T-61.5040 Oppivat mallit ja menetelmät
T-61.5040 Learning Models and Methods
Pajunen, Viitaniemi

Exercises 1, 19.1.2007

Problem 1.

An important task in modeling is to limit the set of possible functions among which the model is chosen. Otherwise, the observations cannot be extrapolated or interpolated, that is, the model does not generalize to new observations.
As an example of this problem we try to model a black box whose input and output can be measured. Describe how you can fit a predefined model exactly into the observations, making as little modifications to the model as possible, if

i) we do not assume anything of the set of functions.
ii) we assume that the model is continuous.
iii) we assume that the model is smooth, that is, infinitely many times differentiable.

Problem 2.

Consider gray shade images of size 256 × 256 where the shade is represented using 256 levels (8 bits per pixel, $2^8 = 256$). Assume that we compress the figures using function K that gives an image X_i a binary code b_i, that is, $K(X_i) = b_i$.

i) How many bits does the image consist of before the compression?
ii) Assume that you wish to compress images so that the size of the code b_i is at least 10 bits shorter than the size of the original image. What percentage of all possible gray shade images can be compressed using this kind of code, if you still wish to obtain lossless compression?
iii) Consider an image of a natural scene, of the above size. Do you think that compressing it 10 bits shorter is as difficult as it seems in Problem ii)? If not, why?

Problem 3.

Assume you have a bag full of binary vectors of length n (n is even). Suppose that your learning problem is to predict the n:th bit given the $n-1$ first bits when a vector is randomly drawn (with replacement) from the bag. Design an optimal prediction method when

i) you know the bag contains one of each possible binary vectors.
ii) you don't know anything about the bag's contents.
iii) you have 10 training vectors which all have different first $n-1$ bits. Mining through different functions $f : \{0,1\}^{n-1} \rightarrow \{0,1\}$ you find a function f_1 which works perfectly on the training set.
iv) you know that the bag contains exactly 10% of all possible binary vectors and they are all different.

Problem 4.

Assume the same type of bag as in problem 3, part iv), and assume nothing else. You are given an imaginary algorithm to be evaluated on the bag, called the Binary Vector Machine, which is a general learning algorithm independent of your binary bag. You draw a vector from the bag and predict the last bit by BVM. The prediction is correct. You draw more vectors and BVM always predicts the last bit correctly! This happens n times in total. You draw the $n+1$:st vector. What is the expected prediction error using BVM?

Problem 5. (demo) In this problem, consider the set of all bags of 2^{n-1} binary vectors of length n such that the first $n-1$ bits of each vector are different. This means that in each bag there is exactly one vector that matches any given bit pattern of length $n-1$.

Compare two algorithms which try to predict the nth bit as a function of the $n-1$ first bits. Algorithm A guesses the last bit with equal probability for 0 and 1. Algorithm B is any algorithm: lets call it the Binary Vector Machine.

When each algorithm is applied to a bag described above, denote the performance by C_A and C_B. These are average prediction errors and are defined as the number of errors divided by 2^{n-1}.

Estimate the fraction of bags for which $|C_A - C_B| \geq \epsilon$ when the bags contain 2^{19} vectors and $\epsilon = 1/128$. You need to use either the Chebyshev inequality or the Hoeffding inequality (easier?) to do this.

Hoeffding inequality:

If $x_1, \ldots, x_n$ are iid random variables for which $x_i - E(x_i) \in [a_i, b_i]$ and $X = \sum_i x_i$, then

$$P(X - E(X) \geq \epsilon) \leq \exp(-2\epsilon^2 / \sum_i (b_i - a_i)^2).$$

(Reminder: Chebyshev: $P(|D - E(D)| > \epsilon) \leq \frac{1}{\epsilon^2}\mathrm{Var}(D)$.)

The purpose of this problem is to show that the proportion of learning problems in which *any algorithm* can beat guessing even by a little margin is very small.

Note: Some copies of the solutions are delivered in the exercise sessions. Extra copies (if any) can be found in the clear plastic boxes in the beginning of the CIS lab corridor. The (possibly corrected) solutions only appear in the course web page in the end of the course, though.