STATISTICAL MACHINE LEARNING (WS2024)
SUPPORT VECTOR MACHINES

Assignment 1. Let $\mathcal{T}^m = \{(\boldsymbol{x}^i, y^i) \in \mathbb{R}^n \times \{-1, +1\} \mid i = 1, \ldots, m\}$ be a training set of linearly separable examples. In this case, the SVM algorithm finds parameters $(\boldsymbol{w}^*, b^*) \in \mathbb{R}^{n+1}$ of a linear classifier

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

by solving a convex quadratic program

$$(\boldsymbol{w}^*, b^*) \in \operatorname*{Arg\,min}_{(\boldsymbol{w}, b) \in \mathbb{R}^{n+1}} \frac{1}{2}\|\boldsymbol{w}\|^2 \quad \text{s.t.} \quad \begin{array}{rcll} \langle \boldsymbol{w}, \boldsymbol{x}^i \rangle + b & \geq & 1\,, & i \in \mathcal{I}_+ \\ \langle \boldsymbol{w}, \boldsymbol{x}^i \rangle + b & \leq & -1\,, & i \in \mathcal{I}_- \end{array} \tag{1}$$

where $\mathcal{I}_+ = \{i \in \{1, \ldots, m\} \mid y^i = +1\}$ and $\mathcal{I}_- = \{i \in \{1, \ldots, m\} \mid y^i = -1\}$ are indices of examples of the positive and the negative class, respectively.

a) Explain relation between the SVM for linearly separable examples, formulated in (1), and the generic SVM algorithm for learning from non-separable examples, which converts learning to

$$\left(\boldsymbol{w}^*, b^*, \boldsymbol{\xi}^*\right) = \operatorname*{arg\,min}_{\substack{(\boldsymbol{w}, b) \in \mathbb{R}^{n+1} \\ \boldsymbol{\xi} \in \mathbb{R}^m}} \left(\frac{1}{2}\|\boldsymbol{w}\|^2 + C \sum_{i=1}^{m} \xi_i \right)$$

subject to

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

b) For a given $(\boldsymbol{w}, b) \in \mathbb{R}^{n+1}$ such that $\boldsymbol{w} \neq 0$, we can define so called margin

$$d(\boldsymbol{w}, b) = \min_{i \in \{1, \ldots, m\}} \frac{y^i(\langle \boldsymbol{w}, \boldsymbol{x}^i \rangle + b)}{\|\boldsymbol{w}\|}$$

which is a signed distance between the hyperplane $\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b = 0$ and the closest examples in $\mathcal{T}^m$. Show that the hyperplane $\langle \boldsymbol{w}^*, x \rangle + b^* = 0$ found by SVM algorithm (1) maximizes the margin, i.e. it holds that

$$d(\boldsymbol{w}^*, b^*) = \max_{\substack{\boldsymbol{w} \in \mathbb{R}^n \setminus \{\mathbf{0}\} \\ b \in \mathbb{R}}} d(\boldsymbol{w}, b)\,. \tag{2}$$

c) How can you compute the value of the maximal margin $d(\boldsymbol{w}^*, b^*)$ from the solution $(\boldsymbol{w}^*, b^*)$?

Assignment 2. Let the input observation be a vector $\boldsymbol{x} \in \mathbb{R}^d$. Let us consider a feature map $\boldsymbol{\phi}_q \colon \mathbb{R}^d \to \mathbb{R}^n$, $n = d^q$, whose entries are all possible q-th degree ordered products of the entries of $\boldsymbol{x}$. For example, if $\boldsymbol{x} = (x_1, x_2, x_3)^T \in \mathbb{R}^3$ and $q = 2$ then

$$\boldsymbol{\phi}_q(\boldsymbol{x}) = \begin{pmatrix} x_1x_1 \\ x_2x_1 \\ x_3x_1 \\ x_1x_2 \\ x_2x_2 \\ x_3x_2 \\ x_1x_3 \\ x_2x_3 \\ x_3x_3 \end{pmatrix}$$

a) Show that for any $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^d$ we can compute the dot product between $\boldsymbol{\phi}_q(\boldsymbol{x})$ and $\boldsymbol{\phi}_q(\boldsymbol{x}')$ as

$$\langle \boldsymbol{\phi}_q(\boldsymbol{x}), \boldsymbol{\phi}_q(\boldsymbol{x}') \rangle = \langle \boldsymbol{x}, \boldsymbol{x}' \rangle^q \,,$$

that is, as the dot product of the original vectors $\boldsymbol{x}$ and $\boldsymbol{x}'$ powered to q.

Assignment 3. Consider a feature map $\boldsymbol{\phi} \colon \mathbb{R}^d \to \mathbb{R}^{d(d+1)/2}$ whose entries are

$$\begin{aligned} \boldsymbol{\phi}(\boldsymbol{x}) \;=\; (\;\; & x_1^2, \;\; \sqrt{2}x_1x_2, \;\; \sqrt{2}x_1x_3, \;\; \ldots, \;\; \sqrt{2}x_1x_d, \\ & x_2^2, \;\;\;\; \sqrt{2}x_2x_3, \;\; \ldots, \;\; \sqrt{2}x_2x_d, \\ & \qquad\qquad\qquad\quad \vdots \\ & \qquad\qquad\qquad\quad x_d^2 \qquad)^T \,, \end{aligned}$$

so that the features correspond to all possible products of unordered pairs of entries from $\boldsymbol{x}$, and the products of different entries are multiplied by a constant factor $\sqrt{2}$. For example, if $\boldsymbol{x} = (x_1, x_2, x_3)^T \in \mathbb{R}^3$ then

$$\boldsymbol{\phi}(x) = (x_1^2, \sqrt{2}x_1x_2, \sqrt{2}x_1x_3, x_2^2, \sqrt{2}x_2x_3, x_3^2)^T \,.$$

This feature map defines a kernel $k(\boldsymbol{x}, \boldsymbol{x}') = \langle \boldsymbol{\phi}'(\boldsymbol{x}), \boldsymbol{\phi}'(\boldsymbol{x}') \rangle$ referred to as the homogeneous polynomial kernel of degree 2. Show that the kernel value equals to the square of the dot product of the input vectors, that is prove the identity

$$k(\boldsymbol{x}, \boldsymbol{x}') = \langle \boldsymbol{\phi}'(\boldsymbol{x}), \boldsymbol{\phi}'(\boldsymbol{x}') \rangle = \langle \boldsymbol{x}, \boldsymbol{x}' \rangle^2 \,, \qquad \forall \boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^d \,.$$

Assignment 4. Assume we are given a training set of examples $\mathcal{T}^m = \{(x^i, y^i) \in (\mathcal{X} \times \{+1, -1\}) \mid i = 1, \ldots, m\}$ which is known to be linearly separable with respect to a feature map $\boldsymbol{\phi} \colon \mathcal{X} \to \mathbb{R}^n$. In this case, we can find parameters $(\boldsymbol{w}, b) \in \mathbb{R}^{n+1}$ of a linear classifier $h(x; \boldsymbol{w}, b) = \operatorname{sign}(\langle \boldsymbol{\phi}(x), \boldsymbol{w} \rangle + b)$ which has zero training error by the Perceptron algorithm:

(1) $\boldsymbol{w} \leftarrow 0$, $b \leftarrow 0$
(2) Find an example $(x^u, y^u) \in \mathcal{T}^m$ whose label is incorrectly predicted by the current classifier, that is $h(x^u; \boldsymbol{w}, b) \neq y^u$.

(3) If all examples are classified correctly exit the algorithm. Otherwise update the parameters by

$$\boldsymbol{w} \leftarrow \boldsymbol{w} + y^u \boldsymbol{\phi}(x^u) \qquad \text{and} \qquad b \leftarrow b + y^u$$

and go to Step 2.

Assume that you cannot evaluate the feature map $\boldsymbol{\phi}(x)$ because it is either unknown or its evaluation is expensive. However, you know how to cheaply 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 $x \in \mathcal{X}$.