# STATISTICAL MACHINE LEARNING (WS2022)
# SEMINAR 9

**Assignment 1.** (*Gambler's ruin*) Consider a random walk on the set $L = \{0, 1, 2, \dots, a\}$ starting in some point $x \in L$. The position jumps by either $\pm 1$ in each time step (with equal probabilities). The walk ends if either of the boundary states $0, a$ is hit. Compute the probability $u(x)$ to finish in state $a$ if the process starts in state $x$.
*Hints:*

(1) What are the values of $u(0)$ and of $u(a)$?
(2) Find a difference equation for $u(x)$, $0 < x < a$ by relating it with $u(x-1)$ and $u(x+1)$.
(3) Translate the difference equation into a relation between the successive differences $u(x+1) - u(x)$ and $u(x) - u(x-1)$.
(4) Deduce that the solution is a linear function of $x$ and find its coefficients from the boundary conditions $u(0)$ and $u(a)$.

**Assignment 2.** Let us consider a Markov chain model for sequences $s = (s_1, \dots, s_n)$ of length $n$ with states $s_i \in K$ from a finite set $K$. Its joint probability distribution is given by

$$p(s) = p(s_1) \prod_{i=2}^{n} p(s_i \mid s_{i-1}).$$

The conditional probabilities $p(s_i \mid s_{i-1})$ and the marginal probability $p(s_1)$ for the first element are known. Let $A \subset K$ be a subset of states and let $\mathcal{A} = A^n$ denote the set of all sequences $s$ with $s_i \in A$ for all $i = 1, \dots, n$.

**a)** Find an efficient algorithm for computing the most probable sequence in $\mathcal{A}$.

**b)** Find an efficient algorithm for computing the probability $\mathbb{P}(\mathcal{A}) = \sum_{s \in \mathcal{A}} p(s)$.

**Assignment 3.** Let us consider the following matching problem. Given a sequence $x = (x_1, \dots, x_m)$ of points $x_i \in \mathbb{R}^2$ and another sequence $y = (y_1, \dots, y_n)$ of points in the same space, we want to find an optimal matching between them. (Notice that the sequences may have different length).

A matching $\tau$ is encoded as a path in the graph $(V, E)$, with nodes $V = \{(i, j) \mid i = 1, \dots, m, j = 1, \dots, n\}$ and edges connecting each node $(i, j)$ with nodes $(i+1, j)$, $(i, j+1)$ and $(i+1, j+1)$. The path should start in $(1, 1)$ and end in $(m, n)$. The cost of the matching $\tau$ is the sum of costs of the traversed nodes, where the cost for the node $(i, j)$ is the Euclidean distance $\|x_i - y_j\|$.

Explain how to find the optimal matching for a pair of point sequences $x$ and $y$ by dynamic programming. What is the run time complexity of your algorithm?

**Assignment 4.** Consider a hidden Markov model

$$p(x, s) = p(s_1) \prod_{i=2}^{n} p(s_i \mid s_{i-1}) \prod_{i=1}^{n} p(x_i \mid s_i),$$

where $x = (x_1, \ldots, x_n)$ is a sequence of features and $s = (s_1, \ldots, s_n)$ is a sequence of hidden states, with values $s_i$ from a finite set $K$. Given a sequence of features $x$ we want to predict the sequence of hidden states that has generated $x$.

**a)** The predictor should minimise the expected loss

$$R(x, h) = \sum_{s \in K^n} p(x, s) \ell(s, h(x)),$$

where $\ell(s, s')$ is the Hamming distance between sequences $s$ and $s'$, i.e.

$$\ell(s, s') = \sum_{i=1}^{n} [\![ s_i \neq s'_i ]\!].$$

Show that the optimal predictor for this loss is given by

$$s_i^* = \arg\max_{k \in K} p(s_i = k \mid x),$$

i.e. predicting the sequence of most probable states.

**b)** The predictor in a) requires to compute the marginal posterior probabilities $p(s_i = k \mid x)$ for all positions $i$ and all states $k \in K$. Show how to compute them for an HMM by performing dynamic matrix-vector multiplications from left to right and from right to left and combining the results.
*Hint:* Your algorithm will in fact compute the probabilities $p(s_i = k, x)$. The required normalisation for the posterior probabilities $p(s_i = k \mid x)$ can be postponed and done in the last step.