# "Machine Learning - Fundamentals and algorithms/ Pattern Recognition" Exam

**(29/03/2018) 2h00: one handwritten A4 sheet allowed.**

R. Emonet, A. Habrard

## Exercise: SVM (5 points)

1. Recall the definition of a kernel.
2. Draw some 2D plots illustrating binary classification problems (with the decision frontier) that:
   (a) can be solved by a linear kernel.
   (b) can be solved by a polynomial kernel but not a linear kernel.
   (c) can be solved by a Gaussian kernel, but not by a polynomial nor a linear one.
3. SVM are able to learn classifiers for binary classification from a sample of labeled examples $\{(\mathbf{x}_i, \ell_i)\}_{i=1}^n$ where $\mathbf{x}_i \in \mathbb{R}^d$ and $\ell_i \in \{-1, 1\}$. We recall here the optimization problem of a linear SVM with soft margin:

$$\begin{aligned} \min_{\mathbf{w},b,\boldsymbol{\xi}} \quad & \tfrac{1}{2}\|\mathbf{w}\|_2^2 \ + \ C\textstyle\sum_{i=1}^n \xi_i \\ \text{subject to} \quad & \ell_i(\langle \mathbf{w}, \mathbf{x_i}\rangle + b) \geq 1 \ - \ \xi_i, \quad 1 \leq i \leq n \\ & \boldsymbol{\xi} \succeq 0. \end{aligned} \tag{1}$$

   In this question, we are rather interested in regression problem.
   Propose an adaptation of the SVM soft margin formulation for regression problems. You must precisely define the learning sample and all the terms of the optimisation problem.

## 1 Exercise: Hidden Markov Models (5 points)

We would like to create a tool able to produce some music pieces automatically. To simplify, we assume that this tool can only produce 5 sounds denoted by the following symbols: `a, b, c, d, e` and we build the tool with 3 electronic boxes able to produce the sounds. Each box can generate the sounds randomly but each box has its own properties. The first box can generate only the sounds `a` and `b`, the second one only `b` and `c`, and the last one `d` and `e`.

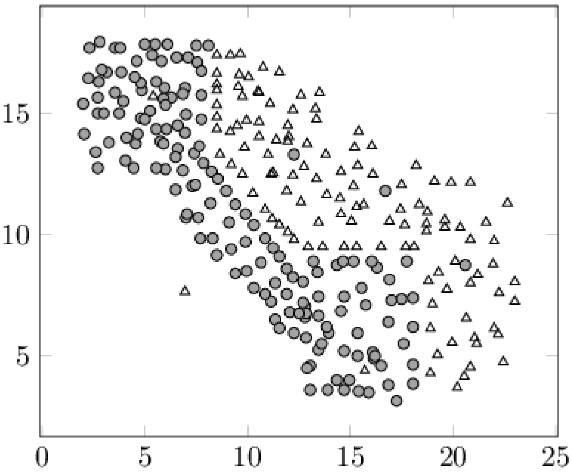
In this context, a sequence `abedac` represents a series of sounds produced randomly by the boxes. To generate a sequence, we begin by fixing its length, say $n$, and we generate $n$ sounds with the tool. To produce a sound, a first box is chosen randomly from a uniform distribution among the 3 boxes. Then, we choose randomly a sound among the possible ones in the considered box, again according to a uniform distribution over the sounds. Then, we move to the next box, all the boxes have the same probability to be chosen given a box.

1. Propose a way to model the tool by a HMM. Draw the HMM graphically (states, transitions and probabilities).
2. Compute the probability to generate the sequence `abe`. You must provide the full justification of your result.
3. If you sum the probabilities of all the sequences of symbols of **length 2**, what do you obtain?
4. Propose a new HMM ensuring that the first symbol is always the symbol `a`.

# 2 Exercise: Decision Trees (DT) (4 points)

Here is a dataset with points in $\mathbb{R}^2$ that we will name DtNn. Each point is of one of two classes (circle or triangle).
In the following exercise(s), you will sometimes need to replicate (copy) this dataset on your answer sheet. When you replicate the dataset, if it is simpler, you are free to:

- use simpler makers (instead of circle and triangle),
- use simplifications (less points, etc.) as long as you keep the important characteristics of the dataset.




Unless explicitly specified, no computations are expected. You are expected to give your best guess of the answer using the knowledge you have about decision trees.

1. Using DtNn, we learn a complete decision tree. Draw on DtNn the frontier between the classes, obtained from this tree.
2. Still using DtNn, what would be a good first split/decision (variable and threshold value) to use at the root of the tree (at depth 1)?
3. Still using DtNn, give the decision tree we obtain when we grow the tree to depth 2 and draw the corresponding decision boundaries on DtNn.
4. In general, should we use the "gini" or the "entropy" criterion to learn an optimal tree?
5. When you have a dataset and you want to do machine learning by exploiting an existing top-down tree learning algorithm, what choices do you have to make?

# 3 Exercise: Neural Networks (NN) (4 points)

1. We learn a neural network with no hidden layer and using a sigmoid/S-shaped activation function. On DtNn, draw the frontier between the classes, obtained from this NN.
2. Consider a setting like DtNn (2D inputs, 2 possible class outputs). We consider a multilayer perceptron with a single hidden layer containing two neurons ( using the activation function of your choice, and having bias(es)). Draw a schema of the network and list its parameters.
3. Give the expression of the output $y$ computed by this network for an input $x = (x_1, x_2)$.
4. Based on this network, explain the gradient descent algorithm: what is its goal, how is it applied for learning and what is its relation to the *chain-rule*. Ideally, give a pseudo-code for the algorithm and detail the computation for one iteration.

# 4 Exercise: Overfitting DT, NN and more (2 points)

1. Briefly explain what is overfitting.
2. For both DT and NN (and SVM if you have time), explain what makes the model susceptible to overfiting and what can or should be done to prevent it.