Decision trees - demonstrations

Alexandre Devert
Software Engineering School of the USTC

December 7, 2012

1 *ID3* scoring for a split

The *ID3* algorithm recursively split a dataset into subsets, until the subsets are contains only one class/label/identifier. At the core of the algorithm, there is the *gain function*, a function that give a score to split. Among a set of splits, *ID3* will select the split that reduce class entropy the most.

The *ID3* gain function is based on entropy : it selects the split that reduce as much as possible the entropy of the resulting two subsets.

2 Gain function on 2d data

Let's consider the following data, 2d points.

There are 9 points, and here we will consider decisions on the x and y variables : $x < \alpha$ and $y < \alpha$. 9 points, two possible decisions per points, it gives 18 possible splits. In this example, we will consider 2 splits, $y < 4$ and $x < 5$. We are going to compute the entropy gain with the formula of the *ID3* algorithm.

2.1 Class entropy of the dataset

Let's consider the complete dataset S. We have 9 points, 4 with the $True$ class and 5 with the $False$ class. The class entropy $E(S)$ of the dataset is

$$E(S) = -\frac{4}{9}\log_2\left(\frac{4}{9}\right) - \frac{5}{9}\log_2\left(\frac{5}{9}\right) \simeq 0.99$$

x	y	class
6	1	True
4	2	False
8	3	True
5	3	False
3	4	True
2	4	False
7	5	True
3	6	False
1	7	False

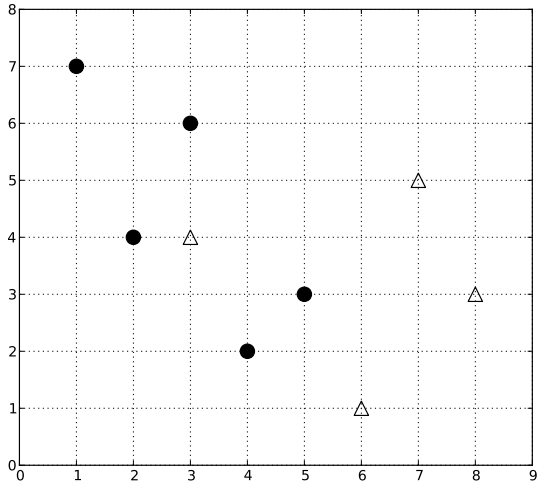

Figure 1: The dataset S

2.2 Entropy gain of the split $y < 4$

This split cut the dataset in two, giving the following subsets A and B

x	y	class
6	1	True
4	2	False
8	3	True
5	3	False

(a) Subset A ($y < 4$)

x	y	class
3	4	True
2	4	False
7	5	True
3	6	False
1	7	False

(b) Subset B ($y \geq 4$)

Subset A have 4 points, 2 with the $True$ class and 2 with the $False$ class. The class entropy $E(A)$ of the subset A.

$$E(A) = -\frac{2}{4}\log_2\left(\frac{2}{4}\right) - \frac{2}{4}\log_2\left(\frac{2}{4}\right) = 1$$

Subset B have 5 points, 2 with the $True$ class and 3 with the $False$ class. The class entropy $E(B)$ of the subset B.

$$E(B) = -\frac{2}{5}\log_2\left(\frac{2}{5}\right) - \frac{3}{5}\log_2\left(\frac{3}{5}\right) \simeq 0.97$$

We can now compute the entropy gain G of the $y < 4$ split.

$$G = 0.99 - \left(\frac{4}{9} \times 1 + \frac{5}{9} \times 0.97\right) \simeq 0.006$$

The gain is very low. As we can see, both subset have about as much as $True$ and $False$ class, so this split does not really help to classify the points.

2.3 Entropy gain of the split $x < 5$

This split cut the dataset in two, giving the following subsets A and B

x	y	class
4	2	False
3	4	True
2	4	False
3	6	False
1	7	False

(c) Subset A ($x < 5$)

x	y	class
6	1	True
8	3	True
5	3	False
7	5	True

(d) Subset B ($x \geq 5$)

Subset A have 5 points, 1 with the $True$ class and 4 with the $False$ class. The class entropy $E(A)$ of the subset A.

$$E(A) = -\frac{1}{5}\log_2\left(\frac{1}{5}\right) - \frac{4}{5}\log_2\left(\frac{4}{5}\right) \simeq 0.72$$

Subset B have 4 points, 3 with the $True$ class and 1 with the $False$ class. The class entropy $E(B)$ of the subset B.

$$E(B) = -\frac{3}{4}\log_2\left(\frac{3}{4}\right) - \frac{1}{4}\log_2\left(\frac{1}{4}\right) \simeq 0.81$$

We can now compute the entropy gain G of the $x < 5$ split.

$$G = 0.99 - \left(\frac{5}{9} \times 0.72 + \frac{4}{9} \times 0.81\right) \simeq 0.23$$

The gain is for the split $x < 5$ is higher than the $y < 4$ split, thus, the split $x < 5$ is a better choice. As we can see, the $x < 5$ split creates two subsets where most of the points have the same class. A proper implementation of $ID3$ would of course compute the gain of *all* the possible splits, to pick the best one.

3 Building a tree with *ID3*

Roger Federer is one of the greatest tennis player since tennis have been invented. We want to learn a little about what makes R. Federer win or loose a match. We would like to be able to predict the outcome of a match. Knowing the conditions in which a tennis match takes place, we would like to predict whether R. Federer will win or loose the match. To do so, we gathered data from games played by R. Federer, shown in Table 1.

Time	**Match type**	**Court surface**	**Best effort**	**Outcome**
Morning	Master	Grass	Yes	*Win*
Afternoon	Grand Slam	Clay	Yes	*Win*
Night	Friendly	Hard	No	*Win*
Afternoon	Friendly	Mixed	No	*Loose*
Afternoon	Master	Clay	Yes	*Loose*
Afternoon	Grand Slam	Grass	Yes	*Win*
Afternoon	Grand Slam	Hard	Yes	*Win*
Night	Master	Mixed	Yes	*Loose*
Afternoon	Master	Grass	Yes	*Win*
Afternoon	Grand Slam	Hard	Yes	*Win*
Afternoon	Master	Clay	Yes	*Loose*
Afternoon	Grand Slam	Hard	Yes	*Win*
Morning	Master	Grass	Yes	*Win*
Afternoon	Grand Slam	Grass	Yes	*Loose*
Night	Friendly	Hard	No	*Win*
Afternoon	Grand Slam	Clay	Yes	*Win*

Table 1: R. Federer games

1. We compute the entropy for the whole dataset. 16 entries, 11 *win* and 5 *loose*, it gives an entropy of about 0.99

2. We split the dataset for each attribute, and compute the entropy of each splits. A table shows the details of the calculations

Attribute value	Win	Loose	Entropy
Time			
Morning	2	0	0
Afternoon	7	4	0.94
Night	2	1	0.91
Match Type			
Master	3	3	1
Grand Slam	6	1	0.59
Friendly	2	1	0.91
Court Surface			
Grass	4	1	0.72
Clay	2	2	1
Hard	5	0	0
Mixed	0	2	0
Best effort			
Yes	9	4	0.89
No	2	1	0.91

3. We compute the entropy gain for each split The best entropy gain is

Attribute	Entropy gain
Time	$0.99 - \frac{1}{16}(2 \times 0 + 13 \times 0.94 + 3 \times 0.91) = 0.17$
Match Type	$0.99 - \frac{1}{16}(6 \times 1 + 7 \times 0.59 + 3 \times 0.91) = 0.18$
Court Surface	$0.99 - \frac{1}{16}(5 \times 0.72 + 4 \times 1) = 0.51$
Best Effort	$0.99 - \frac{1}{16}(13 \times 0.89 + 3 \times 0.91) = 0.09$

obtained if we use the attribute *court surface*. The tree, at this point, will look like this

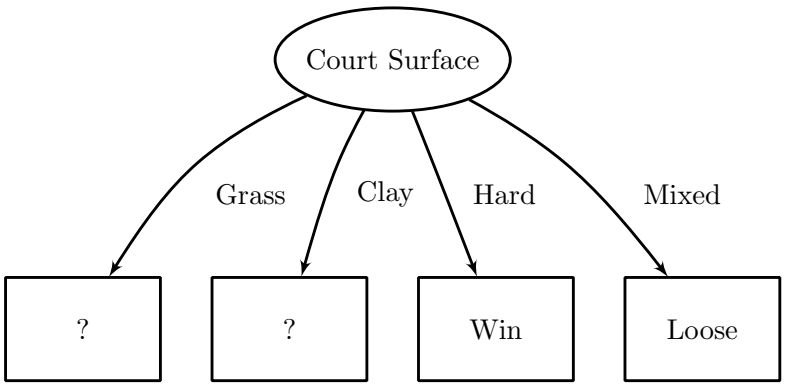

Figure 2: Decision tree, after splitting with the *court surface* attribute

4. We now consider the subset of data for which the *court surface* attribute is *clay*. Without even computing the entropy gain for each attribute,

Time	**Match type**	**Best effort**	**Outcome**
Afternoon	Grand Slam	Yes	*Win*
Afternoon	Master	Yes	*Loose*
Afternoon	Master	Yes	*Loose*
Afternoon	Grand Slam	Yes	*Win*

we can see that *time* and *best effort* are the same for all the subset. So they can't be used to explain the *outcome* attribute, giving an entropy gain of 0. But the *match type* is either *grand slam*, outcome is then always *win*, or *master*, outcome is then always *loose*. The entropy gain for the *match type* attribute is 0.99. Thus, we split this subset with the attribute *match type*. The tree, at this point, will look like this

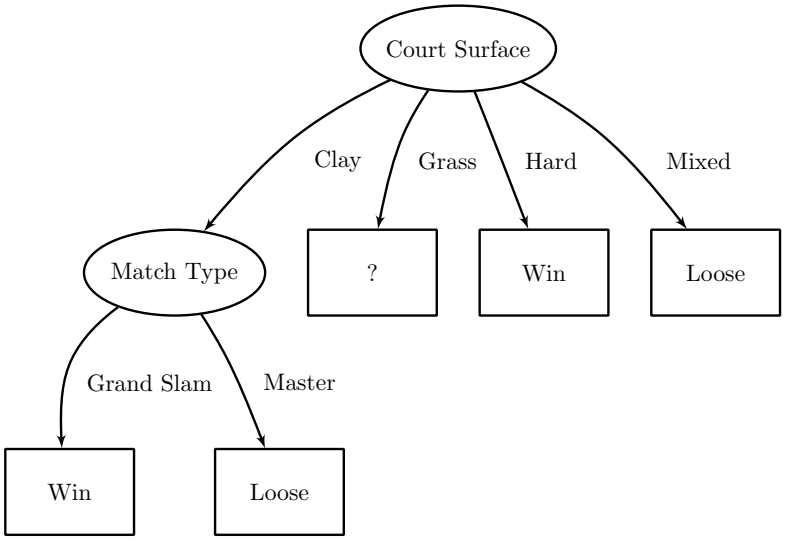

Figure 3: Decision tree, after splitting the subset *court surface = clay* with the *match type* attribute

5. We now consider the subset of data for which the *court surface* attribute is *grass*. The entropy of the subset is 0.72. We split the subset for each

Time	**Match type**	**Best effort**	**Outcome**
Morning	Master	Yes	*Win*
Afternoon	Grand Slam	Yes	*Win*
Afternoon	Master	Yes	*Win*
Morning	Master	Yes	*Win*
Afternoon	Grand Slam	Yes	*Loose*

attribute, and compute the entropy of each splits. A table shows the details of the calculations. We don't need to consider *Best effort*, as it is always equal to *yes*.

Attribute value	Win	Loose	Entropy
Time			
Morning	2	0	0
Afternoon	2	1	0.91
Match Type			
Master	3	0	0
Grand Slam	1	1	1

6. We compute the entropy gain for each split The best entropy gain is

Attribute	Entropy gain
Time	$0.72 - \frac{1}{5}(2 \times 0 + 3 \times 0.91) = 0.17$
Match Type	$0.72 - \frac{1}{5}(3 \times 0 + 2 \times 1) = 0.32$

obtained if we use the attribute *court surface*. The tree, at this point, will look like this

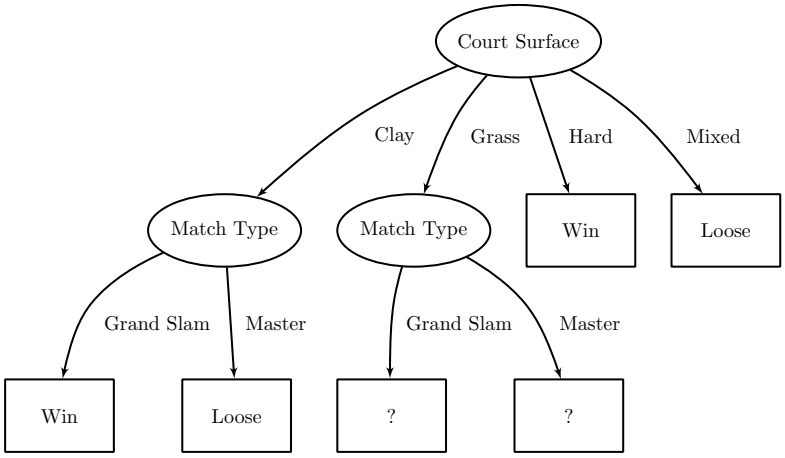

Figure 4: Decision tree, after splitting with the *match type* attribute

7. For *court surface = grass* and *match type = master*, the entropy of the subset is 0, we do not need to subdivide it.

Time	Best effort	Outcome
Morning	Yes	*Win*
Afternoon	Yes	*Win*
Morning	Yes	*Win*

8. For *court surface = grass* and *match type = grand slam*, none of the attribute can help to tell about the outcome, we do not need to subdivide it.

Time	Best effort	Outcome
Afternoon	Yes	*Win*
Afternoon	Yes	*Loose*

The final tree is

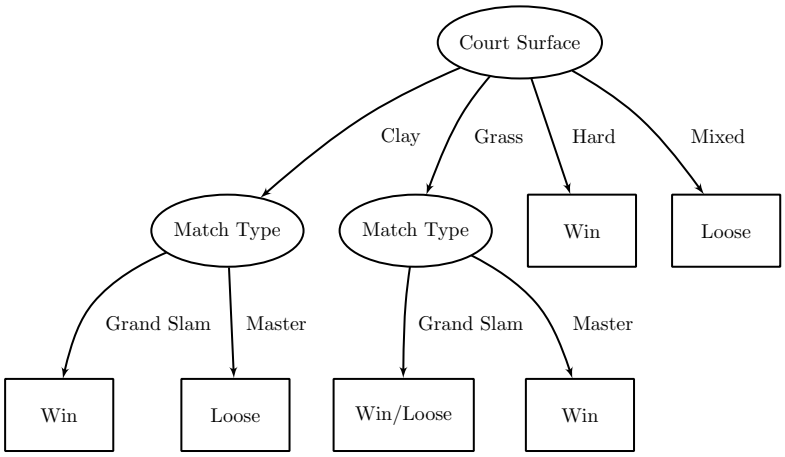

Figure 5: The final decision tree