STRUCTURED MODEL LEARNING (SML2019)
SEMINAR 4.

Assignment 1. Let $s\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a function defined as

$$s(x, x') = (\boldsymbol{\phi}(x) - \boldsymbol{\phi}(x'))^T \mathbf{W} (\boldsymbol{\phi}(x) - \boldsymbol{\phi}(x'))\,,$$

which measures a disimilarity between two images where $\boldsymbol{\phi}\colon \mathcal{X} \to \mathbb{R}^n$ is map extracting n features from image $x \in \mathcal{X}$, and $\boldsymbol{W} \in \mathbb{R}^{n\times n}$ is a matrix. Consider a classifier $h\colon \mathcal{X} \times \mathcal{X} \to \{-1, +1\}$ assigning a pair of images $(x, x') \in \mathcal{X} \times \mathcal{X}$ into the positive class if their disimilarity $s(x, x')$ is not higher than a threshold $b \in \mathbb{R}$ and to the negative class otherwise, i.e.

$$h(x, x'; \boldsymbol{W}, b) = \left\{ \begin{array}{lll} +1 & \text{if} & s(x, x') \leq b\,, \\ -1 & \text{if} & s(x, x') > b\,, \end{array} \right. \tag{1}$$

where $\boldsymbol{W} \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}$ are parameters of the classifier.

a) Let $\mathcal{T}^m = \{(x_A^j, x_b^j, y^j) \in (\mathcal{X} \times \mathcal{X} \times \{+1, -1\}) \mid j = 1, \ldots, m\}$ be a set of training examples composed of a pair of images (x_A, x_b) and their label y. Describe a variant of the Perceptron algorithm which finds the parameters $\boldsymbol{W} \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}$ such that the classifier (1) predicts all examples from $\mathcal{T}^m$ correctly provided such parameters exists.

b) Extend the algorithm from assignment a) such that the found matrix $\boldsymbol{W}$ is symmetric and positive definite, i.e. $\boldsymbol{W}^T = \boldsymbol{W}$ and $\langle \boldsymbol{u}, \boldsymbol{W}\boldsymbol{u} \rangle > 0$, $\forall \boldsymbol{u} \in \mathbb{R}^n$.

Assignment 2. Consider a linear classifier $h\colon \mathcal{X} \to \mathcal{Y}$ assigning inputs $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^n$ to classes $\mathcal{Y} = \{1, \ldots, Y\}$ based on the rule

$$h(\boldsymbol{x}; \boldsymbol{w}, b_1, \ldots, b_{Y-1}) = 1 + \sum_{y=1}^{Y-1} [\![\langle \boldsymbol{x}, \boldsymbol{w} \rangle \geq b_y]\!] \tag{2}$$

where $\boldsymbol{w} \in \mathbb{R}^n$ and $(b_1, b_2, \ldots, b_{Y-1}) \in \mathbb{R}^{Y-1}$ are parameters. Let $\mathcal{T}^m = \{(\boldsymbol{x}^j, y^j) \in (\mathcal{X} \times \mathcal{Y}) \mid j = 1, \ldots, m\}$ be a training set of examples. Describe a variant of the Perceptron algorithm which finds the parameters such that the classifier (2) predicts all examples from $\mathcal{T}^m$ correctly, provided such parameters exist, and makes sure that the found biases are ordered, i.e. $b_1 < b_2 < \cdots < b_{Y-1}$.

Assignment 3. Consider a linear max-sum classifier $h\colon \mathcal{X} \to \mathcal{Y}^n$ for a sequence prediction

$$\boldsymbol{y}^* = h(x; \boldsymbol{w}) = \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{Y}^n} \left(\sum_{i=1}^{n} \langle \boldsymbol{w}, \boldsymbol{\phi}_i^A(x, y_i) \rangle + \sum_{i=1}^{n-1} \langle \boldsymbol{w}, \boldsymbol{\phi}^B(y_i, y_{i+1}) \rangle \right) \tag{3}$$

where

- n is the length of the output sequence
- $\mathcal{X}$ is an arbitrary set of inputs
- $\mathcal{Y}$ is a finite set labels
- $\phi_i^A \colon \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$, $i = 1, \ldots, n$, and $\phi^B \colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^d$ are the fixed feature maps
- $\boldsymbol{w} \in \mathbb{R}^d$ are the weights

Let $\mathcal{T}^m = \{(x^j, y_1^j, \ldots, y_n^j) \in (\mathcal{X} \times \mathcal{Y}^n) \mid j = 1, \ldots, m\}$ be a set of training examples. Describe a variant of the Perceptron algorithm which finds the weights $\boldsymbol{w} \in \mathbb{R}^d$ such that the classifier (3) predicts all examples from $\mathcal{T}^m$ correctly provided such parameters exists. Describe two variants:

a) Perceptron using the dynamic programming to implement the classification oracle.

b) Perceptron which does not use the dynamic programming.

Assignment 4. Consider a linear classifier $h \colon \mathcal{X} \to \mathcal{Y}$ assigning inputs $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^n$ to classes $\mathcal{Y} = \{1, \ldots, Y\}$ based on the rule

$$h(\boldsymbol{x}; \boldsymbol{w}, b_1, \ldots, b_Y) = \operatorname*{argmax}_{y \in \mathcal{Y}} (y\langle \boldsymbol{x}, \boldsymbol{w}\rangle + b_y) \tag{4}$$

where $\boldsymbol{w} \in \mathbb{R}^n$ and $b_y \in \mathbb{R}$, $y \in \mathcal{Y}$, are parameters. Let $\mathcal{T}^m = \{(\boldsymbol{x}^j, y^j) \in (\mathcal{X} \times \mathcal{Y}) \mid j = 1, \ldots, m\}$ be a training set of examples. The goal is to learn parameters $\boldsymbol{w}$ such that the predictor (4) has a small expectation of the Mean Absolute Deviation loss $\ell(y, y') = |y - y'|$. To this end, we employ the SO-SVM framework learning parameters of a generic linear classifier

$$h(x) = \operatorname*{argmax}_{y \in \mathcal{Y}} \langle \boldsymbol{w}, \boldsymbol{\phi}(x, y)\rangle \tag{5}$$

by solving the convex problem $\boldsymbol{w}^* = \operatorname{argmin}_{\boldsymbol{w} \in \mathbb{R}^n} [\frac{\lambda}{2}\|\boldsymbol{w}\|^2 + R(\boldsymbol{w})]$ where $\lambda > 0$ is a regularization constant and

$$R(\boldsymbol{w}) = \frac{1}{m}\sum_{i=1}^{m} \max_{y \in \mathcal{Y}} \Big(\ell(y^i, y) + \langle \boldsymbol{w}, \boldsymbol{\phi}(x^i, y) - \boldsymbol{\phi}(x^i, y^i)\rangle\Big) .$$

a) Give an interpretation of the classification rule (4), i.e. for which type of prediction problems it is appropriate?

b) Define the joint feature map $\boldsymbol{\phi}(x, y)$ so that (5) and (4) are equivalent.

c) Write the risk $R(\boldsymbol{w})$ instantiated for the classification rule (4) and $\ell(y, y') = |y - y'|$. Write a formula for a sub-gradient of $R(\boldsymbol{w})$ at $\boldsymbol{w}$.

Assignment 5. Consider problem of learning a linear two-class SVM classifier

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

from a training set $\{(\boldsymbol{x}^1, y^1), \ldots, (\boldsymbol{x}^m, y^m)\} \in (\mathbb{R}^n \times \{+1,-1\})^m$ by solving

$$(\boldsymbol{w}^*, b^*) = \operatorname*{argmin}_{\boldsymbol{w}\in\mathbb{R}^n, b\in\mathbb{R}} \left[\frac{\lambda}{2}\|\boldsymbol{w}\|^2 + \frac{1}{m}\sum_{i=1}^m \xi_i\right] \tag{6a}$$

subject to

$$\begin{array}{rcll} y^i(\langle \boldsymbol{w}, \boldsymbol{x}^i\rangle + b) & \geq & 1 - \xi_i\,, & i \in \{1,\ldots,m\}\,, \\ \xi_i & \geq & 0\,, & i \in \{1,\ldots,m\}\,. \end{array} \tag{6b}$$

Note that the bias b is not contained in the quadratic regularizer. Convert the problem (6) to an unconstrained convex problem

$$\boldsymbol{w}^* = \operatorname*{argmin}_{\boldsymbol{w}\in\mathbb{R}^n} \left[\frac{\lambda}{2}\|\boldsymbol{w}\|^2 + R(\boldsymbol{w})\right]$$

and derive an algorithm for evaluating $R(\boldsymbol{w})$ and its sub-gradient.