Kolmogorov complexity and Inductive inference

Kolmogorov complexity of an object, such as a piece of text, is the length of a shortest computer program (in a predetermined programming language) that produces the object as output. It is a measure of the computational resources needed to specify the object.

MDL and learning decision tree

> **Minimum description length (MDL) principle.** Given a sample of data and an effective enumeration of the appropriate alternative theories to explain the data, the best theory is the one that minimizes the sum of
>
> - the length, in bits, of the description of the theory;
> - the length, in bits, of the data when encoded with the help of the theory.

code the decision tree and exceptions to reach the minimum message length

Example

No.	Attributes				Class
	Outlook	*Temperature*	*Humidity*	*Windy*	
1	overcast	hot	high	not	N
2	overcast	hot	high	very	N
3	overcast	hot	high	medium	N
4	sunny	hot	high	not	P
5	sunny	hot	high	medium	P
6	rain	mild	high	not	N
7	rain	mild	high	medium	N
8	rain	hot	normal	not	P
9	rain	cool	normal	medium	N
10	rain	hot	normal	very	N
11	sunny	cool	normal	very	P
12	sunny	cool	normal	medium	P
13	overcast	mild	high	not	N
14	overcast	mild	high	medium	N
15	overcast	cool	normal	not	P
16	overcast	cool	normal	medium	P
17	rain	mild	normal	not	N
18	rain	mild	normal	medium	N
19	overcast	mild	normal	medium	P
20	overcast	mild	normal	very	P
21	sunny	mild	high	very	P
22	sunny	mild	high	medium	P
23	sunny	hot	normal	not	P
24	rain	mild	high	very	N

Example

No.	Attributes				Class
	Outlook	*Temperature*	*Humidity*	*Windy*	
1	overcast	hot	high	not	N
2	overcast	hot	high	very	N
3	overcast	hot	high	medium	N
4	sunny	hot	high	not	P
5	sunny	hot	high	medium	P
6	rain	mild	high	not	N
7	rain	mild	high	medium	N
8	rain	hot	normal	not	P
9	rain	cool	normal	medium	N
10	rain	hot	normal	very	N
11	sunny	cool	normal	very	P
12	sunny	cool	normal	medium	P
13	overcast	mild	high	not	N
14	overcast	mild	high	medium	N
15	overcast	cool	normal	not	P
16	overcast	cool	normal	medium	P
17	rain	mild	normal	not	N
18	rain	mild	normal	medium	N
19	overcast	mild	normal	medium	P
20	overcast	mild	normal	very	P
21	sunny	mild	high	very	P
22	sunny	mild	high	medium	P
23	sunny	hot	normal	not	P
24	rain	mild	high	very	N

Alice and Bob both know the four parameters (outlook, temperature, humidity, windy) and their attributes. Alice wishes to send Bob the information in the table, using as few bits as possible. Alice and Bob have to agree in advance on an encoding technique to be used.

Example

No.	Attributes				Class
	Outlook	*Temperature*	*Humidity*	*Windy*	
1	overcast	hot	high	not	N
2	overcast	hot	high	very	N
3	overcast	hot	high	medium	N
4	sunny	hot	high	not	P
5	sunny	hot	high	medium	P
6	rain	mild	high	not	N
7	rain	mild	high	medium	N
8	rain	hot	normal	not	P
9	rain	cool	normal	medium	N
10	rain	hot	normal	very	N
11	sunny	cool	normal	very	P
12	sunny	cool	normal	medium	P
13	overcast	mild	high	not	N
14	overcast	mild	high	medium	N
15	overcast	cool	normal	not	P
16	overcast	cool	normal	medium	P
17	rain	mild	normal	not	N
18	rain	mild	normal	medium	N
19	overcast	mild	normal	medium	P
20	overcast	mild	normal	very	P
21	sunny	mild	high	very	P
22	sunny	mild	high	medium	P
23	sunny	hot	normal	not	P
24	rain	mild	high	very	N

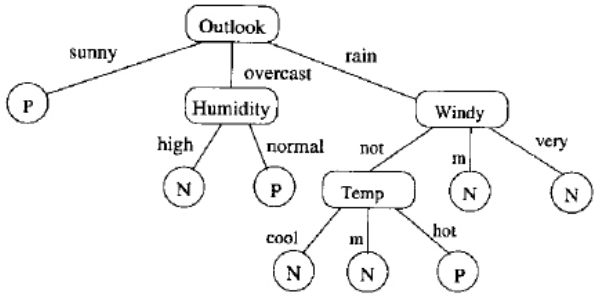


FIGURE 5.5. Perfect decision tree

Example

No.	Attributes				Class
	Outlook	*Temperature*	*Humidity*	*Windy*	
1	overcast	hot	high	not	N
2	overcast	hot	high	very	N
3	overcast	hot	high	medium	N
4	sunny	hot	high	not	P
5	sunny	hot	high	medium	P
6	rain	mild	high	not	N
7	rain	mild	high	medium	N
8	rain	hot	normal	not	P
9	rain	cool	normal	medium	N
10	rain	hot	normal	very	N
11	sunny	cool	normal	very	P
12	sunny	cool	normal	medium	P
13	overcast	mild	high	not	N
14	overcast	mild	high	medium	N
15	overcast	cool	normal	not	P
16	overcast	cool	normal	medium	P
17	rain	mild	normal	not	N
18	rain	mild	normal	medium	N
19	overcast	mild	normal	medium	P
20	overcast	mild	normal	very	P
21	sunny	mild	high	very	P
22	sunny	mild	high	medium	P
23	sunny	hot	normal	not	P
24	rain	mild	high	very	N

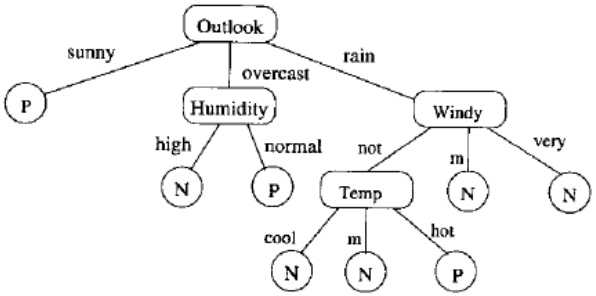
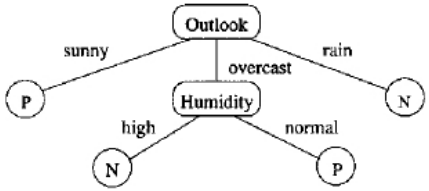


FIGURE 5.5. Perfect decision tree



Example

No.	Attributes				Class
	Outlook	*Temperature*	*Humidity*	*Windy*	
1	overcast	hot	high	not	N
2	overcast	hot	high	very	N
3	overcast	hot	high	medium	N
4	sunny	hot	high	not	P
5	sunny	hot	high	medium	P
6	rain	mild	high	not	N
7	rain	mild	high	medium	N
8	rain	hot	normal	not	P
9	rain	cool	normal	medium	N
10	rain	hot	normal	very	N
11	sunny	cool	normal	very	P
12	sunny	cool	normal	medium	P
13	overcast	mild	high	not	N
14	overcast	mild	high	medium	N
15	overcast	cool	normal	not	P
16	overcast	cool	normal	medium	P
17	rain	mild	normal	not	N
18	rain	mild	normal	medium	N
19	overcast	mild	normal	medium	P
20	overcast	mild	normal	very	P
21	sunny	mild	high	very	P
22	sunny	mild	high	medium	P
23	sunny	hot	normal	not	P
24	rain	mild	high	very	N

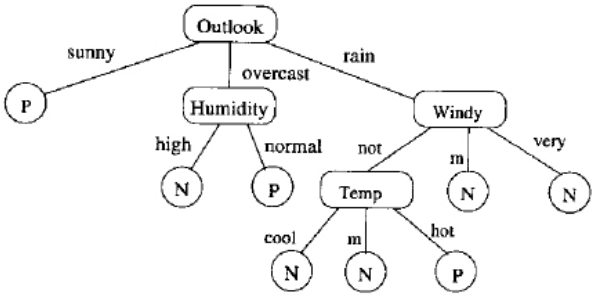
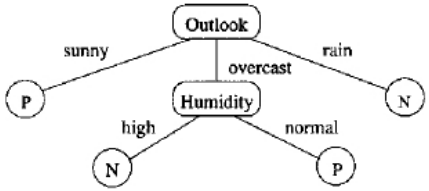


FIGURE 5.5. Perfect decision tree



+ single exception

Example

No.	Attributes				Class
	Outlook	*Temperature*	*Humidity*	*Windy*	
1	overcast	hot	high	not	N
2	overcast	hot	high	very	N
3	overcast	hot	high	medium	N
4	sunny	hot	high	not	P
5	sunny	hot	high	medium	P
6	rain	mild	high	not	N
7	rain	mild	high	medium	N
8	rain	hot	normal	not	P
9	rain	cool	normal	medium	N
10	rain	hot	normal	very	N
11	sunny	cool	normal	very	P
12	sunny	cool	normal	medium	P
13	overcast	mild	high	not	N
14	overcast	mild	high	medium	N
15	overcast	cool	normal	not	P
16	overcast	cool	normal	medium	P
17	rain	mild	normal	not	N
18	rain	mild	normal	medium	N
19	overcast	mild	normal	medium	P
20	overcast	mild	normal	very	P
21	sunny	mild	high	very	P
22	sunny	mild	high	medium	P
23	sunny	hot	normal	not	P
24	rain	mild	high	very	N

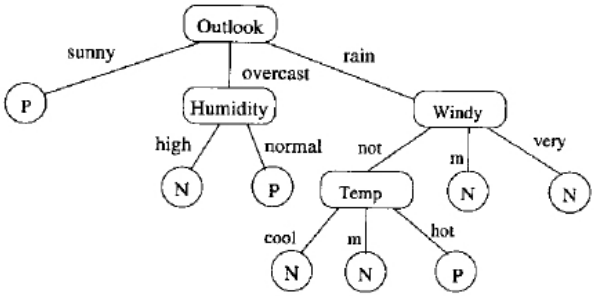
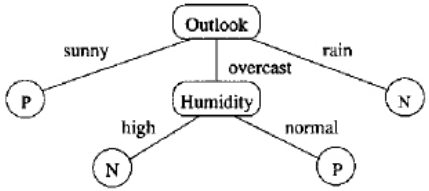


FIGURE 5.5. Perfect decision tree



+ single exception

How to code a decision tree and exceptions

1. Perfect tree

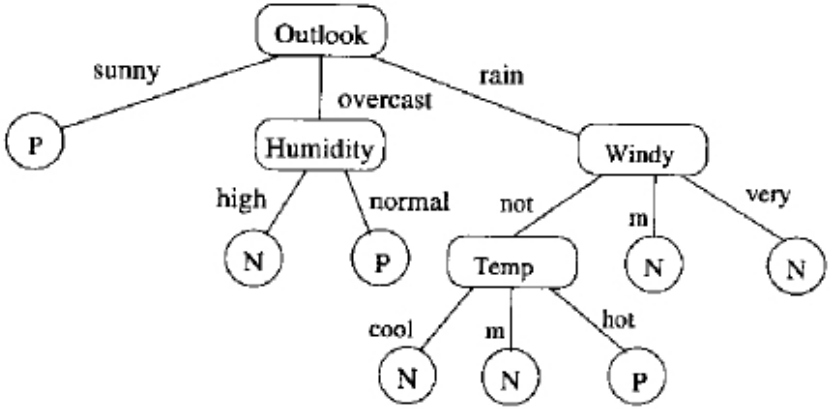


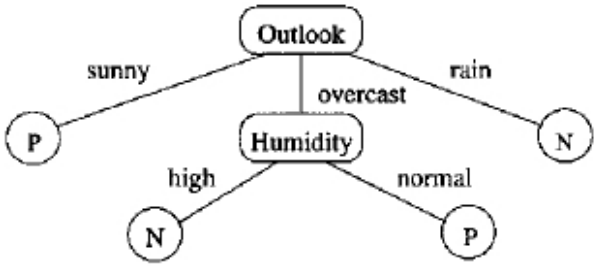
FIGURE 5.5. Perfect decision tree

For the tree in Figure 5.5, she writes down

1 Outlook 0 P 1 Humidity 0 N 0 P 1 Windy 0 N 0 N
1 Temperature 0 N 0 N 1 P.

How to code a decision tree and exceptions

2. Imperfect tree



1 Outlook 0 P 1 Humidity 0 N 0 P 0 N.

Coding the Exceptions Since the decision tree in Figure 5.4 is not perfect, we need to indicate where the exceptions are. In this case there is a single exception. The most straightforward way is to indicate its *position* among all 54 possible combinations of attributes. This costs $\log 54 \approx 5.75$ extra bits.

How to code a decision tree and exceptions

Perfect tree

For the tree in Figure 5.5, she writes down

1 Outlook 0 P 1 Humidity 0 N 0 P 1 Windy 0 N 0 N
1 Temperature 0 N 0 N 1 P.

Imperfect tree

1 Outlook 0 P 1 Humidity 0 N 0 P 0 N.

Coding the Exceptions Since the decision tree in Figure 5.4 is not perfect, we need to indicate where the exceptions are. In this case there is a single exception. The most straightforward way is to indicate its *position* among all 54 possible combinations of attributes. This costs $\log 54 \approx 5.75$ extra bits.

Summary

Thus, the encoding using the decision tree in Figure 5.4 uses 19.335 bits; the encoding using the decision tree in Figure 5.5 uses 25.585 bits. The MDL principle tells us to use the former method, which is also much shorter than the 54-bit plain encoding.

Information sources

- Tom Mitchell, Machine Learning
- Peter Flach Book
- http://work.caltech.edu/lectures.html
- Li & Vitanyi, An Introduction to Komogorov Complexity and Its Applications