STATISTICAL MACHINE LEARNING (WS2018)
SEMINAR 6

Assignment 1. Consider the block-coordinate ascent for Minka's lower bound of the log-likelihood (in the EM algorithm)

$$L_B(\theta, \alpha, \mathcal{T}^m) = \frac{1}{m}\sum_{i=1}^{m}\sum_{y\in\mathcal{Y}} \alpha_i(y)\log p_\theta(x^i, y) - \frac{1}{m}\sum_{i=1}^{m}\sum_{y\in\mathcal{Y}} \alpha_i(y)\log \alpha_i(y).$$

The E-step requires to maximise it w.r.t. α-s for fixed θ.
a) Show that the maximisation decomposes into independent maximisation tasks of the type

$$\begin{aligned} &\sum_{y\in\mathcal{Y}} \alpha(y)\log p_\theta(x^i, y) - \sum_{y\in\mathcal{Y}} \alpha(y)\log\alpha(y) \to \max_{\alpha} \\ \text{s.t.} &\sum_{y\in\mathcal{Y}} \alpha(y) = 1 \;\text{ and }\; \alpha(y) \geqslant 0 \;\; \forall y \in \mathcal{Y} \end{aligned}$$

b) Prove that the function

$$g(x) = \begin{cases} x\log x & \text{if } x > 0, \\ 0 & \text{if } x = 0 \\ \infty & \text{otherwise} \end{cases}$$

is convex for $x > 0$ and implicitly accounts for the constraint $x \geqslant 0$. Conclude that the optimisation task in a) is convex.
c) Analyse the Lagrange dual of this task and deduce the solution $\alpha^*(y) = p_\theta(y \mid x^i)$.

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

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

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 $\boldsymbol{s}$ with $s_i \in A$ for all $i = 1, \ldots, n$. Find an efficient algorithm for computing the probability $p(\mathcal{A})$ of the event $\mathcal{A}$.

Assignment 3. Consider the same Markov model as in the previous assignment. You are given its most probable sequence $\boldsymbol{s}^* \in \arg\max_{\boldsymbol{s}\in K^n} p(\boldsymbol{s})$. The task is to find the most probable sequence $\boldsymbol{s}$ differing from $\boldsymbol{s}^*$ in all positions, i.e. $s_i \neq s_i^* \; \forall i = 1, \dots, n$. Give an algorithm for solving this task.

Assignment 4. (*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 ± 1 in each time period (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 5. Let $\mathcal{X}$ be a set of observations, $\mathcal{Y} = \{+1, -1\}$ a set of hidden states and $h\colon \mathcal{X} \to \mathcal{Y}$ a linear two-class classifier defined as

$$h(x; \boldsymbol{w}, b) = \operatorname{sign}(\langle \boldsymbol{w}, \boldsymbol{\phi}(x) \rangle + b) \tag{1}$$

where $\boldsymbol{w} \in \mathbb{R}^n$, $b \in \mathbb{R}$ are parameters and $\boldsymbol{\phi}\colon \mathcal{X} \to \mathbb{R}^n$ is a feature map. Show that (1) can be re-written in the following equivalent form

$$h(x; \boldsymbol{\theta}) = \operatorname*{arg\,max}_{y \in \mathcal{Y}} \langle \boldsymbol{\theta}, \boldsymbol{\phi}'(x, y) \rangle$$

where $\boldsymbol{\phi}'\colon \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^{n+1}$ and $\boldsymbol{\theta} \in \mathbb{R}^{n+1}$. Write the explicit form of $\boldsymbol{\phi}'(x, y)$ and $\boldsymbol{\theta}$.

Assignment 6. Consider a linear classifier $h\colon \mathcal{X} \to \mathcal{Y}$ assigning inputs $x \in \mathcal{X}$ to classes $\mathcal{Y} = \{1, \dots, Y\}$ based on the rule

$$h(x; \boldsymbol{w}_1, \dots, \boldsymbol{w}_Y, b_1, \dots, b_Y) = \operatorname*{arg\,max}_{y \in \mathcal{Y}} (\langle \boldsymbol{\phi}(x), \boldsymbol{w}_y \rangle + b_y) \tag{2}$$

where $\boldsymbol{\phi}\colon \mathcal{X} \to \mathbb{R}^n$ is a feature map and $(\boldsymbol{w}_y \in \mathbb{R}^n, b_y \in \mathbb{R})$, $y \in \mathcal{Y}$, are parameters.

a) Let $\mathcal{T}^m = \{(x^j, y^j) \in (\mathcal{X} \times \mathcal{Y}) \mid j = 1, \dots, m\}$ be a set of training examples. Describe a variant of the Perceptron algorithm which finds the parameters $(\boldsymbol{w}_y \in \mathbb{R}^n, b_y \in \mathbb{R})$, $y \in \mathcal{Y}$, such that the classifier (2) predicts all examples from $\mathcal{T}^m$ correctly provided such parameters exist.

b) Assume that you cannot evaluate the feature map $\boldsymbol{\phi}(x)$ explicitly, however, you can evaluate a kernel function $k\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such that $k(x, x') = \langle \boldsymbol{\phi}(x), \boldsymbol{\phi}(x') \rangle$, $\forall x, x' \in \mathcal{X}$. Show that you can still use the Perceptron algorithm to find a linear classifier with zero training error and that you can evaluate this classifier on any $x \in \mathcal{X}$.

Hint: Note that the parameter vectors $\boldsymbol{w}_y$, $y \in \mathcal{Y}$, can be in each iteration of the Perceptron algorithm expressed as a linear combination of the inputs $\boldsymbol{\phi}(x^j)$, $j \in \{1, \dots, m\}$.