Machine Learning Exercise Sheet 09

SVM and Kernels

In-Class

1 SVM

Problem 1: What is the connection between soft-margin SVM and logistic regression?

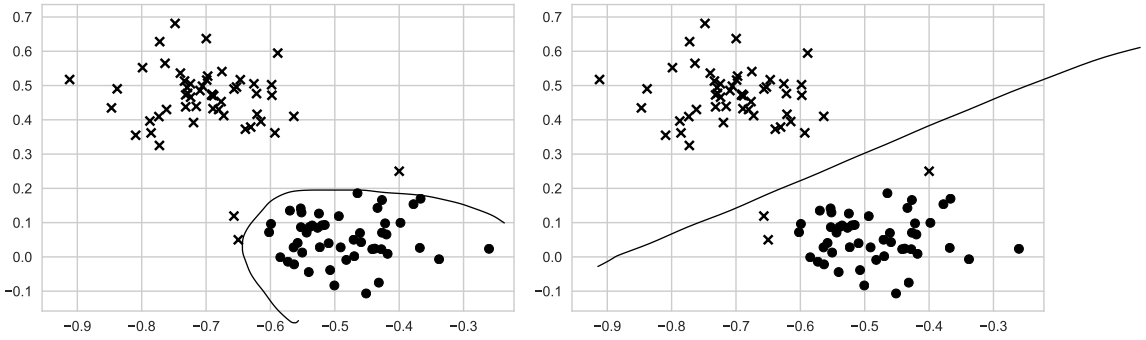
Problem 2: Consider a soft-margin SVM fitted to a linearly separable dataset $\mathcal{D}$ using the Hinge loss formulation of the optimization task.

$$\text{minimize}_{\boldsymbol{w},b} \quad \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_{i=1}^{N}\max\left(0, 1 - y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b)\right)$$

a) Is it guaranteed that all training samples in $\mathcal{D}$ will be assigned the correct label by the model?

b) Prove that if for some $C_0 \geq 0$ the resulting model classifies all training samples correctly and all samples lie outside of the margin then it will also be the case if we train the model with any larger $C > C_0$.

Problem 3: Sketch the decision boundary of an SVM with a quadratic kernel (polynomial with degree 2) for the data in the figure below, for two specified values of the penalty parameter C. The two classes are denoted as •'s and ×'s.



(a) $C = 10^{10}$ (b) $C = 10^{-10}$

P1 soft-svm

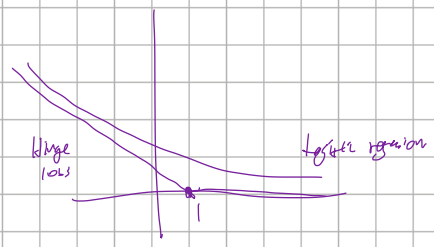
minimize $f_0(w,b,\xi) = \frac{1}{2}w^Tw + C\sum_{i=1}^{N}\xi_i$

subject to $y_i(w^Tx_i+b) - 1 + \xi_i \geq 0$

$\xi_i \geq 0$.

complementary slackness condition

$\alpha_i(y_i(w^Tx_i+b) - 1 + \xi_i) = 0$. $\Rightarrow$ correct classify $y_i(w^Tx_i+b) \geq 1$



$E = \frac{1}{2}w^Tw + C\sum_{i=1}^{N} \text{hinge}(y_i(w_i^Tx_i+b))$

logistic regression

$p(y=1 \mid x_i,w) = \sigma(z) = \sigma(y_i(w^Tx_i+b)) = \sigma(w^Tx_i+b)$

$p(y=-1 \mid x_i,w) = \sigma(z) = \sigma(y_i(w^Tx_i+b)) = \sigma(-(w^Tx_i+b))$

$-\ln p(y \mid x_i,w) = -\prod_{i=1}^{N} \ln p(y_i \mid x_i,w)$

$= -\sum_{i=1}^{N} \ln \sigma(y_i(w^Tx_i+b)) = +\sum_{i=1}^{N} \ln(1+\exp(-y_i(w^Tx_i+b)))^{-1}$

$E = \frac{1}{2}w^T \cdot w + \sum \ln(1+\exp(-y_i(w^Tx_i+b)))$

P2 Nope because of soft-svm, it will allow the some training sample violate the boundary

② $C_0 \geq 0$ all training set are correctly classified.

$h(w,b) = 0$ if all sample are correctly classify

$E(w,b) = \frac{1}{2}w^Tw + C_0h(w,b)$

$E(v,d) = \frac{1}{2}v^Tv + Ch(v,d)$

$\frac{1}{2}w^{*T}w^* + C_0h(w^*,b^*) \leq \frac{1}{2}v^Tv^* + C_0h(v^*,d^*)$

$\frac{1}{2}v^{*T}v^* + Ch(v^*,d^*) \leq \frac{1}{2}w^{*T}w^* + Ch(w^*,b^*)$

$C_0h(w^*,b^*) + Ch(v^*,d^*) \leq C_0h(v^*,d^*) + Ch(w^*,b^*)$

$(C_0-C)(h(w^*,b^*) - h(v^*,d^*)) \leq 0$ $\quad h(v^*,d^*) \leq h(w^*,b^*)$

≤ 0 $\quad \geq 0$

hinge loss decrease for any $C \geq C_0$

2 Kernels

Problem 4: Consider the Gaussian kernel

$$k_{\mathrm{G}}(\boldsymbol{x}_1, \boldsymbol{x}_2) = \exp\left(-\frac{\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|^2}{2\sigma^2}\right), \text{ with } \boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{R}^d .$$

a) Suppose you have found a feature map $\theta : \mathbb{R}^d \to \mathbb{R}^{d_1}$ that transforms your data into a feature space in which a SVM with a Gaussian kernel works well. However, computing $\theta(\boldsymbol{x})$ is computationally expensive and luckily you discover an efficient method to compute the scalar product

$$k(\boldsymbol{x}_1, \boldsymbol{x}_2) = \theta(\boldsymbol{x}_1)^T \theta(\boldsymbol{x}_2)$$

in your feature space without having to compute $\theta(\boldsymbol{x}_1)$ and $\theta(\boldsymbol{x}_2)$ explicitly. Show how you can use the scalar product $k(\boldsymbol{x}_1, \boldsymbol{x}_2)$ to efficiently compute the Gaussian kernel $k_G(\theta(\boldsymbol{x}_1), \theta(\boldsymbol{x}_2))$ in your feature space.

b) One of the nice things about kernels is that new kernels can be constructed out of already given ones. Use the five kernel construction rules from the lecture to prove that k_G is a kernel.

Hint: Use the Taylor expansion of the exponential function to prove that $\exp \circ k_1$ *is a kernel if* k_1 *is a kernel. Also, consider* $k_2(\phi(\boldsymbol{x}_1), \phi(\boldsymbol{x}_2))$ *with the linear kernel* $k_2(\boldsymbol{x}_1, \boldsymbol{x}_2) = \boldsymbol{x}_1^T \boldsymbol{x}_2$ *and a feature map* ϕ *with only one feature.*

c) Can any finite set of points be linearly separated in the feature space of the Gaussian kernel if σ can be chosen freely?

Problem 5: Let $\mathcal{M} = \cup_{n\in\mathbb{N}} \cup_{m\in\mathbb{N}} \mathbb{R}^{n\times m}$ denote the set of all real-valued matrices of arbitrary size. Prove that the function $k : \mathcal{M} \times \mathcal{M} \to \mathbb{R}$ is a valid kernel, where

$$k(\boldsymbol{X}, \boldsymbol{Y}) = \min(\mathrm{rank}(\boldsymbol{X}), \mathrm{rank}(\boldsymbol{Y})) .$$

P4 $k_G(\theta(x_1), \theta(x_2)) = \exp\left(-\frac{x_1^T x_1 - 2x_1^T x_2 + x_2^T x_2}{2\sigma^2}\right)$

$= \exp\left(-\frac{k(x_1,x_1) - 2k(x_1,x_2) + k(x_2,x_2)}{2\sigma^2}\right)$

② $\exp(k(x_1,x_2)) = 1 + \sum_{n=1}^{\infty} \frac{k(x_1,x_2)^n}{n!}$

rule 3 $k(x_1,x_2)^n = k(x_1,x_2) \cdot k(x_1,x_2) \cdots$ valid.

rule 2 $\frac{1}{n!} k(x_1,x_2)^n$ $\frac{1}{n!} > 0$ valid

rule 1 $\sum_{n=1}^{N} \frac{1}{n!} k(x_1,x_2)^n$ valid

rule 1 $1 + \sum_{n=1}^{N} \frac{1}{n!} k(x_1,x_2)^n$ valid.

$\exp\left(-\frac{x_1^T x_1}{2\sigma^2}\right) \exp\left(-\frac{x_2^T x_2}{2\sigma^2}\right) \exp\left(\frac{x_1^T x_2}{\sigma^2}\right)$

↑ $x_1^T y_2$ rule 3

$\frac{1}{\sigma^2} x_1^T x_2$ rule 2.

rule 3 $x_1^T x_1$ valid.

feature map $\phi(x) = \frac{x^T x}{2\sigma^2}$.

Homework

3 SVM

Problem 6: Explain the similarities and differences between the SVM and perceptron algorithms. How do they perform classification? In what way do they differ?

Problem 7: Recall that the dual function in the setting of the SVM training task (Slide 17) can be written as

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

(a) Write down the matrix $\boldsymbol{Q}$ using the vector of labels $\boldsymbol{y}$ and feature matrix $\boldsymbol{X}$. Denote the element-wise product between two matrices (in case you want to use it) by $\odot$ (also known as Hadamard product or Schur product).

(b) Prove that we can search for a *local* maximizer of g to find its *global* maximum (don't forget to prove properties of $\boldsymbol{Q}$ that you decide to use in this task).

Problem 8: Consider training a standard hard-margin SVM on a linearly separable training set of N samples. Let s denote the number of support vectors we would obtain if we would train on the entire dataset. Furthermore, let ε denote the leave-one-out cross validation (LOOCV) misclassification rate. Prove that the following relation holds:

$$\varepsilon \leq \frac{s}{N} \,.$$

Problem 9: Load the notebook `exercise_09_notebook.ipynb` from Moodle. Fill in the missing code and run the notebook. Convert the evaluated notebook to pdf and add it to the printout of your homework.

Note: We suggest that you use Anaconda for installing Python and Jupyter, as well as for managing packages. We recommend that you use Python 3.

For more information on Jupyter notebooks and how to convert them to other formats, consult the Jupyter documentation and nbconvert documentation.

4 Kernels

Problem 10: Show that for $N \in \mathbb{N}$ and $a_i \geq 0$ for $i = 0, \ldots, N$ the following function k is a valid kernel.

$$k(\boldsymbol{x}_1, \boldsymbol{x}_2) = \sum_{i=1}^{N} a_i \left(\boldsymbol{x}_1^T \boldsymbol{x}_2\right)^i + a_0, \text{ with } \boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{R}^d \,.$$

Problem 11: Find the feature transformation $\boldsymbol{\phi}(x)$ corresponding to the kernel

$$k(x_1, x_2) = \frac{1}{1 - x_1 x_2}, \text{ with } x_1, x_2 \in (0, 1) \,.$$

Hint: Consider an infinite-dimensional feature space and infinite series.

P7

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

$$Q = -y_i y_j x_i^T x_j$$

$y \in R^{n \times 1}$ $\quad = -yy^T \odot XX^T$

$x \in R^{n \times 1}$

$(n \times n) \times (n \times n)$

convex. $(Q \rightarrow PSD)$

$$\alpha^* \cdot \underbrace{y_i y_j x_i^T x_j}_{PSD} \alpha = \left(\sum_{i=1}^{N} \alpha_i y_i x_i\right)^T \left(\sum_{j=1}^{N} \alpha_j y_j x_j\right)$$

$$= \left\| \sum_{i=1}^{N} \alpha_i y_i x_i \right\|^2 \geq 0.$$

$Q \rightarrow PSD$ concave $\Leftrightarrow$ maximum

local maximal $=$ global maximum

P8 Because non-support vector won't update the w^* and b^*

only support vector are critical for linear boundary

so $q < \frac{s}{N}$

P10. $q = x_1 x_2$

$$\frac{1 - x_1 x_2^n}{1 - x_1 x_2} \qquad x_1 \cdot x_2 \in (0,1)$$

$n \rightarrow \infty$

$$\frac{1}{1 - x_1 x_2} = \sum x_1^i x_2^i < \phi^T(x_1)\phi(x_2)$$