Decision Trees

Course: Data Mining

Professor: Dr. Tahaei

Author: Nazanin Zarei

Subject: Decision Trees and Overfitting

December 2024

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 is 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).

Post-pruning

- Grow the 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.
- The class label of the leaf node is determined from the majority class of instances in the sub-tree.
- Can use MDL (Minimum Description Length) 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!

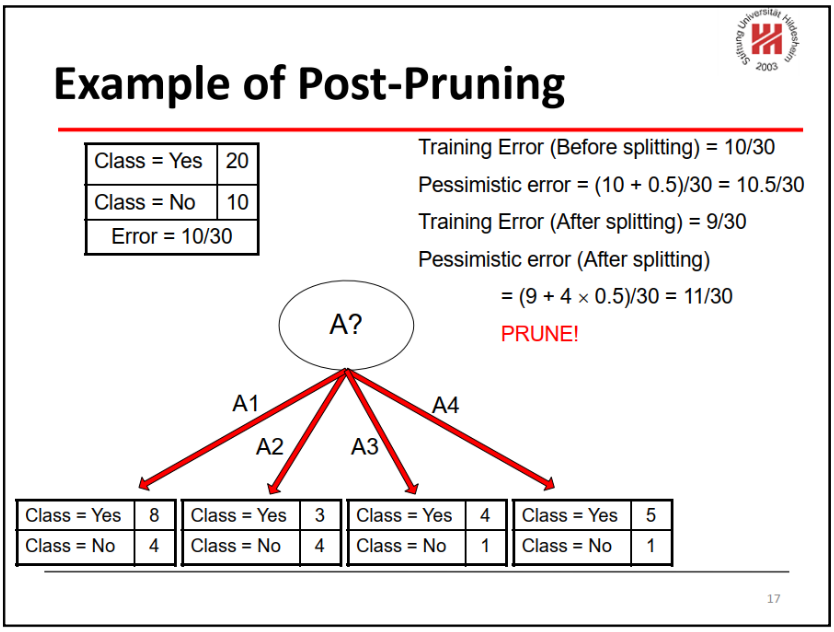

Figure 1: Example of Post-Pruning

Partitioning Data in Tree Induction

Estimating the accuracy of a tree on new data: "Test Set".
Some post-pruning methods need an independent data set: "Pruning Set".

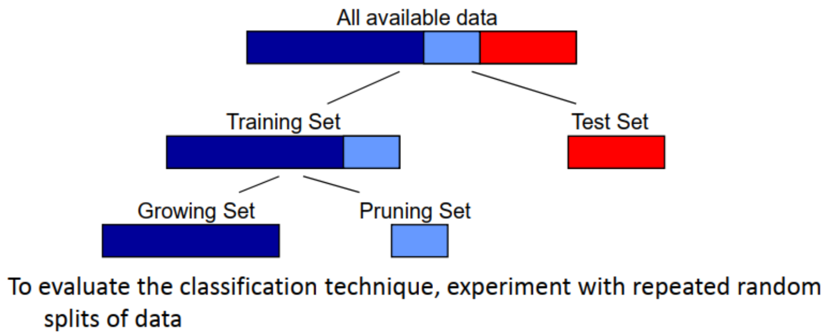

Figure 2: Partitioning Data in Tree Induction

Typical Proportions

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

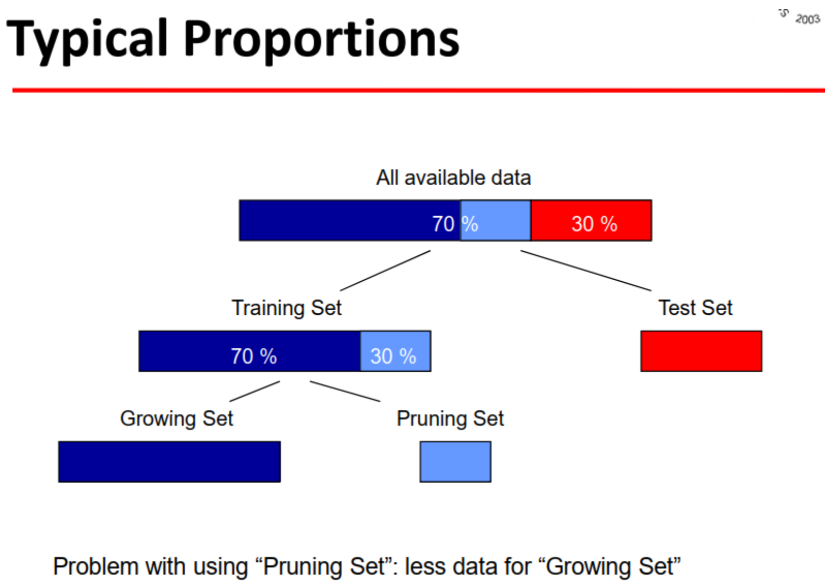

Figure 3: Typical Proportions in Data Splitting

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 the gain from pruning at v:

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

- Repeat: prune at node with largest gain 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

$$E\left(T_{v_2}\right) = 3,\ E\left(v_2\right) = 2,\ E\left(T_{v_3}\right) = 1,\ E\left(v_3\right) = 3$$

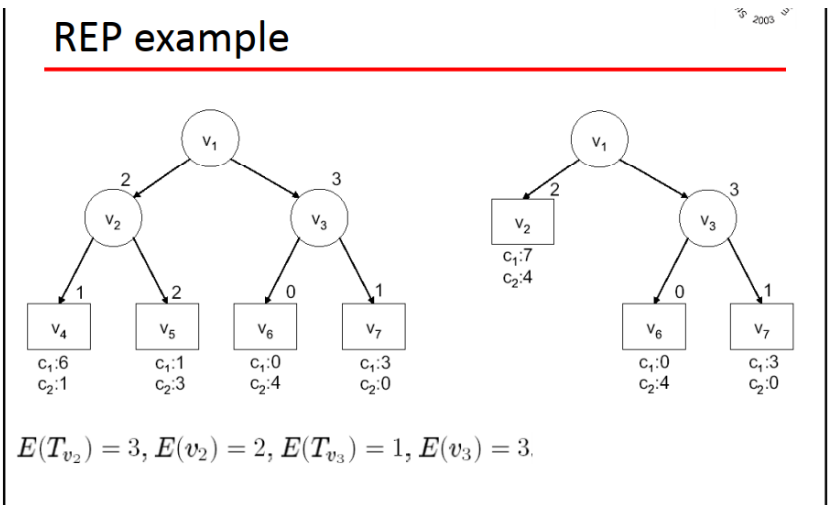

Figure 4: Example of Reduced Error Pruning (REP)