# Exercise

## 08

TUM Department of Informatics

**Supervised by** Prof. Dr. Stephan Günnemann
Informatics 3 - Professorship of Data Mining and Analytics

**Submitted by** Marcel Bruckner (03674122)
Julian Hohenadel (03673879)
Kevin Bein (03707775)

**Submission date** Munich, December 8, 2019

## SVM and Kernels

**Problem 1:**

Similarities:

Both try to find a fitting hyperplane which seperates the data classes.

Difference:

SVM tries to maximize the margin from the hyperplane to the data points, perceptron algorithms only care about a valid seperation of the data classes.

**Problem 2:**

a)

$g(\alpha)$ vectorized definition:

$$g(\alpha) = \frac{1}{2}\alpha^T Q\alpha + \alpha^T 1_N$$

$g(\alpha)$ standard definition:

$$g(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} y_i y_j \alpha_i \alpha_j x_i^T x_j$$

$y$ is a vector of dimension $N \times 1$

$x$ is a matrix of dimension $N \times M$

$\sum_{i=1}^{N}\sum_{j=1}^{N} y_i y_j$ is equivalent to $yy^T$ (dimension is $N \times N$)

$\sum_{i=1}^{N}\sum_{j=1}^{N} x_i^T x_j$ is equivalent to $XX^T$ (dimension is $N \times N$)

$\sum_{i=1}^{N}\sum_{j=1}^{N} y_i y_j x_i^T x_j$ is the Hadamard product so: $[yy^T \odot XX^T]$

Take the $-1$ scalar from the standard definition into the matrix: $[-yy^T \odot XX^T] = Q$

$\implies \frac{1}{2}\alpha^T Q\alpha \equiv \frac{1}{2}\alpha^T[-yy^T \odot XX^T]\alpha \equiv -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} y_i y_j \alpha_i \alpha_j x_i^T x_j$

$\alpha^T 1_N \equiv \sum_{i=1}^{N} \alpha_i$ is trivial.

$\implies g(\alpha)$ vectorized definition $\equiv g(\alpha)$ standard definition

b)

If the search for a local maximizer of $g$ returns the global maximum of $g$, that means that the maximization problem is concave.

To proove this claim $Q$ needs to be negativ (semi)definite (NSD).

For $Q$ to be NSD: $\forall \alpha \in \mathbb{R}^N : \alpha^T Q\alpha \leq 0$ needs to hold.

$$\alpha^T Q \alpha \leq 0 \tag{1}$$

$$-\sum_{i=1}^{N}\sum_{j=1}^{N} y_i y_j \alpha_i \alpha_j x_i^T x_j \leq 0 \tag{2}$$

$$-\sum_{i=1}^{N}\sum_{j=1}^{N} (y_i \alpha_i x_i)^T (y_j \alpha_j x_j) \leq 0 \tag{3}$$

$$-(y \odot \alpha)^T X X^T (y \odot \alpha) \leq 0 \tag{4}$$

$$-(y \odot \alpha)^T (X^T)^T X^T (y \odot \alpha) \leq 0 \tag{5}$$

$$-(X^T (y \odot \alpha))^T (X^T (y \odot \alpha)) \leq 0 \tag{6}$$

$$-(X^T (y \odot \alpha))^2 \leq 0 \tag{7}$$

$$-(\geq 0) \leq 0 \qquad \square \tag{8}$$

(3): $y_i$ and $\alpha_i$ can be dragged inside here because they are scalars.

(4): The whole expression returns a scalar, that's why reshaping is done this way:

$dim((y \odot \alpha)^T) = 1 \times N$

$dim(X) = N \times 1$

$dim(X^T) = 1 \times N$

$dim((y \odot \alpha)) = N \times 1$

(5): $X = (X^T)^T$

(6): $(AB)^T = B^T A^T$

(7): $(...)^2 \geq 0$, (as long as ... is not complex)

(8): Proofs that $Q$ is NSD.

$Q$ is NSD, that means the maximization problem is in fact concave so local maxima = global maxima.

**Problem 3:**

$\epsilon$ is the LOOCV misclassification rate.

$s$ is the number of support vectors.

$N$ is the number of samples.

$$\epsilon \leq \frac{s}{N}$$

Case 1:

$x_i$ from the current LOOCV is a support vector:

$x_i$ is misclassified: $\implies$ $\alpha_i$ of $x_i$ is $0$ $\implies$ the misclassification of $x_i$ will affect $w^*$ and $b^*$.

$$\implies \epsilon_{\text{Case 1}} \leq \frac{s}{N}$$

Case 2:

$x_i$ from the current LOOCV isn't a support vector:

Then $x_i$ will have no effect on $w^*$ and $b^*$ anyway.

$\implies$ nothing will change.

$$\implies \epsilon_{\text{Case 2}} = 0$$

Both cases combined evaluate to the original inequality.

$$\epsilon_{\text{Case 1 and 2}} \leq \frac{s}{N} + 0$$
$$\epsilon \leq \frac{s}{N}$$

**Problem 5:**

"A kernel is valid if it corresponds to an inner product in some feature space."

$x_1^T x_2$ is a valid kernel because it is a scalar product of the input vectors.

$a_0$ is a constant term and a valid kernel because it can be generated by:

$k(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ with $\phi(x) = \sqrt{a_0} \implies k(x_i, x_j) = a_0$

The same holds for $a_i$ so $a_i(x_i^T x_j)^i + a_0$ can be represented as kernels.

Sums and multiplications of kernels are kernel preserving (also scalars are $\geq 0$)

$\implies$ Gram matrix is still PSD.

**Problem 6:**

The "trick" for this Problem:

"The Maclaurin series for $\frac{1}{1-x}$ is the geometric series: $1 + x + x^2 + x^3 + x^4$..."

$$\begin{aligned} k(x_1, x_2) &= \frac{1}{1 - x_1 x_2} \\ &= \sum_{i=0}^{\infty} x_1^i x_2^i \\ &= \phi(x_1)^T \phi(x_2) \\ \text{with } \phi(x) &= (1, x, x^2, x^3, x^4, ...) \end{aligned}$$

**Problem 4:**

# Appendix

We confirm that the submitted solution is original work and was written by us without further assistance. Appropriate credit has been given where reference has been made to the work of others.

Munich, December 8, 2019, Signature Marcel Bruckner (03674122)

Munich, December 8, 2019, Signature Julian Hohenadel (03673879)

Munich, December 8, 2019, Signature Kevin Bein (03707775)