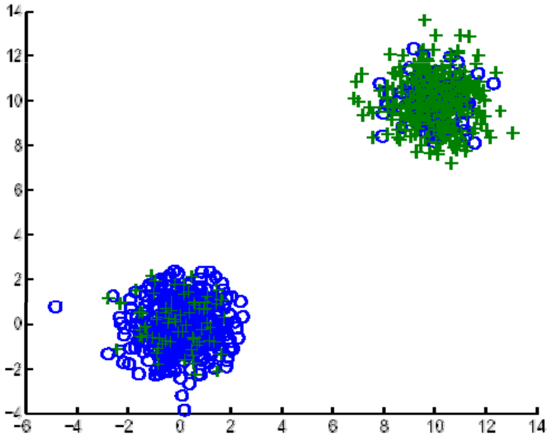
Decision Trees
(Part II: Pruning the tree)

nanopoulos@ismll.de

Underfitting and Overfitting



2000 points in two classes (1000 per class)

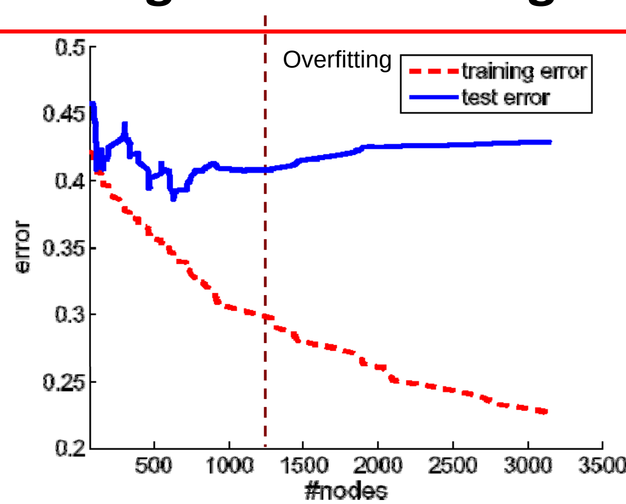
Swap 150 points between the classes

1000 training/1000 test

Swap additional 200 in training set

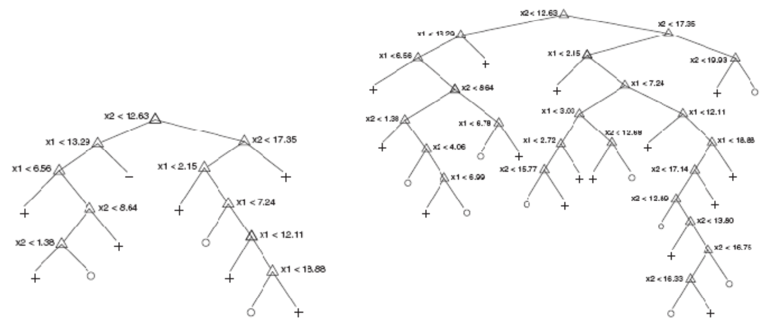
3

Underfitting and Overfitting



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

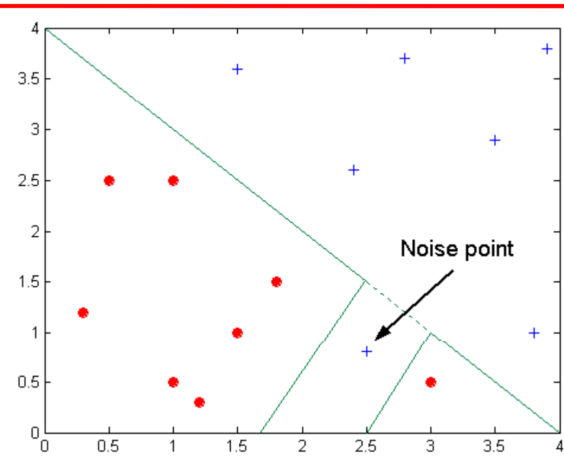
4



(a) Decision tree with 11 leaf nodes.

(b) Decision tree with 24 leaf nodes.

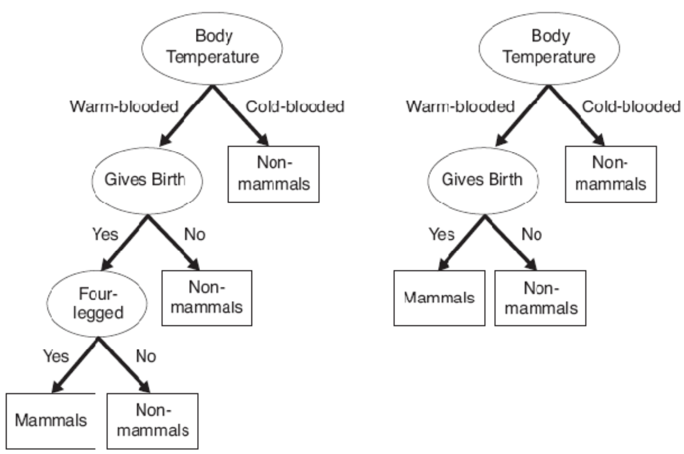
Overfitting due to Noise



Decision boundary is distorted by noise point

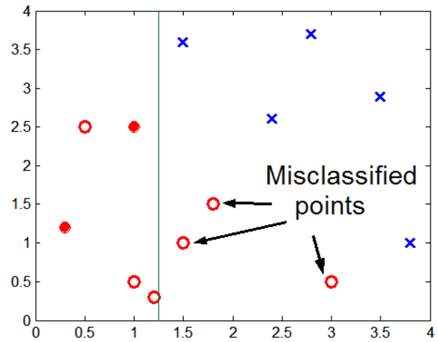
Name	Body Temperature	Gives Birth	Four-legged	Hibernates	Class Label
porcupine	warm-blooded	yes	yes	yes	yes
cat	warm-blooded	yes	yes	no	yes
bat	warm-blooded	yes	no	yes	no*
whale	warm-blooded	yes	no	no	no*
salamander	cold-blooded	no	yes	yes	no
komodo dragon	cold-blooded	no	yes	no	no
python	cold-blooded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

7



8

Overfitting due to Insufficient Examples

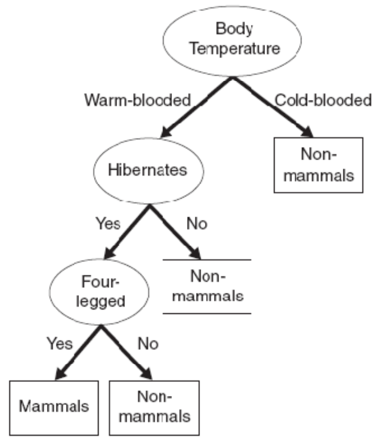


Lack of data points in the lower half of the diagram makes it difficult to predict correctly the class labels of that region

- 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

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

10



11

Notes on Overfitting

Overfitting results in decision trees that are more complex than necessary

Training error no longer provides a good estimate of how well the tree will perform on previously unseen records

Need new ways for estimating errors

Estimating Generalization Errors

Re-substitution errors: error on training (Σ e(t))
Generalization errors: error on testing (Σ e'(t))

Methods for estimating generalization errors:

- Optimistic approach: e'(t) = e(t)
- Pessimistic approach:
  - For each leaf node: e'(t) = (e(t)+0.5)
  - Total errors: e'(T) = e(T) + N $\times$ 0.5 (N: number of leaf nodes)
  - For a tree with 30 leaf nodes and 10 errors on training (out of 1000 instances):
    - Training error = 10/1000 = 1%
    - Generalization error = (10 + 30$\times$0.5)/1000 = 2.5%
- Reduced error pruning (REP):
  - uses validation data set to estimate generalization error

Occam's Razor

Given two models of similar generalization errors, one should prefer the simpler model over the more complex model

For complex models, there is a greater chance that it was fitted accidentally by errors in data

Therefore, one should include model complexity when evaluating a model

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
- If generalization error improves after trimming, replace sub-tree by a leaf node.
- Class label of leaf node is determined from majority class of instances in the sub-tree
- Can use MDL for post-pruning

Example of Post-Pruning

Class = Yes	20
Class = No	10
Error = 10/30	

Training Error (Before splitting) = 10/30

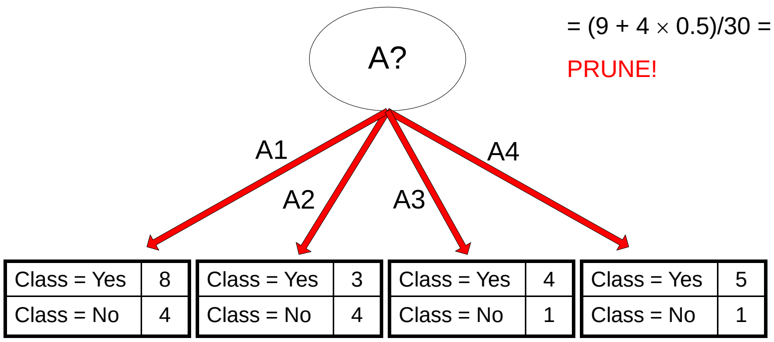
Pessimistic error = (10 + 0.5)/30 = 10.5/30

Training Error (After splitting) = 9/30

Pessimistic error (After splitting)

= (9 + 4 × 0.5)/30 = 11/30

PRUNE!



Reduced Error Pruning

Quinlan 1978

Mingers 1978

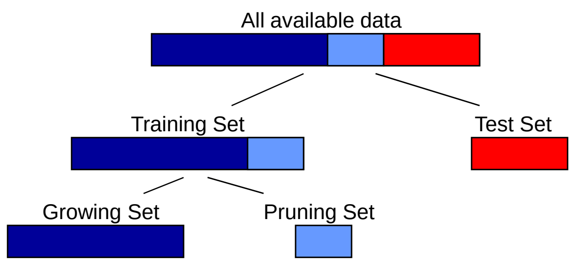
Esposito et al. 1996

Elomaa & Kaariainen 2001

Partitioning Data in Tree Induction

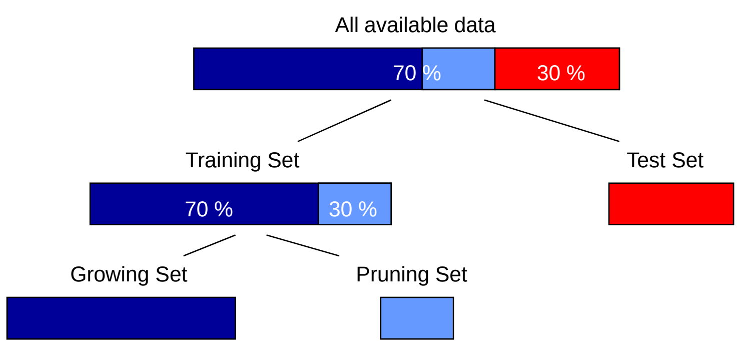
Estimating accuracy of a tree on new data: “Test Set”

Some post pruning methods need an independent data set: “Pruning Set”



To evaluate the classification technique, experiment with repeated random splits of data

Typical Proportions

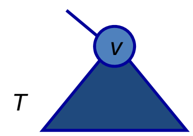


Problem with using “Pruning Set”: less data for “Growing Set”

Reduced Error Pruning (REP)

Use pruning set to estimate accuracy of sub-trees and accuracy at individual nodes

Let *T* be a sub-tree rooted at node *v*



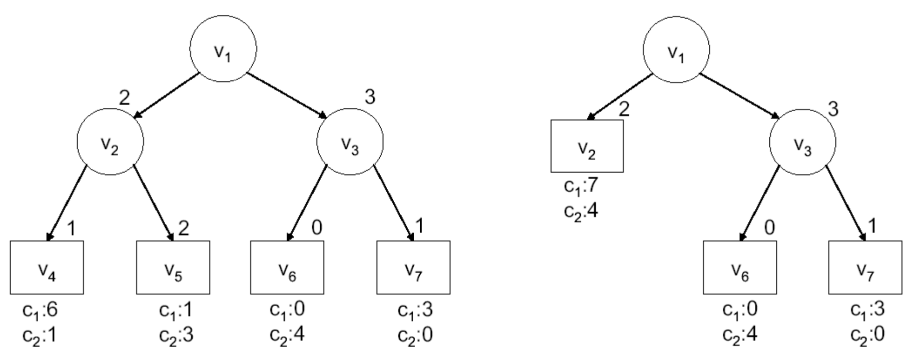
Define:

$$\text{Gain from prunning at } v = \#\text{misclassification in } T - \#\text{misclassification at } v$$

Repeat: prune at node with largest gain until until only negative gain nodes remain

"Bottom-up restriction": *T* can only be pruned if it does not contain a sub-tree with lower error than *T*

REP example



$$E(T_{v_2}) = 3, E(v_2) = 2, E(T_{v_3}) = 1, E(v_3) = 3.$$

Real implementations

name	ChAID	CART	ID3	C4.5
author	Kass 1980	Breiman et al. 1984	Quinlan 1986	Quinlan 1993
selection measure	χ^2	Gini index, twoing index	information gain	information gain ratio
splits	all	binary nominal, binary quantitative, binary bivariate quantitative	complete	complete, binary nominal, binary quantitative
stopping criterion	χ^2 independence test	minimum number of cases/node	χ^2 independence test	lower bound on selection measure
pruning technique	none	error complexity pruning	pessimistic error pruning	pessimistic error pruning, error based pruning

Example: C4.5

Simple depth-first construction.

Uses Information Gain

Sorts Continuous Attributes at each node.

Needs entire data to fit in memory.

Unsuitable for Large Datasets.

Needs out-of-core sorting.

You can download the software from:
http://www.cse.unsw.edu.au/~quinlan/c4.5r8.tar.gz