14 Minimum Description Length Principle (MDL)

- Bayes Theorem:

$$P(H_i|E) = \frac{P(H_i)P(E|H_i)}{\sum_{i=1}^{n} P(H_i)P(E|H_i)}$$

- Take a $-\log$ of both sides of Bayes:

$$-\log_2 P(H_i|E) = -\log_2 P(H_i) - \log_2 P(E|H_i) + C$$

- Information in message A (minimal length of A in bits): $\log_2 P(A) = I(A) = L(A)$
- Then: $L(H|E) = L(H) + L(E|H)$
- MDL: The hypothesis must reduce the information needed to encode the data, i.e.

$$L(E) > L(H) + L(E|H)$$

- The best hypothesis must maximize *information compression*:

$$L(E) - L(H) - L(E|H)$$

15 Computing the description length of H and E

Assume that H is represented as a set of rules, $L(H) = ?$

- $H = R_1 \vee R_2 \vee ... \vee R_m$
- R_i: if $t_1 \wedge t_2 \wedge ... \wedge t_{k_i}$ then C_j, where $j = 1, ..., N$.
- Let ts be the total number of tests t_i (attribute-value pairs) found in data. This is the number of different values for each attribute summed over all attributes, excluding the class attribute (for example, in PlayTennis $ts = 10$).
- To identify the rule R_i left-hand side, we choose k_i out of ts tests. This can be done in $\binom{ts}{k_i}$ different ways.
- Then the probability of each particular choice is $\frac{1}{\binom{ts}{k_i}}$.
- According to information theory (Shannon and Weaver, 1949) the optimal description length of the message that this choice has been made is

$$-\log_2 \frac{1}{\binom{ts}{k_i}}$$

- Then the description length of R_i is:
  - The description length of the rule's left-hand side.
  - Plus the number of bits needed to encode the class value, which is $\log_2 N$, where N is the number of classes.
- Then:

$$L(R_i) = -\log_2 \frac{1}{\binom{ts}{k_i}} + \log_2 N = \log_2 \binom{ts}{k_i} + \log_2 N$$

$$L(H) = \sum_{i=1}^{m} (\log_2 \binom{ts}{k_i} + \log_2 N)$$

$L(E) = ?$

- Apply the same technique for E. Both E and H use the same representation, i.e. each example can be represented as a rule with the number of tests (attribute-values pairs) in the left-hand side equal to the number of attributes (the same for all examples).

16 Encoding exceptions ($L(E|H)$) for two-class hypotheses

Well be encoding here the exceptions, i.e. what we need in addition to H in order to represent E. In other words, if we know H we know E only partially (of course, if H is not 100% correct). Then, in order to know E completely, we need to know the corrections to H too, which are represented by the term $L(E|H)$. We start with the confusion matrix.

- Confusion matrix:

Actual \ Predicted by H	$+$	$-$
$+$	tp	fn
$-$	fp	tn

- H predicts class $+$ for $(tp + fp)$ examples: tp correctly and fp incorrectly.
- H predicts class $-$ for $(tn + fn)$ examples: tn correctly and fn incorrectly.
- We have to find the description length of the incorrect predicitons (exceptions) fp and fn.
- fp examples out of $(tp + fp)$ examples can be chosen in $\binom{tp+fp}{fp}$ different ways.
- fn examples out of $(tn+fn)$ examples can be chosen in $\binom{tn+fn}{fn}$ different ways.
- Using the probabilities of choosing fp examples out of $(tp+fp)$ and fn examples out of $(tn + fn)$, we get:

$$L(E|H) = \log_2 \binom{tp + fp}{fp} + \log_2 \binom{tn + fn}{fn}$$

17 Encoding entropy

The encoding of exceptions approach has two major drawbacks:

- The evaluation is *symetric* with respect to the tp/fp and tn/fn ratios. This is due to the symetry of the binomial coefficients:

$$\binom{n}{k} = \binom{n}{n-k}$$

  This means for example, that two hypotheses both covering 100 instances – the one with 1 fp and the other with 99 fp's, will be equivalent with respect to their $L(E|H)$ measure (!).

- If the hypothesis is 100% accurate ($fp = 0$ and $fn = 0$) then $L(E|H) = 0$. In this situation we cannot evaluate the distribution of the examples that the hypothesis coves.

To avoid the above mentioned drawbacks in some cases we may apply other MDL evaluation measures, based on *entropy*. Here is an example of one simple entropy-based measure:

- For each rule R_i in H calculate:

$$e_i = -\frac{p_i}{n_i} * \log_2 \frac{p_i}{n_i} - \frac{n_i - p_i}{n_i} * \log_2 \frac{n_i - p_i}{n_i},$$

- Where p_i is the number of positive examples covered by R_i, and n_i is the total number of examples covered by R_i.

- Then

$$L(H) + L(E|H) = \sum_i |R_i| + e_i * n_i$$

- Where $|R_i|$ is the number of tests (attribute-value pairs) in rule R_i.

- Since *different encodings* are used for $L(E)$, $L(H)$ and $L(H|E)$, the above formula can be used to *compare hypotheses only*, but not as a measure of the actual compression.

- Where p_i is the number of positive examples covered by R_i, and n_i is the total number of examples covered by R_i.
- Then

$$L(H) + L(E|H) = \sum_i |R_i| + e_i * n_i$$

- Where $|R_i|$ is the number of tests (attribute-value pairs) in rule R_i.
- Since *different encodings* are used for $L(E)$, $L(H)$ and $L(H|E)$, the above formula can be used to *compare hypotheses only*, but not as a measure of the actual compression.

4 Example (encoding exceptions)

4.1 Encoding Data

Outlook	Temperature	Humidity	Wind	Play
x_1	x_2	x_3	x_4	y
sunny	hot	high	weak	no
sunny	hot	high	strong	no
overcast	hot	high	weak	yes
rain	mild	high	weak	yes
rain	cool	normal	weak	yes
rain	cool	normal	strong	no
overcast	cool	normal	strong	yes
sunny	mild	high	weak	no
sunny	cool	normal	weak	yes
rain	mild	normal	weak	yes
sunny	mild	normal	strong	yes
overcast	mild	high	strong	yes
overcast	hot	normal	weak	yes
rain	mild	high	strong	no

Each tuple can be represented as a set of attribute value pairs as follows:

$$
\begin{aligned}
E = \{&\{x_1 = sunny, x_2 = hot, x_3 = high, x_4 = weak\}\\
&\{x_1 = sunny, x_2 = hot, x_3 = high, x_4 = strong\}\\
&\{x_1 = overcast, x_2 = hot, x_3 = high, x_4 = weak\}\\
&...\\
&\}
\end{aligned}
$$

$L(E) =?$

Excluding the class attribute, the total number of different attribute value pairs is $ts = 10$.

In each tuple we have 4 pairs involving independent attributes, which may be chosen in $\binom{10}{4} = \frac{10!}{4! \times 6!} = 210$ different ways, and one pair with the class attribute, which may be chosen in 2 differnt ways.

Thus the probability of choosing a tuple is $\frac{1}{210} \times \frac{1}{2}$.

The code length in bits for this choice is $-\log_2 \frac{1}{210} - \log_2 \frac{1}{2} = \log_2 210 + \log_2 2 = 7.715 + 1 = 8.715$, which is also the number of bits to encode a tuple with its class value.

Then the code length of the whole data set is

$$L(E) = 8.715 \times 14 = 122.01$$

4.2 Hypoteses

Consider applying the Prizm algorithm to our data and a situation where we have generated two rules for class *yes* and are considering whether or not to add more rules, i.e. we have two competing hypotheses H_1 and H_2 (for class *no* we use CWA):

H_1: If $\{outlook = overcast\}$ Then play=yes
If $\{humidity = normal, wind = weak\}$ Then play=yes

H_2: If $\{outlook = overcast\}$ Then play=yes
If $\{humidity = normal, wind = weak\}$ Then play=yes
If $\{temperature = mild, humidity = normal\}$ Then play=yes

Let's compute the compression of H_1 and H_2. First we need the length of each hypothesis – $L(H_1)$ and $L(H_2)$.

As we do not include the class attribute in the rule conditions (and no need to encode it at all, because all rules are for just one class), the total number of tests (ts) is now 10. Summing over the rules (the number of bits to encode the rule conditions) in each hypothesis, we get:

$$L(H_1) = \log_2 \binom{10}{1} + \log_2 \binom{10}{2} = \log_2 10 + \log_2 45 = 3.32 + 5.49 = 8.81$$

$$L(H_2) = \log_2 \binom{10}{1} + \log_2 \binom{10}{2} + \log_2 \binom{10}{2} = \log_2 10 + 2 \times \log_2 45 = 14.30$$

Next, we need to encode the exceptions. For this purpose let's compute the confusion matrices first. Again for class *no* we use CWA (i.e. tuples not covered by any rule are predicted as belonging to class *no*).

H_1:

Actual \ Predicted	*yes*	*no*
yes	7	2
no	0	5

H_2:

Actual \ Predicted	*yes*	*no*
yes	8	1
no	0	5

According to the formula for $L(E|H)$ we have to compute the number of bits needed to encode the exceptions (wrong classifications).

$$L(E|H_1) = \log_2 \binom{7}{0} + \log_2 \binom{7}{2} = \log_2 1 + \log_2 21 = 0 + 4.39 = 4.39$$

$$L(E|H_2) = \log_2 \binom{8}{0} + \log_2 \binom{6}{1} = \log_2 1 + \log_2 6 = 0 + 2.59 = 2.59$$

Then the compression for each hypothesis is

$$\mathit{compr}(H_1) = L(E) - L(H_1) - L(E|H_1) = 122.01 - 8.81 - 4.39 = 108.81$$

$$\mathit{compr}(H_2) = L(E) - L(H_2) - L(E|H_2) = 122.01 - 14.30 - 2.59 = 105.12$$

The above result shows that H_1 has better compression that H_2. So, the conclusion is that we can stop generating Prizm rules and deliver H_1 as a good model of our data.