Pattern Recognition - Exam

(08/04/2015) 2h30: open book exam with a pocket calculator

E. Fromont, A. Habrard and M. Sebban

Exercise: Machine Learning Theory MCQ (3 points)

-Circle the letter corresponding to the correct answer (only one is correct).
-You can leave questions unanswered. Each correct answer adds one point.
-Each incorrect answer subtracts half a point.

1. A training set of labeled examples is a sample drawn over a space $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ is the feature space and $\mathcal{Y}$ is the set of classes. What does $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ mean?

   a. $\mathcal{Z}$ is a joint space over $\mathcal{X}$ and $\mathcal{Y}$.

   b. $\mathcal{Z}$ corresponds to the product of the features by the class.

   c. $\mathcal{Z}$ is a finite set of examples jointly drawn from $\mathcal{X}$ and $\mathcal{Y}$.

2. Why is h called *hypothesis* in machine learning?

   a. Because h is like a random guessing.

   b. Because a statistical test is performed on h w.r.t. an alternative hypothesis.

   c. Because h corresponds to one of the possible models which fit the data.

3. What is the right assertion:

   a. The higher the variance, the higher the risk of overfitting.

   b. The smaller the bias, the smaller the variance.

   c. The total error of a classifier is the sum of the bias and the variance.

4. The following three hypotheses are supposed to be consistent with respect to a given training set S. According to Occam's razor principle, what is the best one?

   a. $y = 2x^2 + x + 1$.

   b. $y = \sqrt{5}x + 2\sqrt{\pi}$.

   c. $(x-1)^2 + (y-1)^2 = 2$.

5. The higher the number of iterations T of Adaboost

   a. The smaller $\prod_{t=1}^{T} Z_t$.

   b. The larger the risk of overfitting.

   c. The higher the Vapnik-Chervonenkis dimension of the weak classifiers.

6. The true risk of a kNN algorithm
   a. Always decreases when k grows.
   b. Is lower bounded by ϵ^*.
   c. Tends towards 0 when the number of training examples tends to the infinity.

Exercise 2: Decision Trees (3 points)

Consider the following set of training examples:

Ex	Target	a1	a2
1	+	F	F
2	+	F	T
3	-	F	F
4	+	T	F
5	-	T	T
6	-	F	T
7	-	T	F
8	+	F	T

1. (0.5 pt) What is the entropy of this collection of training examples with respect to the target attribute?
2. (1 pt) What is the information gain of a1 and a2 on these training examples?
3. (1 pt) Draw the decision tree returned by the ID3 algorithm on this example (use the mode of the Target as prediction in each leaf, in case of a tie, you can predict +)
4. (0.5 pt) What would be the accuracy of your tree on the following test set:

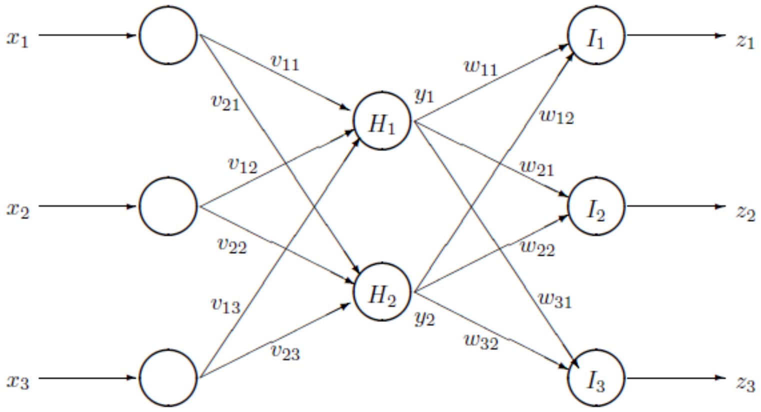
Ex	Target	a1	a2
9	-	T	F
10	-	F	T

Exercise 3: Artificial Neural Networks (3 point)

Consider a unique neuron with 4 entries, a weight vector equal to $\mathrm{w} = [1, 2, 3, 4]^T$ and a bias $\theta = 0$ (zero). Its activation function is linear with a proportionality constant equal to 2 (i.e. $f(x) = 2x$).

1. (1 pt) If the input vector is $\mathrm{x} = [4, 8, 5, 6]^T$, what would be the output of the neuron ? (the possible answers are A= 1. B=56. C=59. D=112. E=118. Explain your answer).

Now we work with the following network:



An input example x = $[x1, x2, x3]^T$ (with true label t = $[t1, t2, t3]^T$) is presented to the network. The computed output is z = $[z1, z2, z3]^T$.

2. (0.75 pt) What would be the usual computing sequence to train this network using a standard back-propagation algorithm ? (the possible answers are A, B, C and D. Explain very briefly why you chose an answer)

```
A. (1) compute yj = f(Hj), (2) compute zk = f(Ik),
   (3) update vji , (4) update wkj.

B. (1) compute yj = f(Hj), (2) compute zk = f(Ik),
   (3) update wkj , (4) update vji.

C. (1) compute yj = f(Hj), (2) update vji,
   (3) compute zk = f(Ik), (4) update wkj.

D. (1) compute zk = f(Ik), (2) update wkj,
   (3) compute yj = f(Hj), (4) update vji.
```

3. (1.25 pts)The weight vector of the network is $v1 = [0.4, -0.6, 1.9]^T$, $v2 = [-1.2, 0.5, -0.7]^T$, $w1 = [1.0, -3.5]^T$, $w2 = [0.5, -1.2]^T$, $w3 = [0.3, 0.6]^T$. For each neuron, the bias θ is fixed to 0. The activation function of each neuron is a sigmoid. If the input vector is $x = [1.0, 2.0, 3.0]^T$, what would be the output computed for y_1 (only the output of the first hidden layer). (the possible answers are A= -2.300, B= 0.091, C= 0.644, D= 0.993, E= 4.900. Explain your answer)

Exercise 4: SVM and Optimization (7 points)

4.1 - True or false (1.5 points)

1. The dual form of the SVM soft margin does not verify the strong duality property.
2. The kernel trick cannot be used in the perceptron learning algorithm.
3. Optimizing the margin leads to better generalization bound.

4.2 - Optimization (3 points) Let $S = \{(\mathbf{x_i}, y_i)\}_{i=1}^n$ be a learning sample such that $\mathbf{x_i} \in \mathbb{R}^d$ and $y_i \in \{-1, 1\}$, consider now the following optimization problem:

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \|\mathbf{w}\|_1 \\ \text{subject to} \quad & \ell_i(\langle \mathbf{w}, \mathbf{x_i} \rangle + b) \geq 2, \quad 1 \leq i \leq n. \end{aligned}$$

1. Relax the problem to allow some amount of constraint violation (introduce slack variables).
2. Reformulate the resulting problem to avoid using the $\|\cdot\|_1$ norm and put it in standard form.

4.3 - SVM dual formulation (2.5 points) We would like to implement an SVM solver with AMPL. We assume to have access to a parameter file encoding the learning data, as presented below:

```
set dim:= 1 2;
set examples:= 1 2 3 4;
param points :=
  1 1 1
  1 2 1
  2 1 0
  2 2 1
  3 1 1
  3 2 0
  4 1 2
  4 2 0;

param labels := 1 1
 2 1
 3 -1
 4 -1;
```

Write an AMPL code corresponding to an implementation of the SVM soft-margin dual form.

Exercise 5: HMM (4 points)

We would like to create a tool able to produce some human laugh such has *hahaha* or *hihihi*. To simplify, we assume that this tool can only produce 3 sounds denoted by the following symbols: `ha, hi, hou` and we build the tool with 3 blocks able to produce the sounds. Each block can generate the sounds randomly such that:

- the first block can only produce `ha` or `hi`, and the probability of getting an `hi` is strictly greater than the proba of getting an `ha`,
- the second can only generate `hou, hi` in a uniform way,
- the last one can generate the 3 sounds such that the probability of `ha` is greater than the sum of the probability of `hi` and `hou`.

In this context, a sequence `ha ha hi hi hou` represents a sequence of sounds produced randomly by the blocks. To generate a sequence, we fix a target length, say n, and we generate n sounds with the tool. Then a first block is chosen randomly from a uniform distribution among the 3 blocks. Then, we choose randomly a sound among the possible ones in the considered block, again according to a uniform distribution over the sounds. Then, we move to the next block, all the blocks have the same probability to be chosen from given a block.

1. Propose a way to model this tool by a HMM. Draw the HMM graphically (states, transitions and probabilities).
2. Compute the probability to generate the sequence `hou hou ha`.
3. What is the most likely sequence of states that explains the previous sequence?