Machine learning

Decision Trees

Hamid Beigy

Sharif University of Technology

November 12, 2021

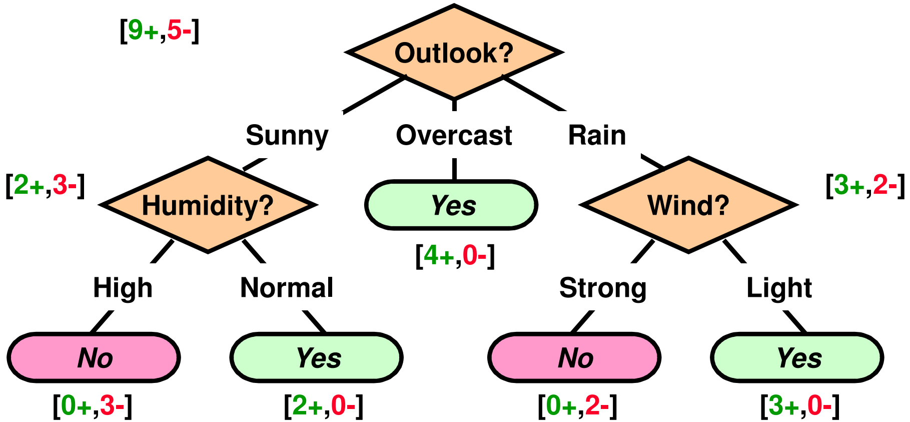
Introduction

Introduction

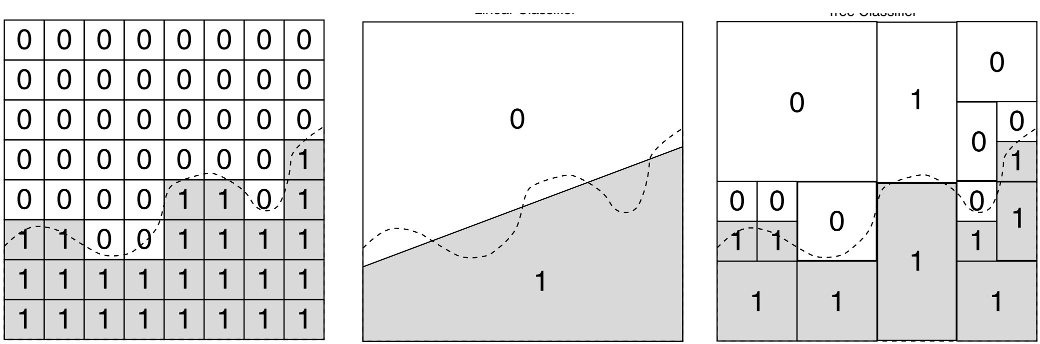
1. The decision tree is a classic and natural model of learning.
2. It is closely related to the notion of divide and conquer.
3. A decision tree partitions the instance space into axis-parallel regions, labeled with class value
4. Why decsion trees?
   - Interpretable, popular in medical applications because they mimic the way a doctor thinks
   - Can model discrete outcomes nicely
   - Can be very powerful, can be as complex as you need them
   - C4.5 and CART decision trees are very popular.

Decision tree classification

1. Structure of decsion trees
   - Each internal node tests an attribute
   - Each branch corresponds to attribute value
   - Each leaf node assigns a classification.
2. Decision Tree for PlayTennis

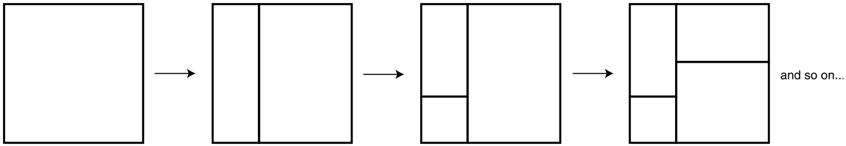


Decision surface

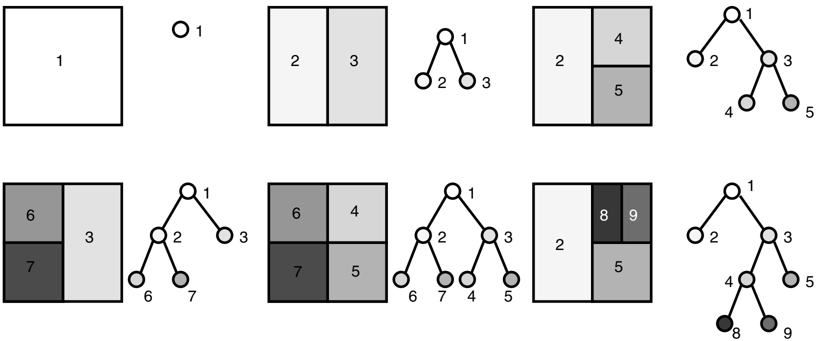


Building decision trees

1. Decsion trees recursively subdivide the feature space.



2. The test variable specifies the division

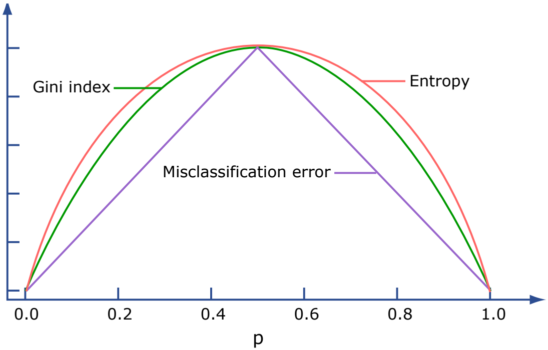


Building decision trees (example)

- Training examples for PlayTennis

Day	Outlook	Temperature	Humidity	Wind	*PlayTennis?*
1	Sunny	Hot	High	Light	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Light	Yes
4	Rain	Mild	High	Light	Yes
5	Rain	Cool	Normal	Light	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Light	No
9	Sunny	Cool	Normal	Light	Yes
10	Rain	Mild	Normal	Light	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Light	Yes
14	Rain	Mild	High	Strong	No

- How to build a decision tree?
  1. Start at the top of the tree.
  2. Grow it by splitting attributes one by one.
  3. Assign leaf nodes.
  4. When we get to the bottom, prune the tree to prevent overfitting.
- How choose a test variable for an internal node?
- Choosing different measures result in different algorithms. We describe ID3.



ID3 Algorithm

- ID3 uses information gain to choose a test variable for an internal node.
- The information gain of S relative to attribute A is the expected reduction in entropy due to splitting on A.

$$Gain(S, A) = H(S) - \sum_{v \in values(A)} \left[\frac{|S_v|}{|S|} H(S_v) \right]$$

where S_v is $\{x \in S : x.A = v\}$, the set of examples in S where attribute A has value v

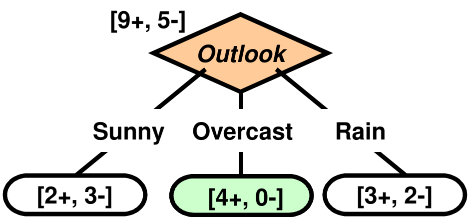
Day	Outlook	Temperature	Humidity	Wind	*PlayTennis?*
1	Sunny	Hot	High	Light	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Light	Yes
4	Rain	Mild	High	Light	Yes
5	Rain	Cool	Normal	Light	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Light	No
9	Sunny	Cool	Normal	Light	Yes
10	Rain	Mild	Normal	Light	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Light	Yes
14	Rain	Mild	High	Strong	No

$$
\begin{aligned}
H(S) &= -(9/14)\log(9/14) - (5/14)\log(5/14) = 0.94 bits \\
H(S, Humidity = High) &= -(3/7)\log(3/7) - (4/7)\log(4/7) = 0.985 bits \\
H(S, Humidity = Normal) &= -(6/7)\log(6/7) - (1/7)\log(1/7) = 0.592 bits \\
Gain(S, Humidity) &= 0.94 - (7/14) * 0.985 - (7/14) * 0.592 = 0.151 bits \\
Gain(S, Wind) &= 0.94 - (8/14) * 0.811 + (6/14) * 1.0 = 0.048 bits
\end{aligned}
$$

ID3 Algorithm

Day	Outlook	Temperature	Humidity	Wind	*PlayTennis*?
1	Sunny	Hot	High	Light	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Light	Yes
4	Rain	Mild	High	Light	Yes
5	Rain	Cool	Normal	Light	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Light	No
9	Sunny	Cool	Normal	Light	Yes
10	Rain	Mild	Normal	Light	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Light	Yes
14	Rain	Mild	High	Strong	No

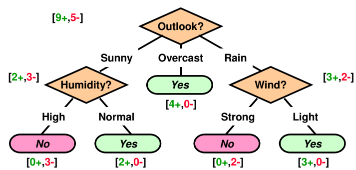
$$Gain(S, Humidity) = 0.151 bits$$
$$Gain(S, Wind) = 0.048 bits$$
$$Gain(S, Temperature) = 0.029 bits$$
$$Gain(S, Outlook) = 0.246 bits$$



ID3 Algorihm

Day	Outlook	Temperature	Humidity	Wind	*PlayTennis*?
1	Sunny	Hot	High	Light	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Light	Yes
4	Rain	Mild	High	Light	Yes
5	Rain	Cool	Normal	Light	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Light	No
9	Sunny	Cool	Normal	Light	Yes
10	Rain	Mild	Normal	Light	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Light	Yes
14	Rain	Mild	High	Strong	No

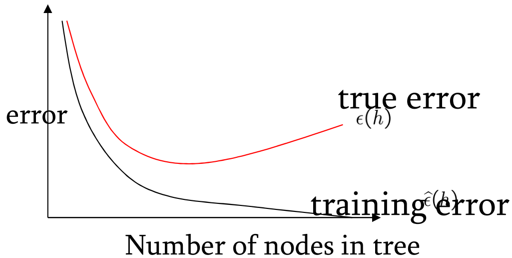
$$
\begin{aligned}
Gain(S_{Sunny}, Humidity) &= 0.97 bits \\
Gain(S_{Sunny}, Wind) &= 0.02 bits \\
Gain(S_{Sunny}, Temperature) &= 0.57 bits
\end{aligned}
$$



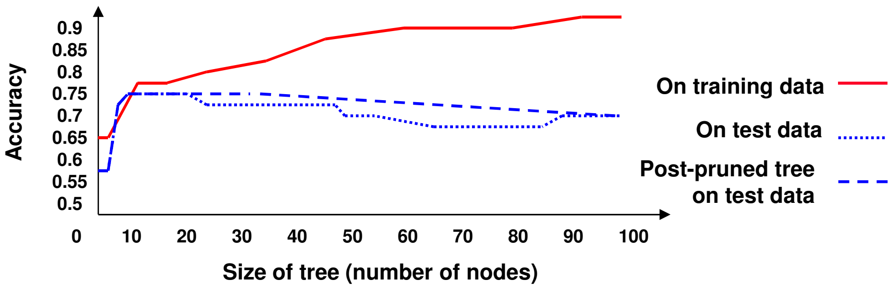
- Types of Biases
  1. Preference (search) bias
     Put priority on choosing hypothesis.
  2. Language bias
     Put restriction on the set of hypotheses considered
- Which Bias is better?
  1. Preference bias is more desirable.
  2. Because, the learner works within a complete space that is assured to contain the unknown concept.
- Inductive Bias of ID3
  1. Shorter trees are preferred over longer trees.
  2. Occam's razor : Prefer the simplest hypothesis that fits the data.
  3. Trees that place high information gain attributes close to the root are preferred over those that do not.

- How can we avoid over-fitting?
  1. Prevention
     - Stop training (growing) before it reaches the point that overfits.
     - Select attributes that are relevant (will be useful in the decision tree)
     - Requires some predictive measure of relevance
  2. Avoidance
     - Allow to over-fit, then improve the generalization capability of the tree.
     - Holding out a validation set (test set)
  3. Detection and Recovery
     - Letting the problem happen, detecting when it does, recovering afterward
     - Build model, remove (prune) elements that contribute to overfitting
- How to select Best tree?
  1. Training and validation set
     Use a separate set of examples (distinct from the training set) for test.
  2. Statistical test
     Use all data for training, but apply the statistical test to estimate the over-fitting.
  3. Define the measure of complexity
     Halting the grow when this measure is minimized.

- Reduced-Error Pruning.



- The effect of reduced error pruning on error



- Variant of this method called Rule Post-Pruning used in C4.5, an outgrowth of ID3

- Prunning algorithms (Esposito, Malerba, and Semeraro 1997).
  1. Reduced Error Pruning
  2. Pessimistic Error Pruning
  3. Minimum Error Pruning
  4. Critical Value Pruning
  5. Cost-Complexity Pruning

- Two methods for handling continuous attributes
  1. Discretization (e.g., histogramming)
     Break real-valued attributes into ranges in advance

Example

$$\begin{aligned} high &= \{Temp > 35C\} \\ med &= \{10C < Temp \leq 35C\} \\ low &= \{Temp \leq 10C\} \end{aligned}$$

  2. Using thresholds for splitting nodes

Example

$A \leq a$ produces subsets $A \leq a$ and $A > a$.

  3. Information gain is calculated the same way as for discrete splits
- How to find the split with highest Gain

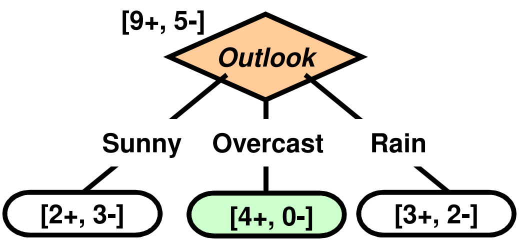
Example

length	10		15	21		28		32	40		50
label	-		+	+		-		+	+		-
Thresholds		12.5			24.5		30			45	

- Problem: What If Some Examples Missing Values of A?
- Consider dataset.

Day	Outlook	Temperature	Humidity	Wind	*PlayTennis*?
1	Sunny	Hot	High	Light	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Light	Yes
4	Rain	Mild	High	Light	Yes
5	Rain	Cool	Normal	Light	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	???	Light	No
9	Sunny	Cool	Normal	Light	Yes
10	Rain	Mild	Normal	Light	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Light	Yes
14	Rain	Mild	High	Strong	No

- What is the decision tree?



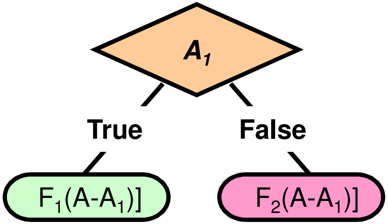
- Solutions: Incorporatinga Guess into Calculation of $Gain(S, A)$.

Attributes with Many Values

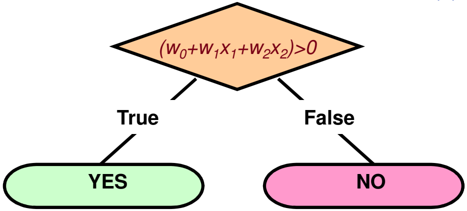
- Problem: If attribute has many values such as Date, *Gain*(.) will select it (why?)
- One Approach: Use *GainRatio* instead of *Gain*
- *SplitInformation*: directly proportional to $|values(A)|$, i.e., penalizes attributes with more values.
- What is its inductive bias?
- Preference bias (for lower branch factor) expressed via *GainRatio*(.)
- Alternative attribute selection : Gini Index

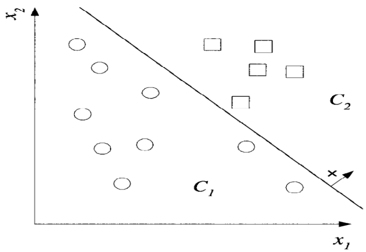
- Problem: In some learning tasks the instance attributes may have associated costs.
- Solutions
  1. ExtendedID3
     $$\frac{Gain(S, A)}{Cost(A)}$$
  2. TanandSchlimmer
     $$\frac{Gain^2(S, A)}{Cost(A)}$$
  3. Nunez
     $$\frac{2^{Gain(S,A)} - 1}{(Cost(A) + 1)^w}$$
     where $w \in [0, 1]$ is a constant.

- In regression tree, the goodness of a split is measured by the mean square error from the estimated value (Breiman et al. 1984; Malerba et al. 2004).

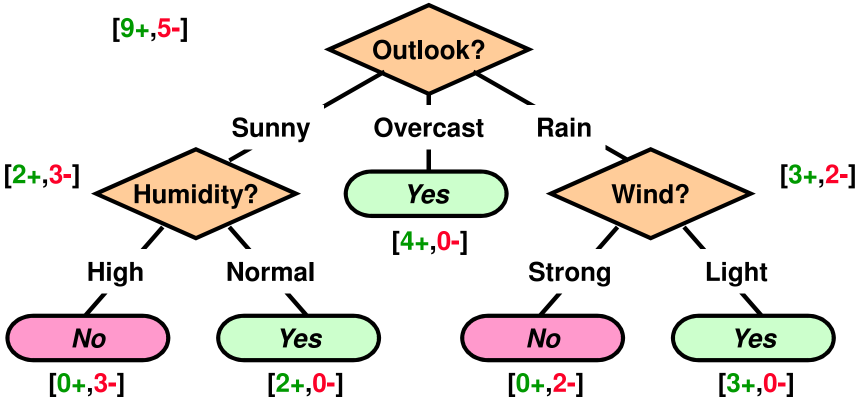


- Univariate trees
  In univariate trees, the test at each internal node just uses only one of input attributes.
- Multivariate trees
  In multivariate trees, the test at each internal node can use all input attributes (Brodley and Utgoff 1995).





- ID3 can not be trained incrementally.
- ID4, ID5, ID5R are samples of incremental induction of decision trees (Utgoff 1989).



1. Chapter 3 of Machine Learning Book (Mitchell 1997).
2. Papers (Esposito, Malerba, and Semeraro 1997; Murthy 1998)

- Breiman, Leo et al. (1984). *Classification and Regression Trees*. Wadsworth.
- Brodley, Carla E. and Paul E. Utgoff (1995). "Multivariate Decision Trees". In: *Machine Learning* 19.1, pp. 45–77.
- Esposito, Floriana, Donato Malerba, and Giovanni Semeraro (1997). "A Comparative Analysis of Methods for Pruning Decision Trees". In: *IEEE Transactions on Pattern Analysis Machine Intelligence* 19.5, pp. 476–491.
- Malerba, Donato et al. (2004). "Top-Down Induction of Model Trees with Regression and Splitting Nodes". In: *IEEE Transactions on Pattern Analysis Machine Intelligence* 26.5, pp. 612–625.
- Mitchell, Tom M. (1997). *Machine Learning*. McGraw-Hill.
- Murthy, Sreerama K. (1998). "Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey". In: *Data Min. Knowl. Discov.* 2.4, pp. 345–389.
- Utgoff, Paul E. (1989). "Incremental Induction of Decision Trees". In: *Machine Learning* 4, pp. 161–186.

Questions?