Decision Tree: Issues

CSE 5334 Data Mining
Spring 2020

Won Hwa Kim

(Slides courtesy of Pang-Ning Tan, Michael Steinbach and Vipin Kumar, and Jiawei Han, Micheline Kamber and Jian Pei)

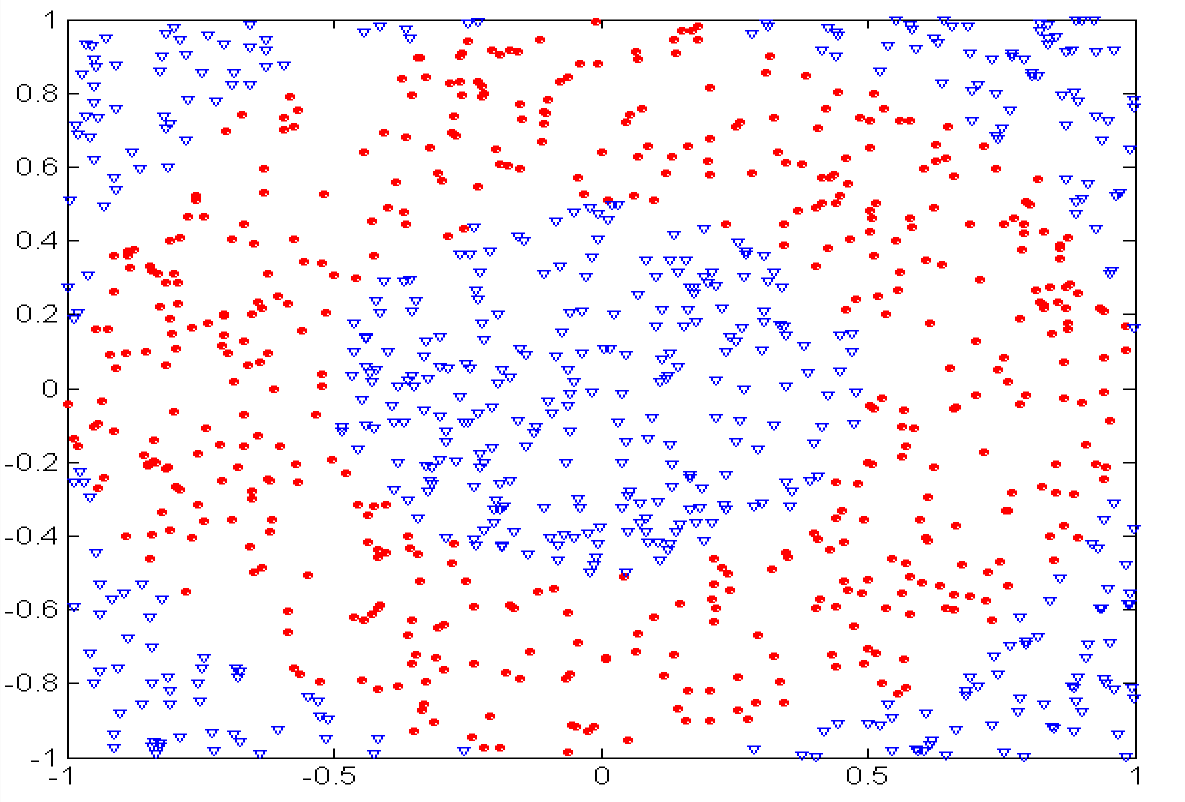
Practical Issues of Classification

Underfitting and Overfitting

Missing Values

Costs of Classification

Underfitting and Overfitting



500 circular and 500 triangular data points.

Circular points:

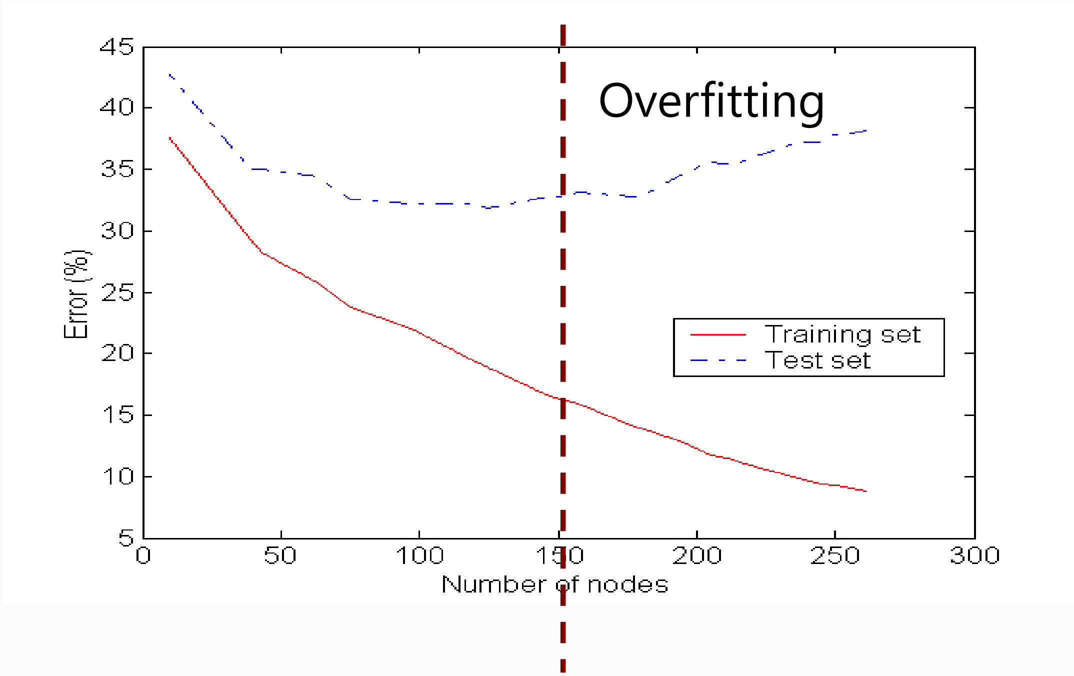
$0.5 \leq \text{sqrt}(x_1^2+x_2^2) \leq 1$

Triangular points:

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

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

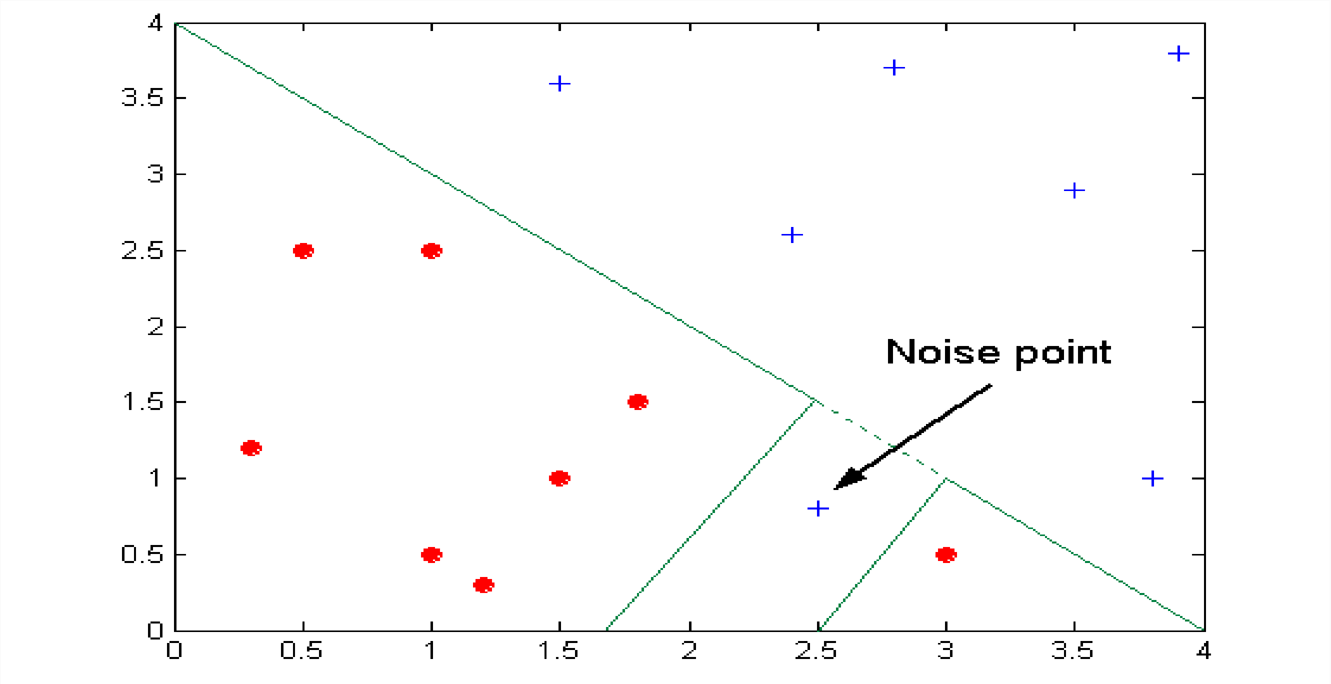
Underfitting and Overfitting



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

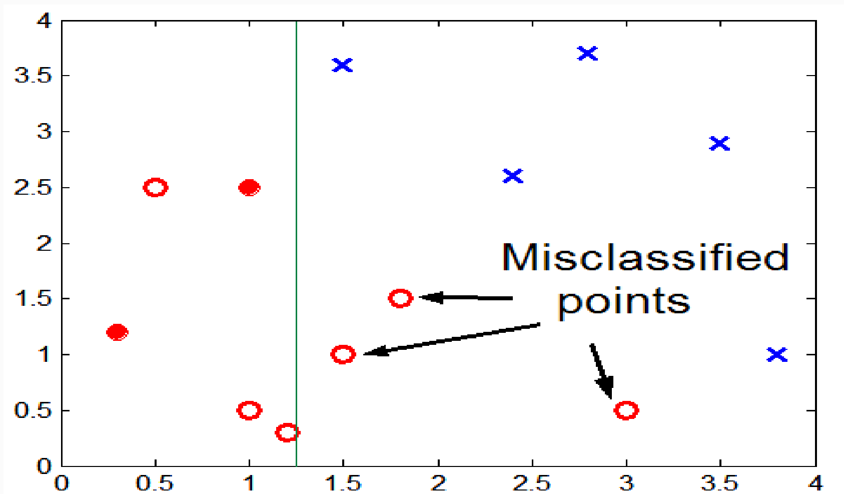
Overfitting: when model is too complex, test error increases even though training error decreases

Overfitting due to Noise



Decision boundary is distorted by noise point

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

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 × 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 + 30×0.5)/1000 = 2.5%
- Reduced error pruning (REP):
  - uses validation data set to estimate generalization error

Occam's Razor

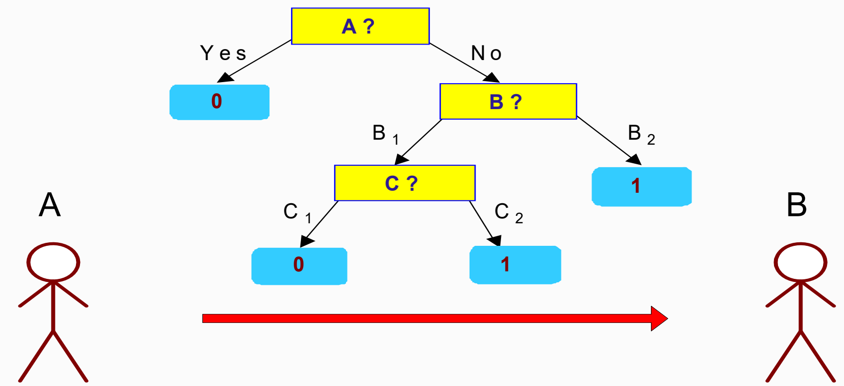
Given two models of similar generalization errors, one should prefer the simpler model over the more complex model

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

Minimum Description Length (MDL)

X	y
X_1	1
X_2	0
X_3	0
X_4	1
…	…
X_n	1



X	y
X_1	?
X_2	?
X_3	?
X_4	?
…	…
X_n	?

Cost(Model,Data) = Cost(Data|Model) + Cost(Model)

- Cost is the number of bits needed for encoding.
- Search for the least costly model.

Cost(Data|Model) encodes the misclassification errors.

Cost(Model) uses node encoding (number of children) plus splitting condition encoding.

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

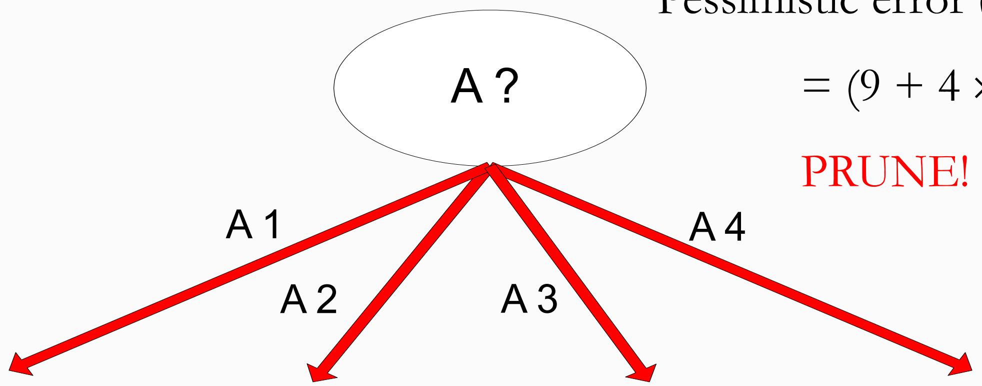
Pessimistic error = (10 + 0.5)/30 = 10.5/30

Training Error (After splitting) = 9/30

Pessimistic error (After splitting)

$$= (9 + 4 \times 0.5)/30 = 11/30$$

PRUNE!



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

Examples of Post-pruning

- Optimistic error?

  Don't prune for both cases

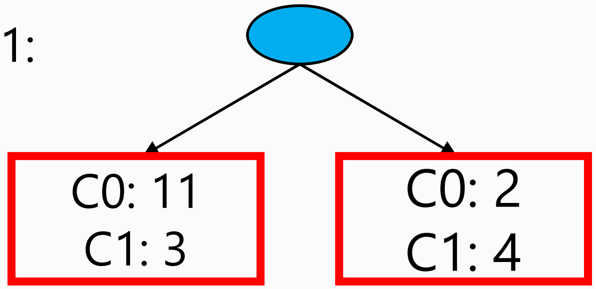
- Pessimistic error?

  Don't prune case 1, prune case 2

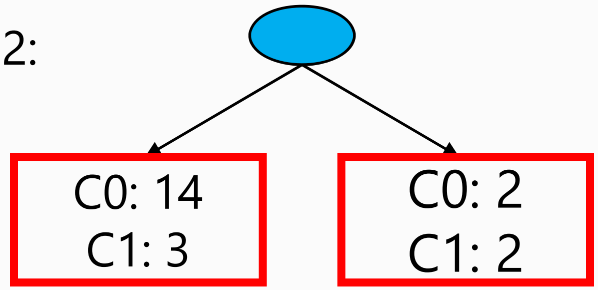
- Reduced error pruning?

  Depends on validation set

Case 1:



Case 2:

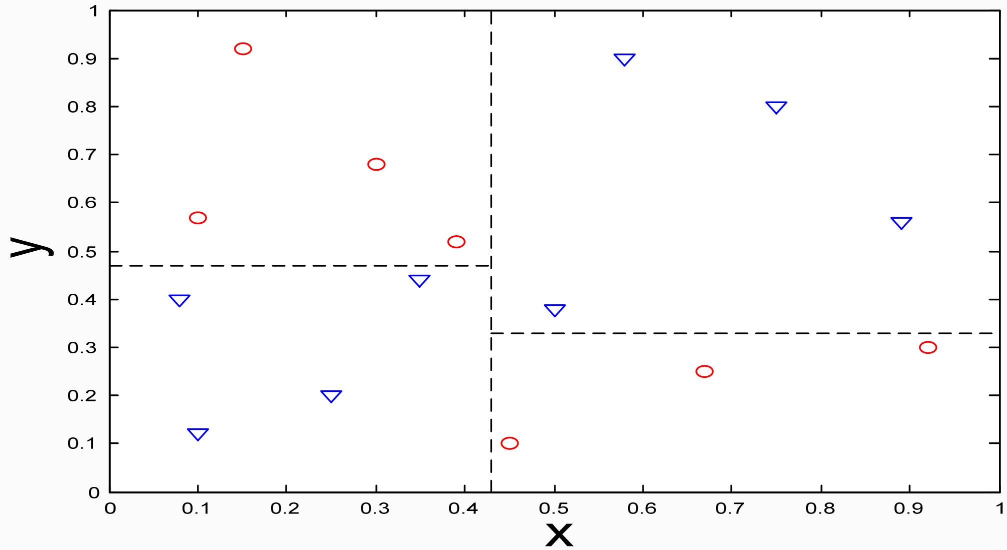


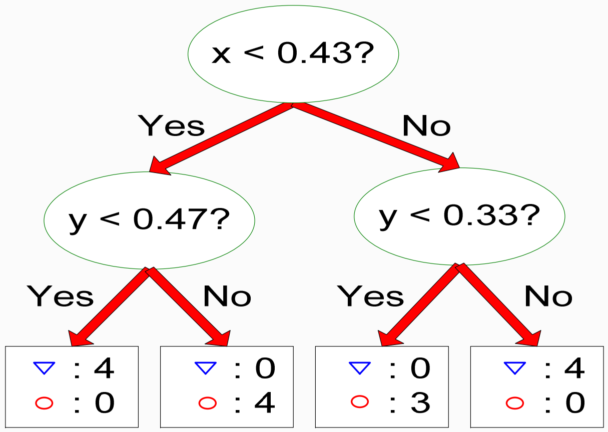
Handling Missing Attribute Values

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

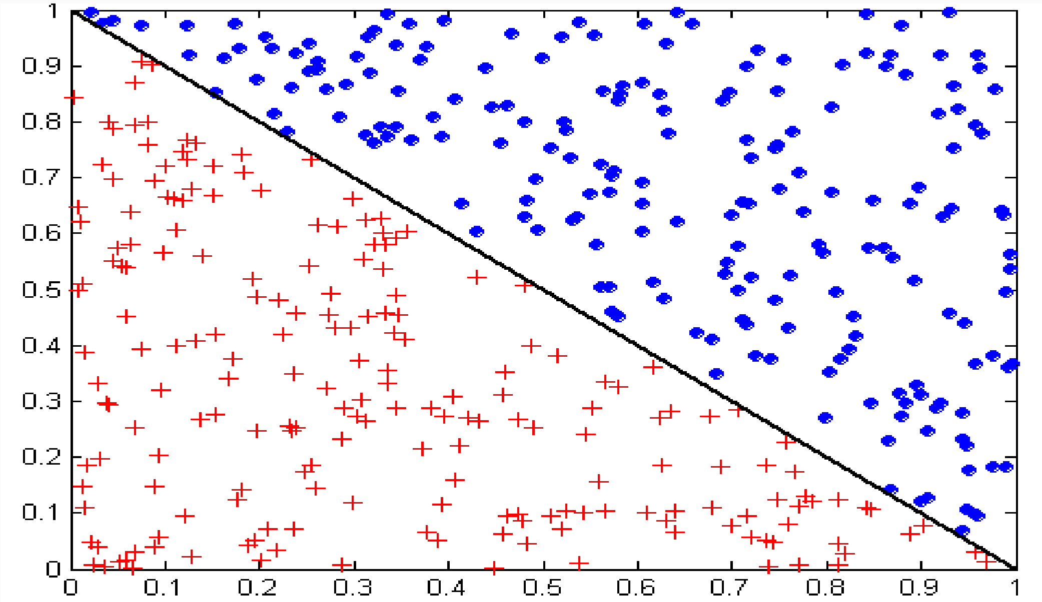
Decision Boundary

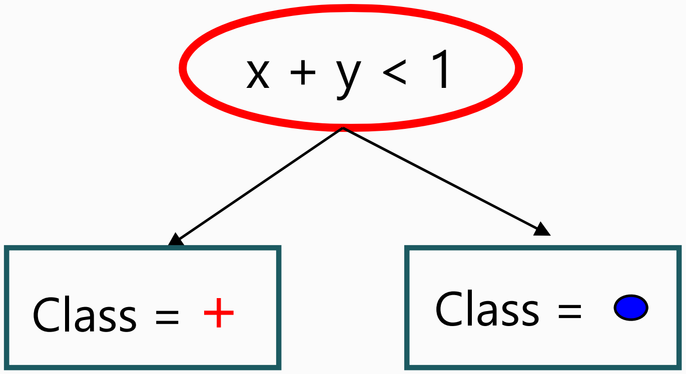




- Border line between two neighboring regions of different classes is known as decision boundary
- Decision boundary is parallel to axes because test condition involves a single attribute at-a-time

Oblique Decision Trees





- Test condition may involve multiple attributes
- More expressive representation
- Finding optimal test condition is computationally expensive