STATISTICAL MACHINE LEARNING (WS2017)
SEMINAR 7

Assignment 1. Let $s_0, s_2, \ldots, s_{n-1}$ be K-valued random variables, where K is a finite set. Their joint probability distribution is a Markov model on a *cycle*

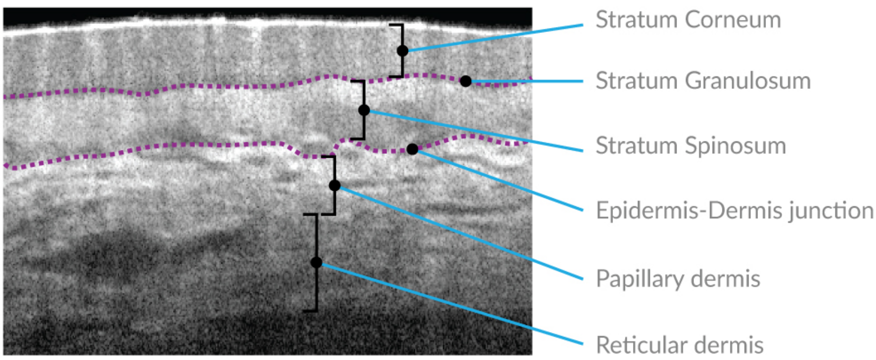
$$p(s) = \frac{1}{Z} \prod_{i=0}^{n-1} g_i(s_i, s_{i+1})$$

where indices $i+1$ are considered modulo n. The functions $g_i \colon K^2 \to \mathbb{R}_+$ are given and Z is a normalisation constant. Find an algorithm for searching the most probable realisation

$$s^* = \arg\max_{s \in K^n} p(s).$$

What complexity has it?

Assignment 2. Suppose your task is to automatically determine the thickness of the epidermis layer in OCT images (Optical Coherence Tomography) of skin . The epidermis is the topmost skin layer followed by the dermis. The boundary between them is called epidermis-dermis junction (see Figure). Propose an approach that combines a Deep Network with a Hidden Markov Model for sequences. Discuss how to learn the parameters of the respective model parts provided you are given annotated training data.



Assignment 3. Consider the class of $(\min, +)$-problems on graphs, which require to find the labelling

$$\boldsymbol{s}^* = \arg\min_{\boldsymbol{s} \in K^V} \sum_{i \in V} u_i(s_i) + \sum_{\{i,j\} \in E} u_{ij}(s_i, s_j),$$

where (V, E) is an undirected graph, K is a finite label set and $u_i\colon K \to \mathbb{R}$ and $u_{ij}\colon K^2 \to \mathbb{R}$ are given functions. Prove that this class is NP-hard by reducing the maximum clique problem to it.
Hint: Suppose that the graph (V', E') is an input instance for the maximum clique problem. Consider the graph (V, E) with $V = V'$, $E = \overline{E'}$ and the label set $K = \{0, 1\}$. Find functions u_i and u_{ij} such that a labelling $\boldsymbol{s}$ is optimal if and only if it "encodes" a maximum clique.

Assignment 4. Let $\mathcal{X} \subseteq \mathbb{R}^d$ be a set of input observations and $\mathcal{Y} = \{+1, -1\}$ a set of hidden states. A two-class linear classifier is defined as

$$h(\boldsymbol{x}; \boldsymbol{v}, b) = \begin{cases} +1 & \text{if} \quad \langle \boldsymbol{v}, \boldsymbol{x} \rangle + b \geq 0 \\ -1 & \text{if} \quad \langle \boldsymbol{v}, \boldsymbol{x} \rangle + b < 0 \end{cases}$$

where $(\boldsymbol{v}, b) \in \mathbb{R}^{d+1}$ denote its parameters. The parameters $(\boldsymbol{v}, b)$ can be learned from examples $\mathcal{T}^m = \{(\boldsymbol{x}^i, y^i) \in \mathcal{X} \times \mathcal{Y} \mid i = 1, \ldots, m\}$ by the SVM algorithm which minimizes the average of the hinge loss

$$F(\boldsymbol{v}, b) = \frac{1}{m} \sum_{i=1}^{m} \max\{0, 1 - y^i(\langle \boldsymbol{x}^i, \boldsymbol{v} \rangle + b)\} \,.$$

A generic linear classifier is defined as

$$h'(\boldsymbol{x}; \boldsymbol{w}) = \operatorname*{arg\,max}_{y \in \mathcal{Y}} \langle \boldsymbol{w}, \boldsymbol{\phi}(\boldsymbol{x}, y) \rangle \tag{1}$$

where $\boldsymbol{w} \in \mathbb{R}^n$ are parameters and $\boldsymbol{\phi}\colon \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^n$ is a joint feature map. The parameters $\boldsymbol{w}$ can be learned from examples $\mathcal{T}^m$ by the SO-SVM algorithm which minimizes the average of the margin re-scaling loss

$$F'(\boldsymbol{w}) = \frac{1}{m} \sum_{i=1}^{m} \max\{0, \max_{y \in \mathcal{Y} \setminus \{y^i\}} (\ell(y^i, y) + \langle \boldsymbol{w}, \boldsymbol{\phi}(\boldsymbol{x}^i, y) \rangle - \langle \boldsymbol{w}, \boldsymbol{\phi}(\boldsymbol{x}^i, y^i) \rangle)\} \tag{2}$$

where $\ell\colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ is some target loss depending on the application at hand.

Your task is to show that the standard SVM is a special case of the SO-SVM algorithm. To this end, define the joint feature map $\boldsymbol{\phi}$ and the target loss ℓ such that the two-class classifier $h(\boldsymbol{x}; \boldsymbol{v}, b)$ is equivalent to the generic linear classifier $h'(\boldsymbol{x}; \boldsymbol{w})$ and the objectives of the standard SVM and the SO-SVM are equivalent as well. In other words, you need to define $\boldsymbol{\phi}$ and ℓ such that

$$h(\boldsymbol{x}; \boldsymbol{v}, b) = h'(\boldsymbol{x}; (\boldsymbol{v}, b)),\ \forall \boldsymbol{x} \in \mathcal{X}\,, \quad \text{and} \quad F(\boldsymbol{v}, b) = F'((\boldsymbol{v}, b))\,, \forall \boldsymbol{v} \in \mathbb{R}^d, b \in \mathbb{R}\,,$$

where $(\boldsymbol{v}, b) \in \mathbb{R}^{d+1}$ denotes a vector obtained by concatenating $\boldsymbol{v}$ and b.

Assignment 5. Let $\mathcal{X} \subseteq \mathbb{R}^d$ be a set of input observations and $\mathcal{Y} = \{1, \ldots, Y\}$ a set of hidden states. The linear multi-class classifier is defined as

$$h(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{b}) = \operatorname*{arg\,max}_{y \in \mathcal{Y}} (\langle \boldsymbol{w}_y, \boldsymbol{x} \rangle + b_y) \tag{3}$$

where $\boldsymbol{W} = (\boldsymbol{w}_1, \ldots, \boldsymbol{w}_Y) \in \mathbb{R}^{d \times Y}$ is a matrix whose columns are the class templates and $\boldsymbol{b} = (b_1, \ldots, b_Y) \in \mathbb{R}^Y$ is a vector of the class biases.

a) Define the joint feature map $\boldsymbol{\phi}\colon \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^n$ and the corresponding joint parameter vector $\boldsymbol{w} \in \mathbb{R}^n$ composed of $\boldsymbol{W}$ and $\boldsymbol{b}$ such that the generic linear classifier (1) and the multi-class classifier (3) are equivalent, that is, $h'(\boldsymbol{x}; \boldsymbol{w}) = h(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{b})$, $\forall \boldsymbol{x} \in \mathcal{X}$.

b) Given a training set $\mathcal{T}^m = \{(\boldsymbol{x}^i, y^i) \in \mathcal{X} \times \mathcal{Y} \mid i = 1, \ldots, m\}$, the SO-SVM algorithm learns the parameters of the generic linear classifier (1) by solving a convex problem

$$\boldsymbol{w}^* = \operatorname*{arg\,min}_{\boldsymbol{w} \in \mathbb{R}^n} \left(\frac{\lambda}{2} \|\boldsymbol{w}\|^2 + F'(\boldsymbol{w}) \right) \tag{4}$$

where $\lambda > 0$ is a regularization constant and the empirical risk proxy $F'(\boldsymbol{w})$ is defined by (2). Use $\boldsymbol{\phi}$ derived in point **a)** to instantiate the problem (4) for the multi-class linear classifier (3) and the 0/1-loss $l(y, y') = [\![y \neq y']\!]$.

c) Rewrite the convex program from point **b)** as an equivalent quadratic programming task. What is the number of linear constraints of the quadratic program ?