Tutorial Business Analytics

Homework 6 - Solution

Homework 6.1

Calculate the optimal splits for the following attributes using Info Gain. Therefore, find every possible split-point and determine its goodness.

a)

35	35	37	40	40	40
F	F	T	F	T	T

b)

36.8	36.8	37.2	38.3	38.3	39.7
T	F	F	T	F	F

Solution 6.1

a) Possible split points:

   a. 36: info([0,2], [3,1]) = 0.541
   b. 38.5: info([1,2], [2,1]) = 0.918

   → decide to split at 36.

b) Possible split points:

   a. 37: info([1,1], [1,3]) = 0.874
   b. 37.75: info([1,2], [1,2]) = 0.918
   c. 39: info([2,3], [0,1]) = 0.809

   → decide to split at 39.

Homework 6.2

You want to predict match results of soccer matches. In order to improve your forecasts, you decide to use your knowledge on data mining and construct a decision tree from the following table.

Host	Better form	Referee's preference	Tradition	Result
A	B	B	4	X
A	A	None	4	A
B	A	B	1	B
B	A	None	3	X
A	B	None	1	B
A	B	None	2	X
B	A	B	2	B
B	Same	None	1	B
A	Same	None	5	A
A	B	None	5	A
B	Same	None	4	A
B	Same	A	3	A
A	Same	A	3	A
A	B	None	3	A

Construct the first two levels of the decision tree using Gain Ratio.

Note: Tradition is a numerical attribute; you need to split it using a binary split. In order to construct the root use 2.5 as value for the split point (breakpoint). If necessary, find the optimal split point for the second level. The attribute Tradition indicates how many games out of the last six team A won.

Note: A and B are teams. The value Same indicates that both teams are in equal form. X means that the game was a draw.

Solution 6.2

Notation: [A, X, B]
Whole data record: info([7, 3, 4]) = 1.493

Level 1:

Host:

- Gain(Host) = Info([7, 3, 4]) – Info([5, 2, 1], [2, 1, 3]) = 1.493 – 1.368 = 0.125
- IntrinsicInfo(Host) = Info([8, 6]) = 0.985
- GainRatio(Host) = Gain(Host)/IntrinsicInfo(Host) = 0.125/0.985 = 0.127

Better form:

- Gain(Better Form) = Info([7, 3, 4]) – Info([1, 1, 2], [4 ,0, 1], [2, 2, 1])
  = 1.493 – 1.230 = 0.263
- IntrinsicInfo(Better Form) = Info([4, 5, 5]) = 1.577
- GainRatio(Better Form) = 0.263/1.577 = 0.167

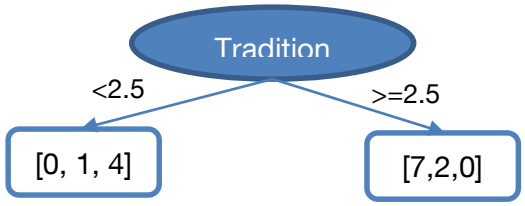
Referee's preference:

- Gain(Ref. pref.) = Info([7, 3, 4]) – Info([2, 0, 0], [5, 2, 2], [0, 1, 2])
  = 1.493 – 1.120 = 0.373
- IntrinsicInfo(Ref. pref.) = Info([2, 9, 3]) = 1.287
- GainRatio(Ref. pref.) = 0.373/1.287 = 0.290

Tradition:

- Gain(Tradition, split 2.5) = Info([7, 3, 4]) – Info([0, 1, 4], [7, 2, 0])
  = 1.493 – 0.749 = 0.744
- IntrinsicInfo(Tradition, split 2.5) = Info([5, 9]) = 0.940
- GainRatio(Tradition, split 2.5) = 0.744 / 0.940 = 0.791

We choose Tradition with split at 2.5 as the first node!



Level 2 – Left sub tree (Tradition < 2.5):

Host	Better form	Referee's preference	Tradition	Result
B	A	B	1	B
A	B	None	1	B
A	B	None	2	X
B	A	B	2	B
B	Same	None	1	B

info(root) = info([0, 1, 4]) = 0.722

Host:

- Gain(Host) = Info([0, 1, 4]) – Info([0, 1, 1], [0, 0, 3]) = 0.722 – 0.4 = 0.322
- IntrinsicInfo(Host) = Info([2,3]) = 0.971
- GainRatio(Host) = 0.322/0.971 = 0.332

Better form:

- Gain(Better Form) = Info([0, 1, 4]) – Info([0, 0, 2], [0, 0, 1],[0, 1, 1])
  = 0.722 – 0.4 = 0.322
- IntrinsicInfo(Better form) = Info([2, 1, 2]) = 1.522
- GainRatio(Better form) = 0.322/1.522 = 0.212

Referee's preference:

- Gain(Ref. pref.) = Info([0, 1, 4]) – Info([0, 1, 2], [0, 0, 2]) = 0.722 – 0.551 = 0.171
- IntrinsicInfo(Ref. pref.) = Info([3, 2]) = 0.971
- GainRatio(Ref. pref.) = 0.171/0.971 = 0.176

Tradition, split at 1.5

- Gain(Tradition) = Info([0, 1, 4]) – Info([0, 0, 3], [0, 1, 1]) = 0.722 – 0.4 = 0.322
- IntrinsicInfo(Tradition) = Info([3, 2]) = 0.971
- GainRatio(Tradition) = 0.322/0.971 = 0.332

We choose attribute Host or Tradition with split at 1.5.

Level 2 – Right sub tree (Tradition >= 2.5):

Host	Better form	Referee's preference	Tradition	Result
A	B	B	4	X
A	A	None	4	A
B	A	None	3	X
A	Same	None	5	A
A	B	None	5	A
B	Same	None	4	A
B	Same	A	3	A
A	Same	A	3	A
A	B	None	3	A

Info(root) = Info([7, 2, 0]) = 0.764

Host:

- Gain(Host) = Info([7, 2, 0]) – Info([5, 1, 0], [2, 1, 0]) = 0.764 – 0.739 = 0.025
- IntrinsicInfo(Host) = Info([6, 3]) = 0.918
- GainRatio(Host) = 0.025/0.918 = 0.027

Better form:

- Gain(Better form) = Info([7, 2, 0]) – Info([1, 1, 0], [4, 0, 0], [2, 1, 0])
  = 0.764 – 0.528 = 0.236
- IntrinsicInfo(Better form) = Info([2, 4, 3]) = 1.530
- GainRatio(Better form) = 0.236/1.530 = 0.154

Referee's preference:

- Gain(Ref. pref.) = Info([7, 2, 0]) – Info([2, 0, 0], [5, 1, 0], [0, 1, 0])
  = 0.764 – 0.433 = 0.331
- IntrinsicInfo(Ref. pref.) = Info([2, 6, 1]) = 1.224
- GainRatio(Ref. pref.) = 0.331/1.224 = 0.270

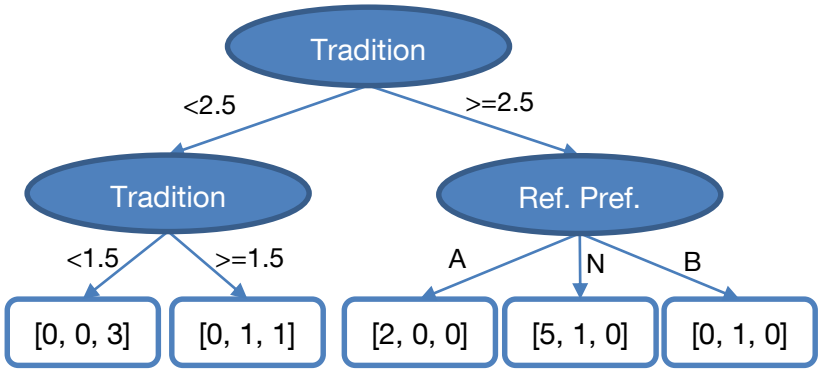
Tradition, split at 3.5:

- Gain(Tradition) = Info([7, 2, 0]) – Info([3, 1, 0], [4, 1, 0]) = 0.764 – 0.762 = 0.002
- IntrinsicInfo(Tradition) = Info([4, 5]) = 0.991
- GainRatio(Tradition) = 0.002/0.991 = 0.002

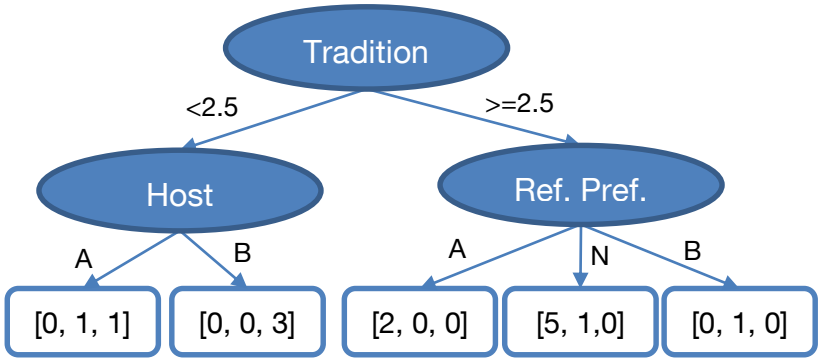
Tradition, split at 4.5:

- Gain(Tradition) = Info([7, 2, 0]) – Info([5, 2, 0], [2, 0, 0]) = 0.764 – 0.671 = 0.093
- IntrinsicInfo(Tradition) = Info([7, 2]) = 0.764
- GainRatio(Tradition) = 0.093/0.764 = 0.122

We choose the attribute Referee's preference:



or



Homework 6.3

The problem of overfitting with of too complex trees might arise when constructing a decision tree. Solution: **Tree pruning**. Its purpose is shortening or simplifying the tree. Types of pruning are:

- Prepruning (during the construction of the tree): Abort the construction before the tree is too complex. Hard decision to make because of the number of possible attribute combinations
- Postpruning (after the tree has been constructed): Construct the whole tree and then prune it. Waste of computing time

Replacing a subtree with a leaf node may decrease the accuracy on the training set and increase accuracy on the test set.

Criterion for replacement:

- The error rate of a node and the error rate of its leaf nodes is *estimated*
- If the *estimated error rate* of the node is smaller than the estimated error rate of its leaf nodes → Replace the node with one leaf node

Candidates:

- Estimate error based on the training set: Bad choice, because tree adapts to the training set
- Withhold part of the training set and use it as test set (reduced-error pruning): Less data to construct the tree
- **Method of C4.5:**
  - Pessimistic *estimation* of error rate *e*
  - Based on *observed* error rate *f = E/N* with *E* errors and *N* instances
  - Confidence level: *c* (e.g., 25%) → $1 - c = \Phi(z)$ (z = z_{1-c}), *z* is called confidence limit
  - Formula (<u>these values would be provided during the exam!</u>):

$$e = \frac{f + \frac{z^2}{2N} + z\sqrt{\frac{f}{N} - \frac{f^2}{N} + \frac{z^2}{4N^2}}}{1 + \frac{z^2}{N}}$$

Decide if the subtree starting at the leafs a) and b) should be pruned. Follow the C4.5 algorithm with a confidence factor c = 0.25 (z = 0.69).

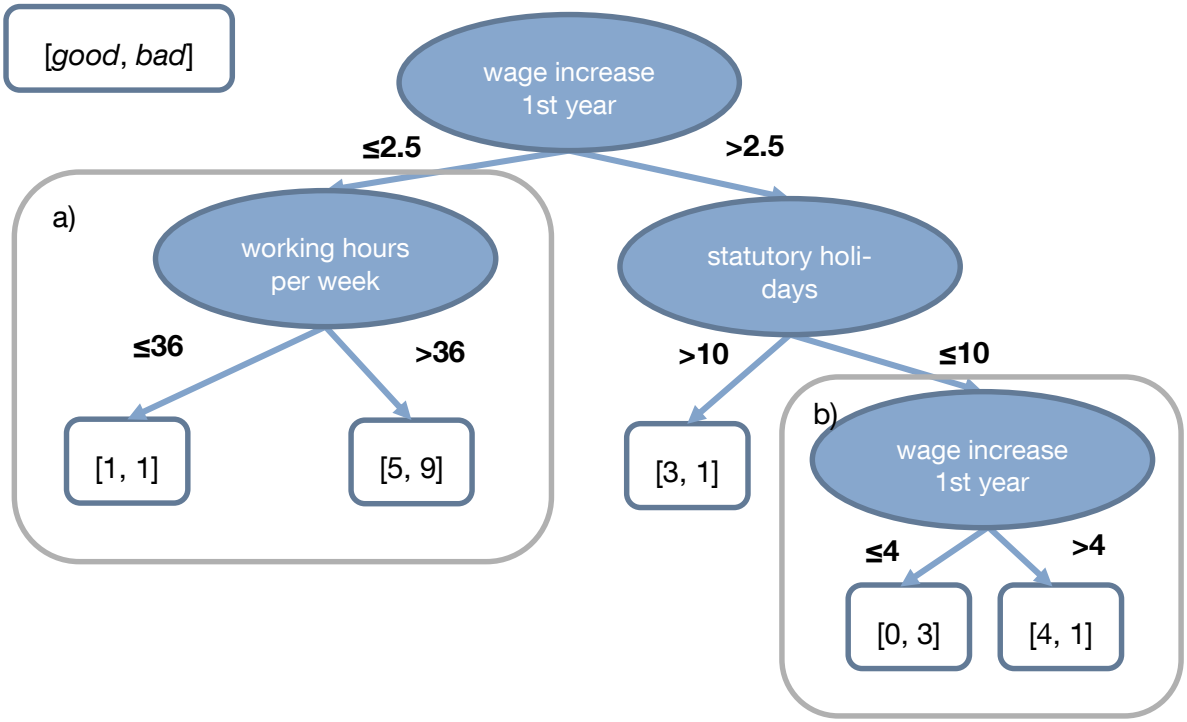
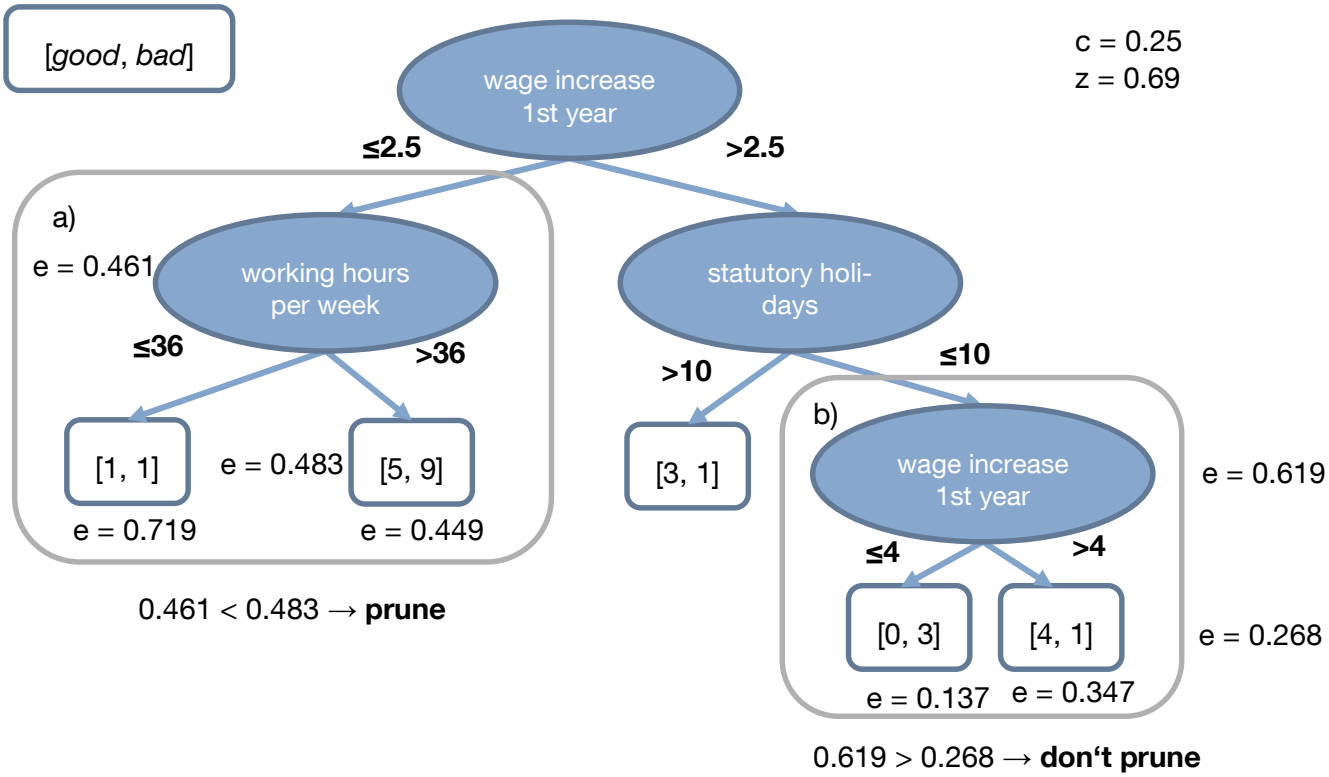


Table of estimated error rates e:

	E=0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
N=1	0.323	1.000															
2	0.192	0.719	1.000														
3	0.137	0.532	0.820	1.000													
4	0.106	0.420	0.663	0.867	1.000												
5	0.087	0.347	0.553	0.736	0.895	1.000											
6	0.074	0.295	0.474	0.636	0.783	0.913	1.000										
7	0.064	0.257	0.414	0.558	0.692	0.815	0.926	1.000									
8	0.056	0.227	0.368	0.497	0.619	0.733	0.840	0.935	1.000								
9	0.050	0.204	0.330	0.448	0.559	0.664	0.764	0.858	0.942	1.000							
10	0.045	0.185	0.300	0.407	0.509	0.607	0.700	0.789	0.873	0.948	1.000						
11	0.041	0.169	0.275	0.373	0.467	0.558	0.645	0.729	0.809	0.885	0.953	1.000					
12	0.038	0.156	0.253	0.345	0.432	0.516	0.598	0.677	0.753	0.826	0.895	0.957	1.000				
13	0.035	0.144	0.235	0.320	0.402	0.480	0.557	0.631	0.703	0.773	0.839	0.903	0.960	1.000			
14	0.033	0.134	0.219	0.299	0.375	0.449	0.521	0.591	0.659	0.725	0.789	0.851	0.910	0.963	1.000		
15	0.031	0.126	0.205	0.280	0.352	0.421	0.489	0.555	0.620	0.683	0.744	0.804	0.862	0.916	0.966	1.000	
16	0.029	0.118	0.193	0.263	0.331	0.397	0.461	0.524	0.585	0.645	0.704	0.761	0.817	0.870	0.921	0.968	1.000

Solution 6.3



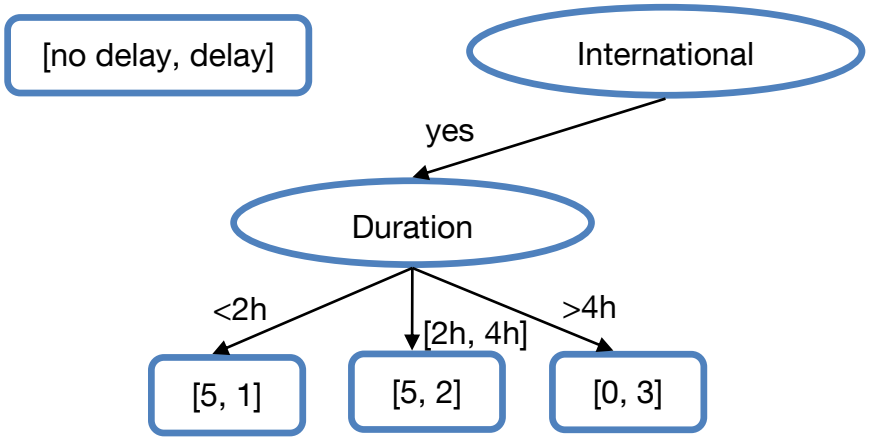
0.461 < 0.483 → **prune**

0.619 > 0.268 → **don‘t prune**

Homework 6.4

Below-mentioned part of a decision tree illustrates the prediction whether flights will be delayed or not.

Use subtree replacement in order to decide if the subtree starting at the leaf “Duration” should be pruned. Follow the C4.5 algorithm with a confidence factor c = 0.25.



Solution 6.4

c = 0.25, z = 0.69.

Leaf node 1 (<2h):

$$N = 6, \quad E = 1, \quad f = \frac{1}{6}$$
$$e = 0.295$$

Leaf node 2 ([2h, 4h]):

$$N = 7, \quad E = 2, \quad f = \frac{2}{7}$$
$$e = 0.414$$

Leaf node 3 (>4h):

$$N = 3, \quad E = 0, \quad f = 0$$
$$e = 0.137$$

Average estimated error rate:

$$e = \frac{6}{16} \cdot 0.295 + \frac{7}{16} \cdot 0.414 + \frac{3}{16} \cdot 0.137 = 0.317$$

Estimated error rate of the leaf node ([10, 6]):

$$N = 16, \quad E = 6, \quad f = \frac{6}{16}$$
$$e = 0.461$$

Result: The leaf nodes' error rate is smaller than the node's error rate, therefore we do **not** prune the tree (0.317 < 0.461).