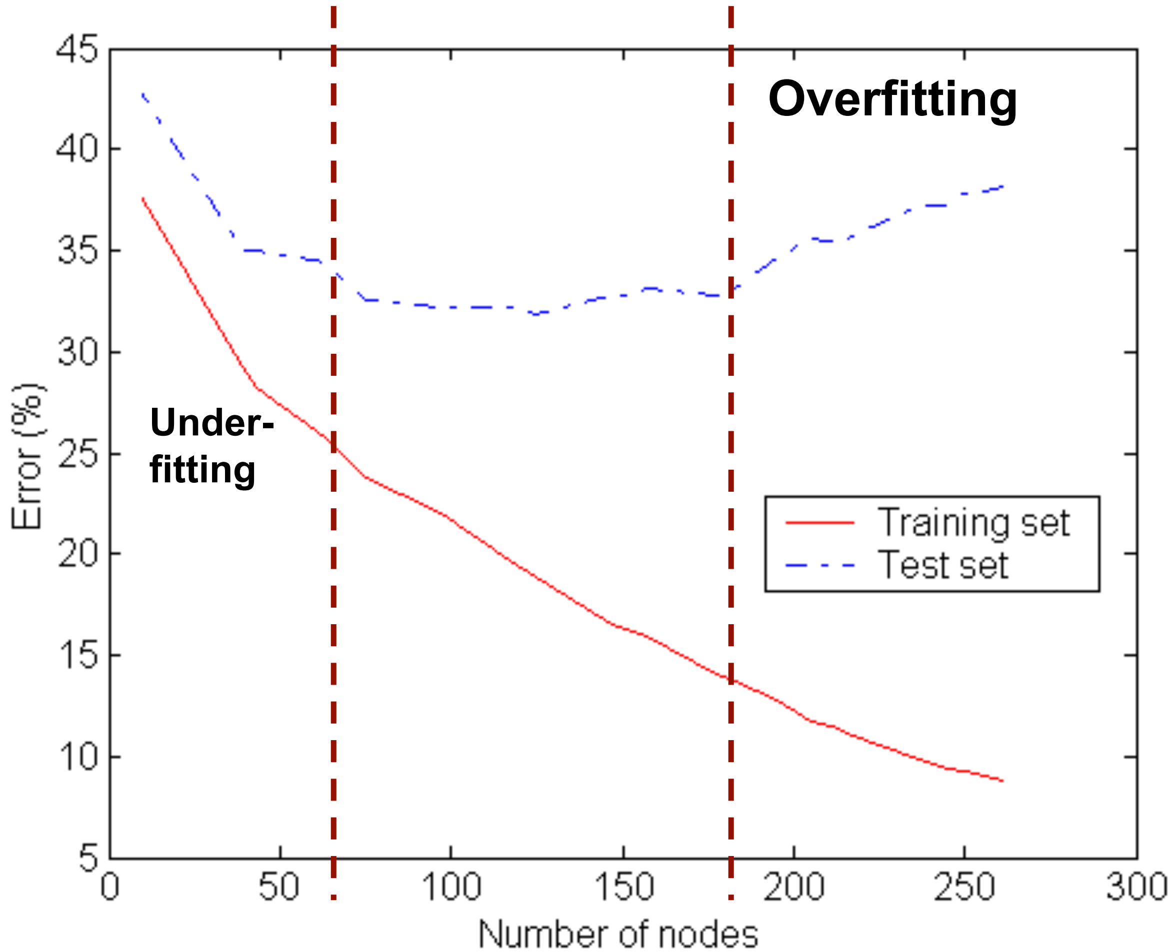
- Optimal decision tree construction is an NP-complete problem, but there are efficient heuristic-based algorithms.

  (here: greedy, top-down, recursive partitioning strategy for growing a decision tree; other search strategies ?)

- Robust against redundant attributes
  (only one of two strongly correlated attributes will be chosen for splitting)

- Remaining problems:
  - Underfitting and overfitting
  - Data Fragmentation
  - Missing values
  - Expressiveness

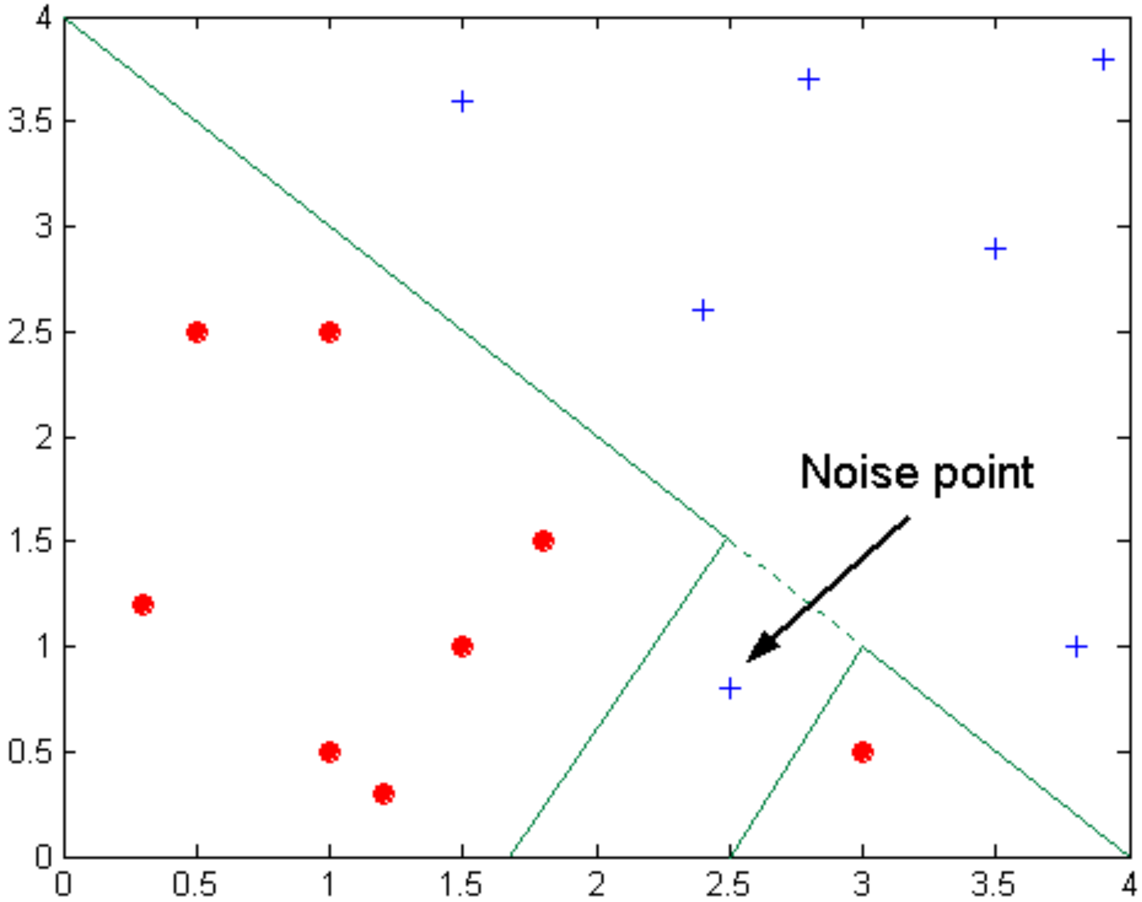
Underfitting and Overfitting



Underfitting: when model is too simple, both training and test errors are large

Overfitting: when model is too specific, test errors start to increase though training errors continue to decrease

Overfitting due to Noise



Decision boundary is distorted by noise point

Table 4.3. An example training set for classifying mammals. Class labels with asterisk symbols represent mislabeled records.

Name	Body Temperature	Gives Birth	Four-legged	Hibernates	Class Label
porcupine	warm-blooded	yes	yes	yes	yes
cat	warm-blooded	yes	yes	no	yes
bat	warm-blooded	yes	no	yes	no*
whale	warm-blooded	yes	no	no	no*
salamander	cold-blooded	no	yes	yes	no
komodo dragon	cold-blooded	no	yes	no	no
python	cold-blooded	no	no	yes	no
salmon	cold-blooded	no	no	no	no
eagle	warm-blooded	no	no	no	no
guppy	cold-blooded	yes	no	no	no

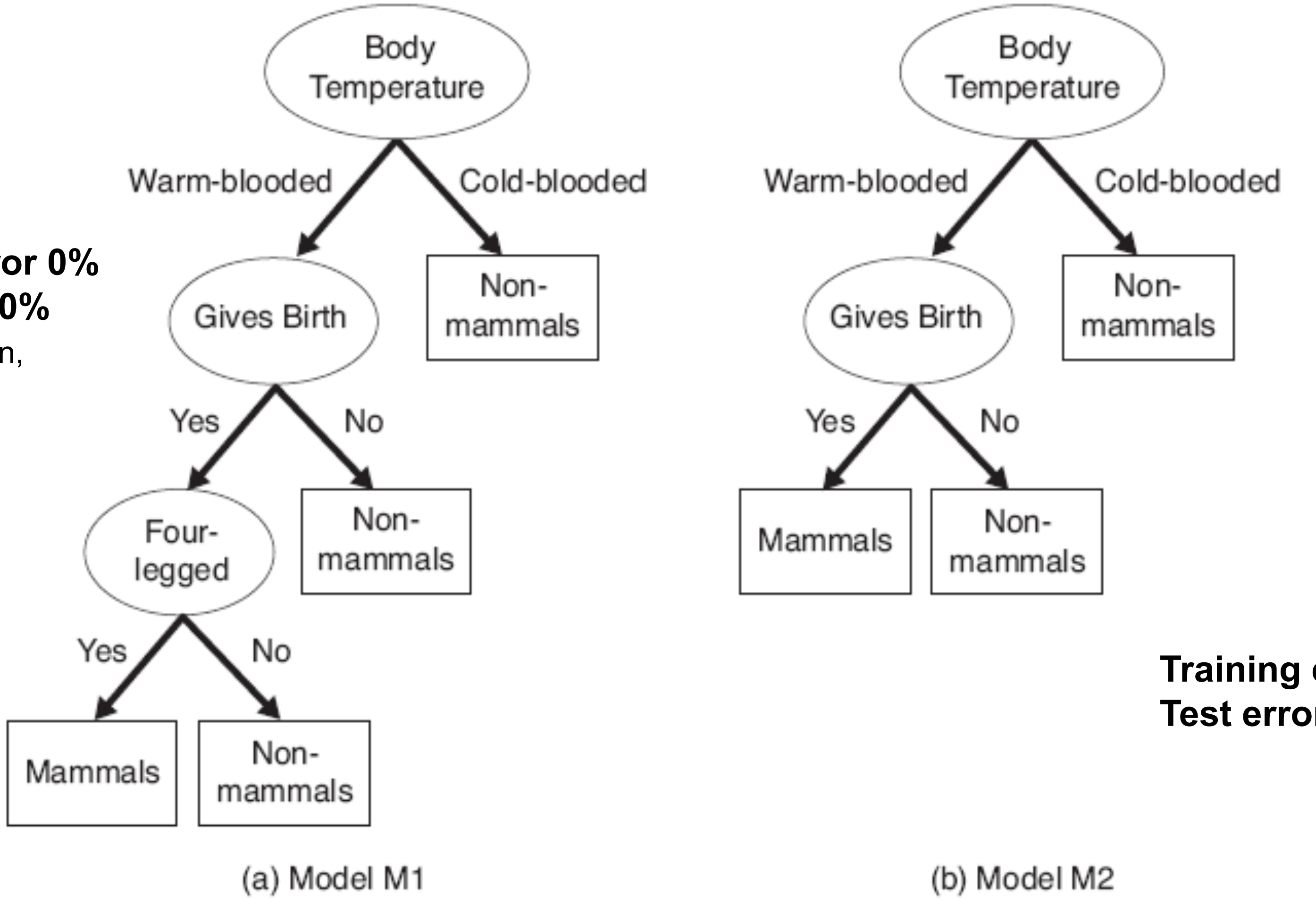
hibernate=
Winterschlaf
halten

Table 4.4. An example test set for classifying mammals.

Name	Body Temperature	Gives Birth	Four-legged	Hibernates	Class Label
human	warm-blooded	yes	no	no	yes
pigeon	warm-blooded	no	no	no	no
elephant	warm-blooded	yes	yes	no	yes
leopard shark	cold-blooded	yes	no	no	no
turtle	cold-blooded	no	yes	no	no
penguin	cold-blooded	no	no	no	no
eel	cold-blooded	no	no	no	no
dolphin	warm-blooded	yes	no	no	yes
spiny anteater	warm-blooded	no	yes	yes	yes
gila monster	cold-blooded	no	yes	yes	no

Overfitting due to Noise: An Example

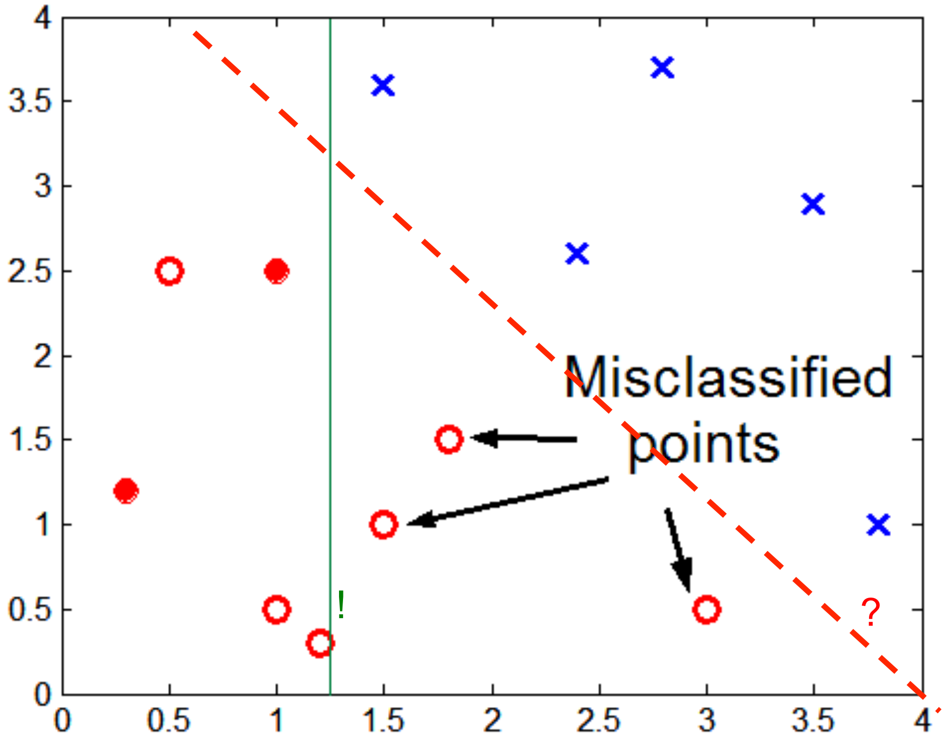
Training error 0%
Test error 30%
(human,dolphin,
anteater)



Training error 20%
Test error 10%

Figure 4.25. Decision tree induced from the data set shown in Table 4.3.

Overfitting due to Lack of Representative Samples



Lack of data points in the lower half of the diagram makes it difficult to predict correctly the class labels of that region.

Insufficient number of training records in the region causes the decision tree to predict the test examples using other training records that are irrelevant to the classification task.

Overfitting due to Lack of Representative Samples

Table 4.5. An example training set for classifying mammals.

Name	Body Temperature	Gives Birth	Four-legged	Hibernates	Class Label
salamander	cold-blooded	no	yes	yes	no
guppy	cold-blooded	yes	no	no	no
eagle	warm-blooded	no	no	no	no
poorwill	warm-blooded	no	no	yes	no
platypus	warm-blooded	no	yes	yes	yes

All warm-blooded creatures that do not hibernate will be classified as non-mammals, since there is only one training record with this characterictics.

Estimating Generalization Errors

- Re-substitution errors: error e(T) on training (for a tree T)
- Generalization errors: error e'(T) on testing…on previously unseen records

- Let e(t),e'(t) be the number of misclassified records at a leaf node t, n(T) the number of training records classified by T. $e(T) := \sum_{t \text{ leaf}} e(t) / n(T)$

- Methods for estimating generalization errors:
  - Optimistic approach: e'(T) = e(T)
  - Pessimistic approach:
    - For each leaf node: e'(t):= e(t)+penalty [e.g. 0.5 or 1]
    - Total error: e'(T):= e(T) + (nl(T)×penalty) / n(T)) [nl(T): number of leaf nodes]
    - Total a tree with 30 leaf nodes and 10 misclassified training records (out of 1000 instances), and penalty=0.5:

      Training error = 10/1000 = 1%

      Generalization error = 1%+(30×0.5 / 1000) = (10+30x0.5)/1000= 2.5%
  - Using a validation set:
    - Use a part of the training data set to estimate generalization error

Pessimistic Generalization Error: An Example

[Classes +,-]

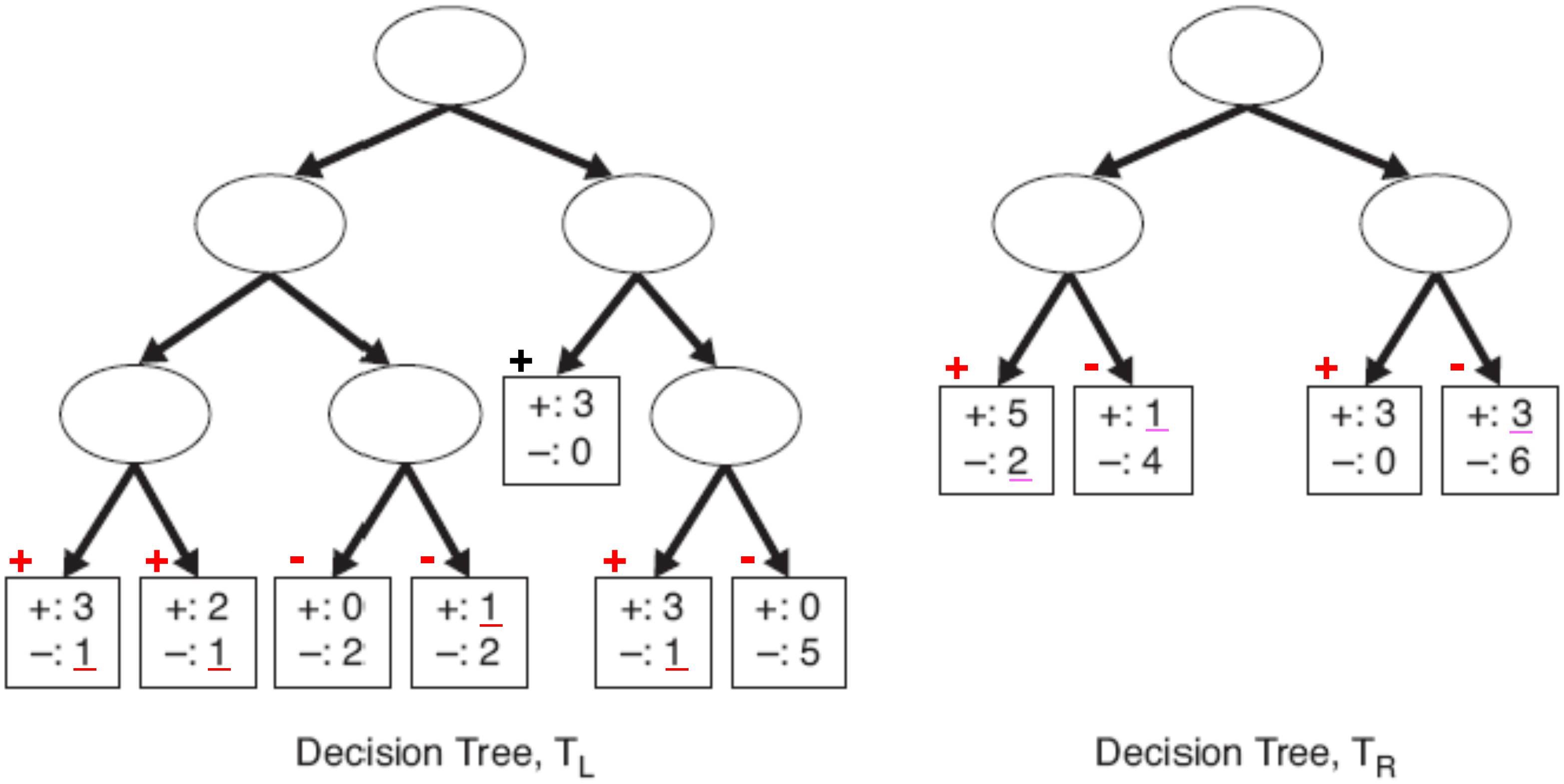


Figure 4.27. Example of two decision trees generated from the same training data.

Let: Training error: 4/24 = 0.167, penalty=1.0
Then: Pessimistic error= (4+7x1.0)/24= 0.458

Training error: 6/24 = 0.25, penalty=1.0
Pessimistic error= (6+4x1.0)/24= 0.417

Penalty 1.0|0.5 means: A node should not be split [nl(T):=nl(T)+1] unless it reduces the training error for >/>= 1 record.

Pessimistic Generalization Error: Review

- By considering the number of leaves, we have included model complexity when evaluating a model

- In the context of decision trees, it makes sense to prefer the simpler model over a similar, but more complex model, since:
- For complex models (trees), there is a greater chance that they were fitted accidentally by statistically insignificant data or by errors in data (noise)

How to Address Overfitting

- **Pre-Pruning (Early Stopping Rule)**
  - Stop the algorithm before it becomes a fully-grown tree
  - Typical stopping conditions for a node:
    - Stop if all instances belong to the same class
    - Stop if all the attribute values are the same
  - More restrictive conditions:
    - Stop if number of instances is less than some user-specified threshold
    - Stop if class distribution of instances are independent of the available features (e.g., using χ^2 test)
    - Stop if expanding the current node does not improve impurity measures (e.g., Gini or information gain).

How to Address Overfitting...

- **Post-Pruning**
  - Grow decision tree to its entirety
  - Trim the nodes of the decision tree in a bottom-up fashion by replacing a subtree with
    a) a new leaf node whose class label of leaf node is determined from majority class of instances in the sub-tree
    b) the most frequently used branch of the subtree (subtree raising)

    as long as generalization error improves after trimming.
  - More reliable than pre-pruning but more expensive.

Example of Post-Pruning

Class = Yes	20
Class = No	10

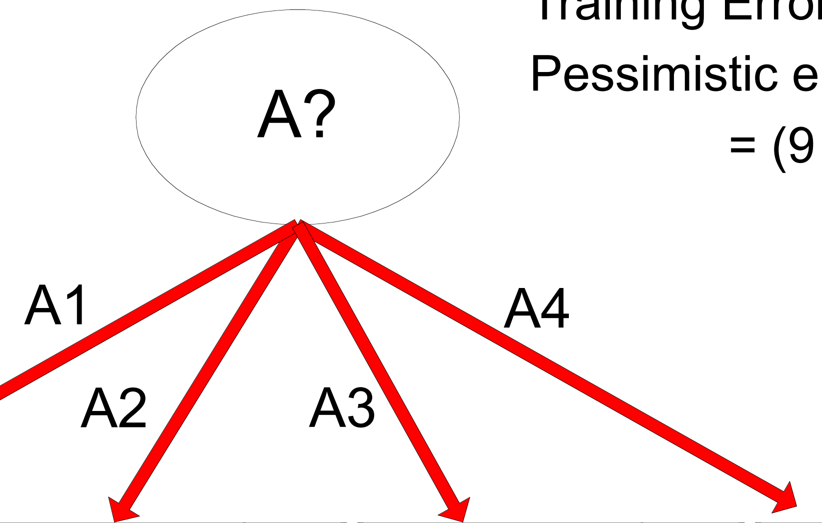
Training Error (before splitting) = 10/30

Pessimistic error = (10 + 0.5)/30 = 10.5/30

Training Error (after splitting) = 9/30

Pessimistic error (after splitting)

= (9 + 4 × 0.5)/30 = 11/30



Class = Yes	8	Class = Yes	3	Class = Yes	4	Class = Yes	5
Class = No	4	Class = No	4	Class = No	1	Class = No	1

Prune! *(replace A-subtree by leaf node above)*

Post-pruning in Web Robot Application

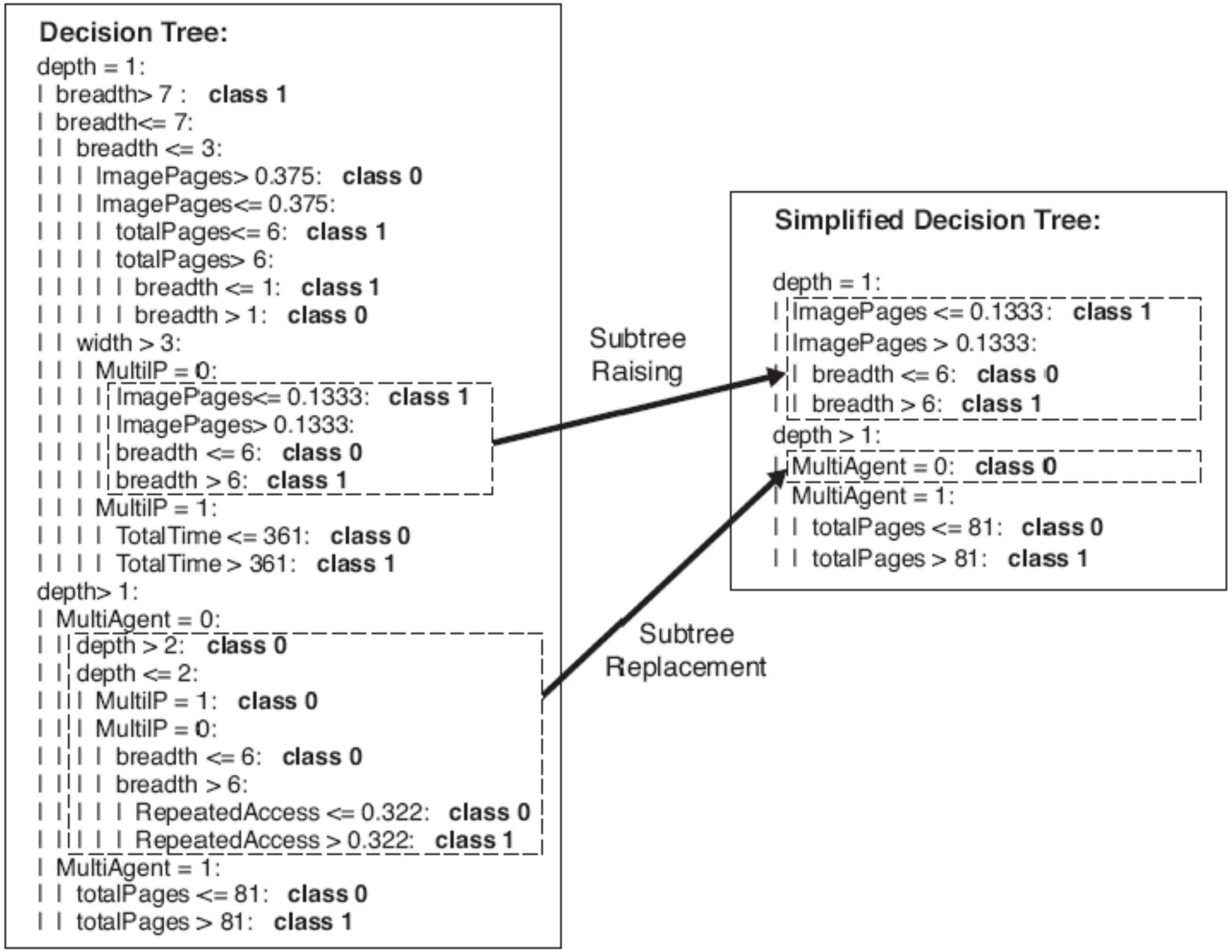


Figure 4.29. Post-pruning of the decision tree for Web robot detection.

Data Fragmentation

- Number of instances gets smaller as you traverse down the tree
- Number of instances at the leaf nodes could be too small to make any statistically significant decision

- Can be avoided by restrictive stopping conditions (compare pre-pruning)

Missing Attribute Values

- Missing values affect decision tree construction in three different ways:
  - Affects how impurity measures are computed
  - Affects how to distribute instance with missing value to child nodes
  - Affects how a test instance with missing value is classified

Missing Attribute Values: Computing Impurity Measure

Tid	Refund	Marital Status	Taxable Income	Class
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	?	Single	90K	Yes

Missing value

Before Splitting:

Entropy(Parent)
= -(0.3)log(0.3) – (0.7)log(0.7) = 0.8813

	Class = Yes	Class = No
Refund=Yes	0	3
Refund=No	2	4
Refund=?	1	0

Split on Refund:

Entropy(Refund=Yes) = 0

Entropy(Refund=No)
= -(2/6)log(2/6) – (4/6)log(4/6) = 0.9183

Entropy(Children)
= 0.3 × 0 + 0.6 × 0.9183 = 0.551

Gain = **0.9 ×** (0.8813 – 0.551) = 0.3303

Missing Attribute Values: Distribute Training Instances

Tid	Refund	Marital Status	Taxable Income	Class
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No

Refund

Yes

Class=Yes	0
Class=No	3

No

Class=Yes	2
Class=No	4

Tid	Refund	Marital Status	Taxable Income	Class
10	?	Single	90K	Yes

Refund

Yes

Class=Yes	0 + 3/9
Class=No	3

No

Class=Yes	2 + 6/9
Class=No	4

Probability that Refund=Yes is 3/9

Probability that Refund=No is 6/9

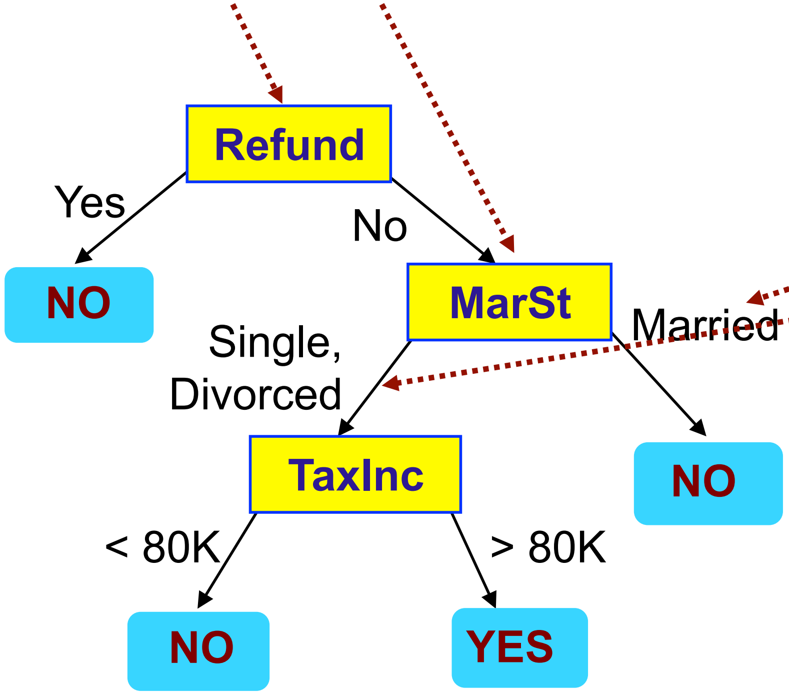
Assign record with missing value
to the left child with weight = 3/9 and
to the right child with weight = 6/9

Missing Attribute Values: Classify New Instances

New record:

Tid	Refund	Marital Status	Taxable Income	Class
11	No	?	85K	?

	Married	Single	Divorced	Total
Class=No	3	1	0	4
Class=Yes	0	1+6/9	1	2.67
Total	3	2.67	1	6.67



Follow several branches and **do bookkeeping of probabilities** for the branches:

Probability that Marital Status = Married is 3/6.67

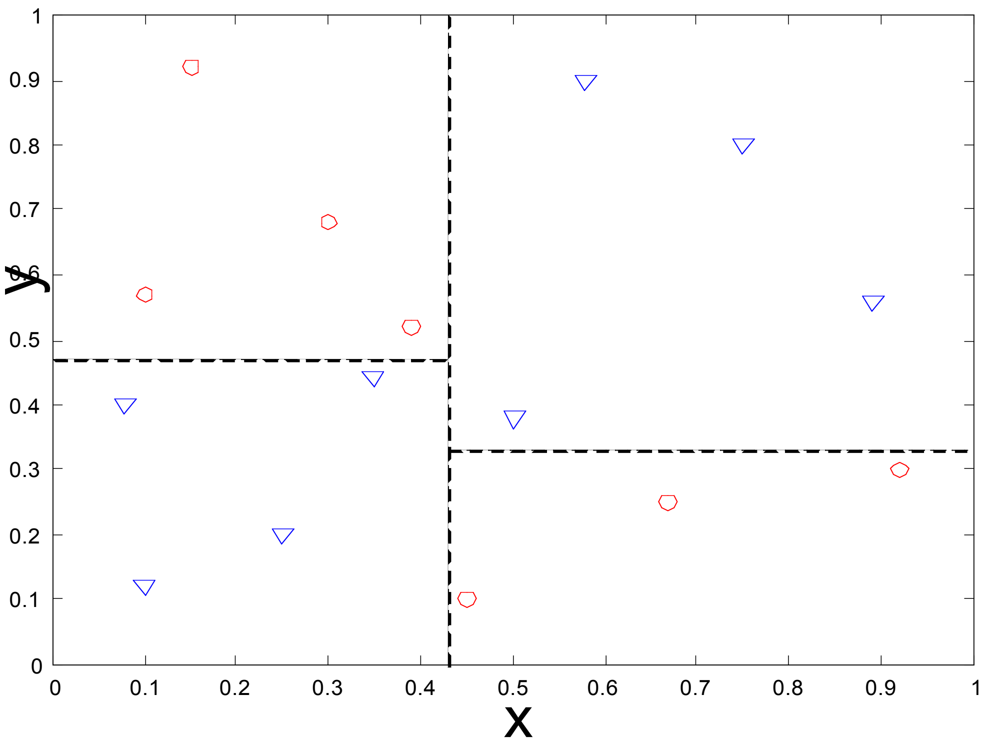
Probability that Marital Status = {Single,Divorced} is 3.67/6.67

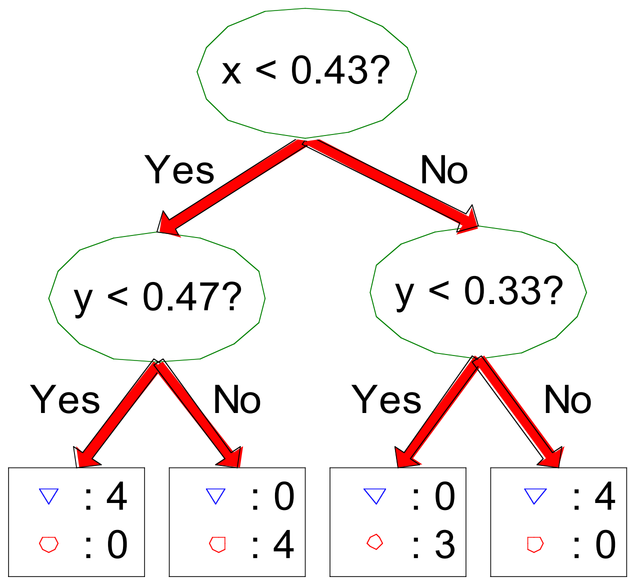
Expressiveness

- Decision tree provides expressive representation for learning discrete-valued function
  - But they do not generalize well to certain types of Boolean functions
    - Example: parity function:
      - Class = 1 if there is an even number of Boolean attributes with truth value = True
      - Class = 0 if there is an odd number of Boolean attributes with truth value = True
    - For accurate modeling, you must have a complete tree

- Not expressive enough for modeling continuous variables
  - Particularly when test condition involves only a single attribute at-a-time

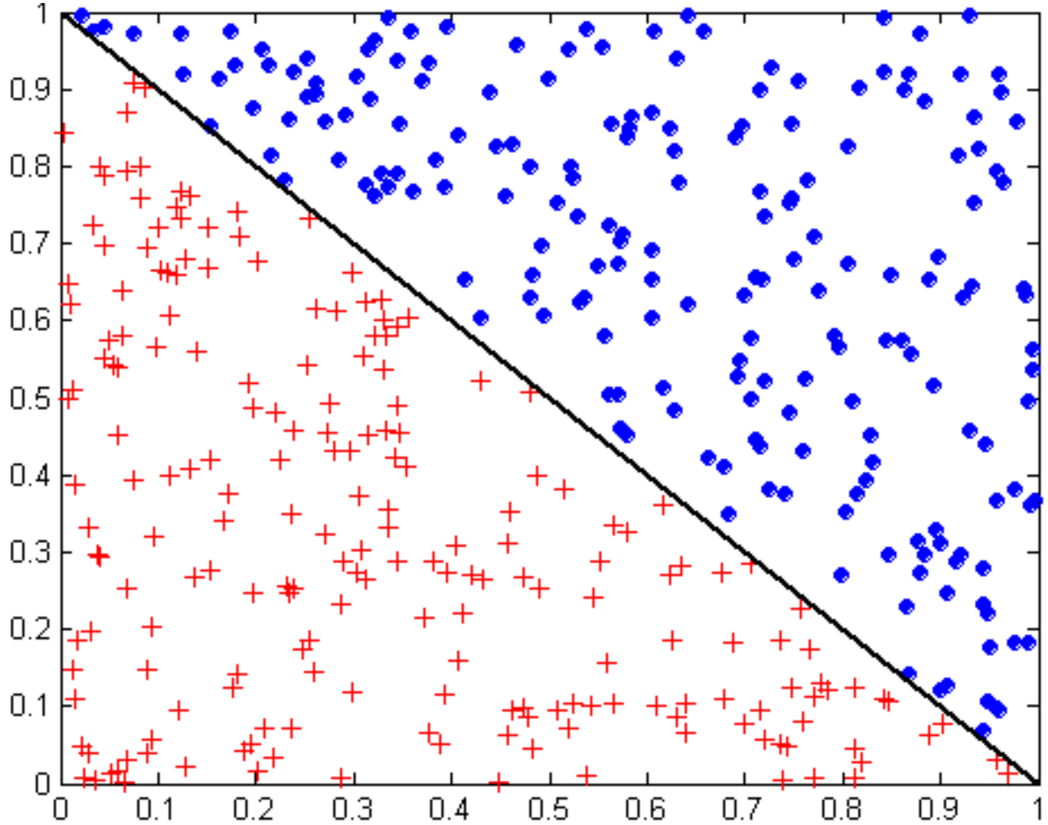
Expressiveness: Decision Boundary

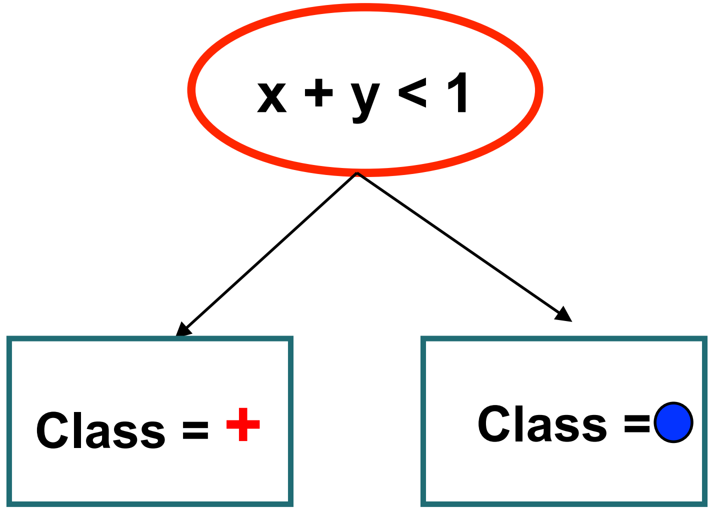




- Border line between two neighboring regions of different classes is known as decision boundary
- Decision boundary can be only parallel to axes (rectilinear) because test condition involves a single attribute at-a-time

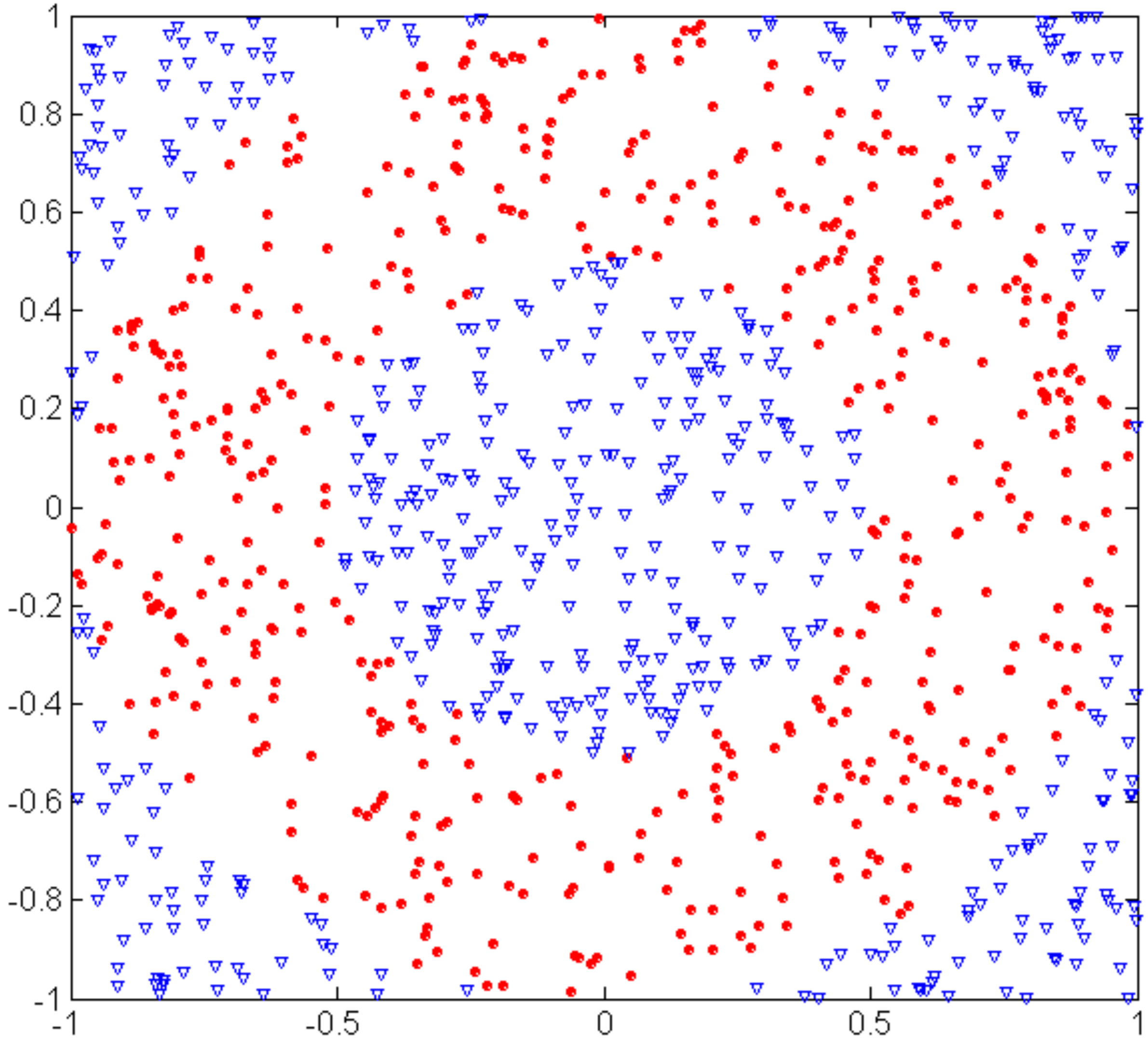
Expressiveness: Decision Boundary





- **Test condition should involve multiple attributes !**
- More expressive representation
- But finding optimal test condition is computationally expensive

Expressiveness: Decision Boundary



Circular points:

$0.5 \leq \text{sqrt}(x_1^2+x_2^2) \leq 1$

Triangular points:

$\text{sqrt}(x_1^2+x_2^2) < 0.5$ or

$\text{sqrt}(x_1^2+x_2^2) > 1$