Answers Final Exam Machine Learning
Normal Group,
Resit-of-intermediate-exam Version

January 15, 2008

Grading works as follows: You start with 1 point, and for each of the 12 subquestions you can get 3/4 points. Partial points may be awarded for partially correct answers.

1. (a) The hypothesis h_1 is a member of $\mathcal{H}$. To see this, note that x_2 can only take the values 'Warm' and 'Cold'. Hence h_1 can more simply be expressed as

$$h_1(\mathbf{x}) = \begin{cases} \text{Yes} & \text{if } x_3 = \text{'Normal'}, \\ \text{No} & \text{otherwise,} \end{cases}$$

which in Mitchell's notation becomes $\langle ?, ?, \text{Normal}, ?, ?, ? \rangle$.

The hypothesis h_2 is not a member of $\mathcal{H}$. The reason is that the constraints cannot represent dependencies between attributes: The first case in the definition of h_2 implies that any value is allowed for x_3. In addition, the first two cases together imply that x_2 can also take any value, and x_1 can at least take the value 'Sunny'. But then an input $\mathbf{x}$ with x_1 = 'Sunny', x_2 = 'Warm' and x_3 = 'Normal', which is classified as 'No' by h_2, would also be classified as 'Yes' by any choice of constraints that is consistent with the first two cases of h_2.

(b) For example,

$$h(\mathbf{x}) = \begin{cases} \text{Yes} & \text{if } x_2 = \text{'Warm' and } x_3 = \text{'Normal'}, \\ \text{Yes} & \text{if } x_2 = \text{'Cold' and } x_3 = \text{'High'}, \\ \text{No} & \text{otherwise.} \end{cases}$$

See the answers to the second set of homework exercises for an elaborate discussion.

2. It greedily selects attributes with the highest information gain. These are the attributes that it estimates to have the highest mutual information with the class labels. So it puts the attributes that it thinks give the most information about the class labels at the top.

3. (a) The hypothesis space of 14th degree polynomials would still be much too large relative to the number of data points, so least squares would still overfit if we used a sum of absolute errors.

(b) Increasing the number of data points does help against overfitting: If we have an extremely large train set and we find a hypothesis that has small error on the train set, then this is most likely not due to luck, even if we have searched a pretty large hypothesis space.

Compare this to the dice prediction game: If we throw the die only two times, then it is pretty likely that some student will predict both throws correctly just by luck. But if we throw the die, say, 100 000 times and a student predicts most of the outcomes correctly, then we may take this as a strong indication that the student will be good at predicting future throws as well, even if we searched among a group of 50 students to find this one.

4. (a)

k	1	3	5
Instance 1	White	Black	Black
Instance 2	White	White	Black

(b) Representing x_1 by assigning integers to Black, White and Brown does not work, because the difference between these integers would be meaningless. One way to represent x_1 would be as

Value	Black	White	Brown
Representation	$\begin{pmatrix}1\\0\\0\end{pmatrix}$	$\begin{pmatrix}0\\1\\0\end{pmatrix}$	$\begin{pmatrix}0\\0\\1\end{pmatrix}$

The other feature, x_2 can just be represented by its own value. The feature vector $\mathbf{x}$ can now be composed from the three components of x_1 and one component for x_2.

For example, x_1 = 'Brown' and $x_2 = 32$ would give $\mathbf{x} = \begin{pmatrix}0\\0\\1\\32\end{pmatrix}$.

5. Examples that are not linearly separable can never be classified correctly by the perceptron. So for example if the target function is xor, a perceptron will always make at least one mistake if we see all possible inputs:

$$D = \begin{array}{|c||r|r|r|r|}\hline y & 1 & 1 & -1 & -1 \\ \hline x_1 & -1 & 1 & 1 & -1 \\ \hline x_2 & 1 & -1 & 1 & -1 \\ \hline \end{array}$$

6. Naive Bayes would classify the new example as True:

$$\begin{aligned} &P(X_1 = \text{True} \mid Y = \text{False})P(X_2 = \text{True} \mid Y = \text{False})P(Y = \text{False}) = 0 \cdot 0 \cdot \frac{1}{3} = 0 \\ < &P(X_1 = \text{True} \mid Y = \text{True})P(X_2 = \text{True} \mid Y = \text{True})P(Y = \text{True}) = \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{2}{3} = \frac{1}{6} \end{aligned}$$

7. (a) Maximum likelihood would select P_1:

$$P_1(D) = (9/10)^5(1/10)^3 > (1/10)^5(9/10)^3 = P_2(D).$$

(b) MAP with the given prior would select P_2:

$$\begin{aligned} \pi(\theta = 1 \mid D) &= \frac{P_1(D)\pi(1)}{P_{\text{Bayes}}(D)} = \frac{(9/10)^5(1/10)^3 \cdot 1/100}{P_{\text{Bayes}}(D)} \\ &= \frac{9^5/10^{10}}{P_{\text{Bayes}}(D)} \\ \pi(\theta = 2 \mid D) &= \frac{P_2(D)\pi(2)}{P_{\text{Bayes}}(D)} = \frac{9^3 \cdot 99/10^{10}}{P_{\text{Bayes}}(D)} \\ &> \frac{9^3 \cdot 81/10^{10}}{P_{\text{Bayes}}(D)} = \frac{9^5/10^{10}}{P_{\text{Bayes}}(D)} = \pi(\theta = 1 \mid D) \end{aligned}$$

8. The promiscuous grammar doesn't help at all in compressing the text, because it can generate all possible texts. So, although $L(H)$ is small for the promiscuous grammar, $L(D \mid H)$ is at least as large as the uncompressed text.

The ad hoc grammar also doesn't help at all in compressing the text. It can only generate the given text, so $L(D \mid H)$ is very small, but the encoding of the grammar contains a literal description of the text and therefore $L(H)$ is at least as large as the uncompressed text.

Finally, for the 'right' grammar, the number of grammatically correct texts is exponentially smaller (in n) than the number of possible texts. Therefore the difference between $L(D \mid H)$ and the size of the uncompressed text becomes larger and larger if we look at larger and larger values of n. As the grammar doesn't change with increasing n, its codelength $L(H)$ is constant. Therefore, for sufficiently large n, also the total codelength $L(H) + L(D \mid H)$ for the 'right' grammar is much smaller than the size of the uncompressed text.

Together these arguments imply that $L(H) + L(D \mid H)$ will be smallest for the 'right' grammar (for sufficiently large n), and hence this grammar will be selected by two-part MDL.