582669 Supervised Machine Learning (Spring 2011)
Homework 2 (3 February)

Turn this homework in no later than **Tuesday, 1 February, at 15:00**.

1. We consider the Weighted Majority algorithm (page 31 of the lecture notes) with η and c as in Theorem 1.5 (page 34). In this problem you are asked to generalise the proof of Theorem 1.5.

   (a) Show that if for some $M \geq 0$ there are k different hypotheses h_i that all satisfy $L_{0,1}(S, h_i) \leq M$, then
   $$L_{0,1}(S, \mathrm{WM}) \leq c\eta M + c \ln \frac{n}{k}.$$

   (b) Change the initialisation of the algorithm so that $w_{1,i} = p_i$ for all i, where $p_i > 0$ for all i and $\sum_{i=1}^{n} p_i = 1$ but otherwise $\boldsymbol{p}$ is arbitrary. Show that for the modified algorithm WM' we have
   $$L_{0,1}(S, \mathrm{WM}') \leq \min_{1 \leq i \leq n} \left(c\eta L_{0,1}(S, h_i) + c \ln \frac{1}{p_i} \right).$$

2. We modify the Weighted Majority algorithm (page 31) as follows, in order to use it for soft classification with logarithmic loss:

   - In Step 5 we set $w_{t+1,i} = w_{t,i} \exp(-\eta L_{\log}(y_t, h_i(x_t))$, where $\eta = -\ln \beta$. (Notice that the original update can be written as $w_{t+1,i} = w_{t,i} \exp(-\eta L_{0-1}(y_t, h_i(x_t))$, so this seems a natural generalisation.)
   - Instead of the majority vote of Steps 3 and 4, we predict with the weighted average
   $$\hat{y}_t = \sum_{i=1}^{n} v_{t,i} h_i(x_t)$$
   where $v_{t,i} = w_{t,i} / \sum_{j=1}^{n} w_{t,j}$.

   As before, consider the potential $P_t = c \ln W_t$ where $W_t = \sum_{j=1}^{n} w_{t,j}$. Show that if $\eta = c = 1$ we have
   $$L_{\log}(y_t, \hat{y}_t) = P_t - P_{t+1}.$$
   What loss bound do you get from this?

   Notice: You should also assume that the hypotheses h_i are soft classifiers, i.e. functions $h_i \colon X \to [0, 1]$.

3. We take 1000 fair coins and toss each of them 10 times. All tosses are assumed to be independent.

   (a) What is the probability for a given coin to come up heads 10 times?

   (b) What is the probability that at least one coin comes up heads 10 times? Calculate the exact value.

   (c) Derive an upper bound for at least one coin coming up heads 10 times, using the results from part (a) and the union bound. Compare with the exact value from part (b).

Continues on the next page!

4. *Boolean formulae* are a common form of representing binary classifiers over $X = \{0,1\}^n$. We introduce Boolean variables $v_1, \ldots, v_n$, where v_i intuitively means that $x_i = 1$.

   A *monotone conjunction* is a formula of the form $v_{i_1} \wedge \ldots \wedge v_{i_k}$. It represents a classifier $f\colon \{0,1\}^n \to \{0,1\}$ such that $f((x_1, \ldots, x_n)) = 1$ if $x_{i_1} = \cdots = x_{i_k} = 1$. Also such classifiers f are called monotone conjuntions.

   More generally, a *conjunction* is of the form $\tilde{v}_{i_1} \wedge \ldots \wedge \tilde{v}_{i_k}$, where $\tilde{v}_j$ is either v_j or $\overline{v_j}$. It represents a classifier f where $f((x_1, \ldots, x_n)) = 1$ if $x_j = 1$ for all j such that v_j appears in the conjunction, and $x_j = 0$ for all j such that $\overline{v_j}$ appears in the conjunction,

   (a) Give an efficient algorithm that receives as input a sample $((\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m))$ where $(\boldsymbol{x}_j, y_j) \in \{0,1\}^n \times \{0,1\}$, and outputs a monotone conjunction that is consistent with the sample, if one exists. In other words, give an algorithm for Empirical Risk Minimisation of monotone conjunctions in the noise-free PAC model. Your algorithm is not required to work if there is no consistent monotone conjunction. Use Theorem 1.7 to estimate how large a sample you need for $n = 100$, $\varepsilon = 0.1$ and $\delta = 0.001$.

   Hint: find the longest consistent monotone conjunction.

   (b) Use you algorithm from part (a) as a basis for a similar algorithm for general conjunctions. Again, use Theorem 1.7 to estimate how large a sample you need for $n = 100$, $\varepsilon = 0.1$ and $\delta = 0.001$.

   Hint: transform input vectors $(x_1, \ldots, x_n)$ into $(x_1, \ldots, x_n, 1 - x_1, \ldots, 1 - x_n)$.

   (c) **For extra credit** (worth one regular problem). You may wish to skip this unless you are familiar with NP-completeness. Consider learning monotone conjunctions with Empirical Risk Minimisation in the agnostic PAC model. Show that the problem

   input: a sample $((\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m))$

   output: a monotone conjunction f such that $\hat{R}(f)$ is as small as possible (for a monotone conjunction f)

   is NP-hard. The difference to part (a) is that now we do not assume the existence of a consistent monotone conjunction.

   Hint: reduction from Set Cover.

Remark: Generally on this course we do not spend much time on computational complexity. However it is worth noticing that Empirical Risk Minimisation in the agnostic case is often computationally much more difficult than in the noise-free case.