University of Stuttgart | ANALYTIC COMPUTING

Machine Learning (SS 23)

Assignment 09: Support Vector Machine (Solution)

Mojtaba Nayyeri
Mojtaba.Nayyeri@ipvs.uni-stuttgart.de

Akram Sadat Hosseini
Akram.Hosseini@ipvs.uni-stuttgart.de

Nadeen Fathallah
Nadeen.Fathallah@ipvs.uni-stuttgart.de

Rodrigo Lopez Portillo Alcocer
rodrigo.lopez-portillo-alcocer@ipvs.uni-stuttgart.de

Tim Schneider
timphillip.schneider@ipvs.uni-stuttgart.de

Osama Mohammed
osama.mohammed@ipvs.uni-stuttgart.de

Daniel Frank
daniel.frank@ipvs.uni-stuttgart.de

This assignment sheet consists of 6 pages with 4 Questions:

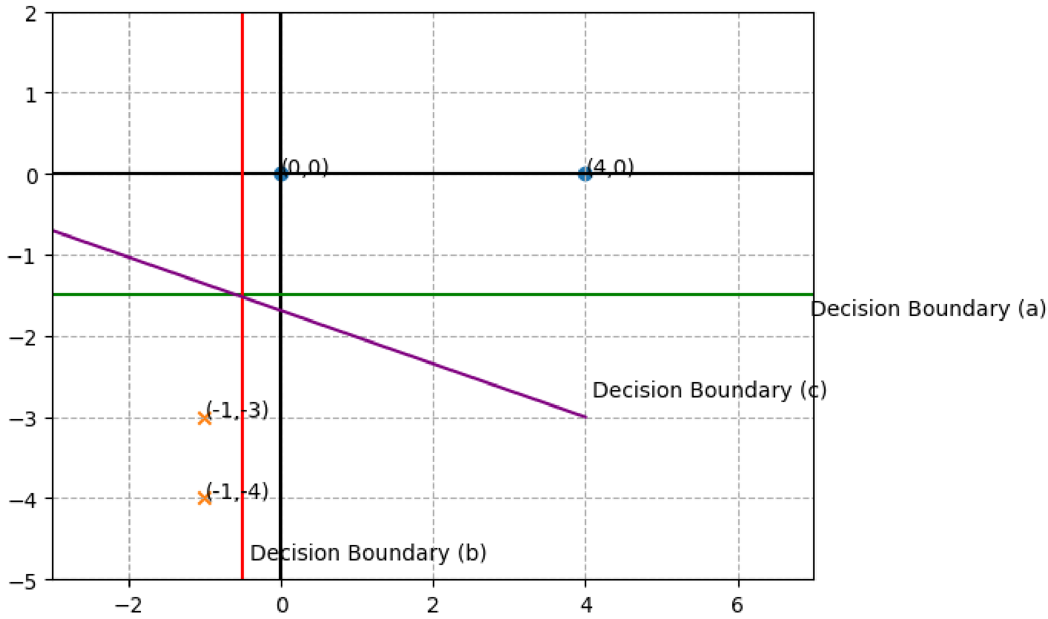
Submit your solution in ILIAS as a single PDF file.[1] Make sure to list full names of all participants, matriculation number, study program and B.Sc. or M.Sc on the first page. Optionally, you can *additionally* upload source files (e.g. PPTX files). If you have any questions, feel free to ask them in the exercise forum in ILIAS.

Submission is open until Monday, 3rd July 2023, 12:00 noon.

[1] Your drawing software probably allows to export as PDF. An alternative option is to use a PDF printer. If you create multiple PDF files, use a merging tool (like pdfarranger) to combine the PDFs into a single file.

Question 1: Support Vector Machine

Consider the dataset shown in the figure below where a linear Support Vector Machine (SVM) without slack variables is supposed to be used:



1. Which of the decision boundaries (a), (b) or (c) shown on the figure would be the resulting decision boundary of linear SVM? Show your calculations. When answering this question, no need to solve by optimizing the SVM objective function.
2. What are the support vectors based on your answer in (1.)?
3. How would adding a training point in location (1, 1) to the dataset that belongs to the o class change the decision boundary?

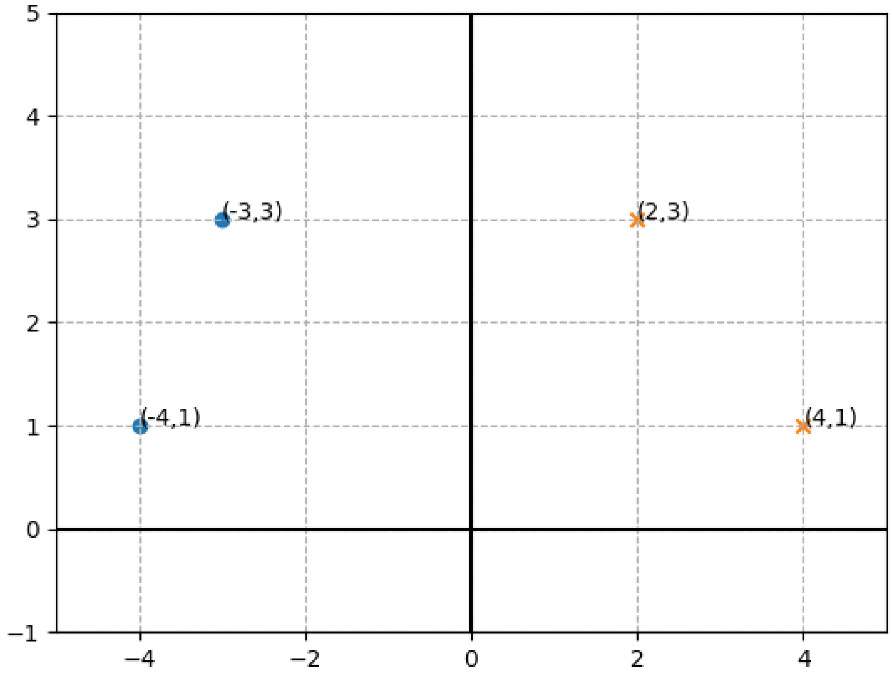
Solution:

1. For (a), the margin is 1.5 (half the distance between the lines passing through the nearest points from both classes).
   For (b), the margin is 0.5.
   For (c), the margin is 1.5811.
   Since the SVM is a maximum margin classifier, then the decision boundary is (c).
2. (0, 0) from Class O and (-1, -3) from Class X.
3. It will have no impact since this point is outside the margin beyond the support vectors and the decision boundary of SVM is solely based on the location of the support vectors.

Question 2: Perceptron

For the data given below, apply the perceptron algorithm (slide 10) to find the weight vector w of the decision boundary. Show the output of each iteration till convergence. Assume that the weight vector is initialized $w^{(0)} = \begin{bmatrix} 1 \\ 3 \end{bmatrix}$ with $\alpha = 0.1$ (there is no bias).

For class x, $t = 1$
For class o, $t = -1$



Solution:
Given the initial weight vector $w^{(0)}$, the four point are classified as follows:

Point	Actual Target Value	Predicted Target Value	Classification Result
x_1: (-4,1)	t_1:-1	$({w^{(0)}}^T x_1 = -1 < 0) = -1$	Classified Correctly
x_2: (-3,3)	t_2:-1	$({w^{(0)}}^T x_2 = 6 > 0) = 1$	Misclassified
x_3: (2,3)	t_3:1	$({w^{(0)}}^T x_3 = 11 > 0) = 1$	Classified Correctly
x_4: (4,1)	t_4:1	$({w^{(0)}}^T x_4 = 7 > 0) = 1$	Classified Correctly

Based on the above results, x_2 is misclassified. Therefore, we will update the weight vector using x_2.

Iteration 1: $w^{(1)} = w^{(0)} + 0.1 x_2 t_2$

$$w^{(1)} = \begin{bmatrix} 1 \\ 3 \end{bmatrix} + (0.1) * \begin{bmatrix} -3 \\ 3 \end{bmatrix} * (-1) = \begin{bmatrix} 1.3 \\ 2.7 \end{bmatrix}$$

Given that ${w^{(1)}}^T x_2 = 4.2 > 0$, the predicted target value of x_2 is 1. Therefore, it is still misclassified.

Iteration 2: $w^{(2)} = w^{(1)} + 0.1 x_2 t_2$

$$w^{(2)} = \begin{bmatrix} 1.3 \\ 2.7 \end{bmatrix} + (0.1) * \begin{bmatrix} -3 \\ 3 \end{bmatrix} * (-1) = \begin{bmatrix} 1.6 \\ 2.4 \end{bmatrix}$$

Given that ${w^{(2)}}^T x_2 = 2.4 > 0$, the predicted target value of x_2 is 1. Therefore, it is still misclassified.

Iteration 3: $w^{(3)} = w^{(2)} + 0.1 x_2 t_2$

$$w^{(3)} = \begin{bmatrix} 1.6 \\ 2.4 \end{bmatrix} + (0.1) * \begin{bmatrix} -3 \\ 3 \end{bmatrix} * (-1) = \begin{bmatrix} 1.9 \\ 2.1 \end{bmatrix}$$

Given that ${w^{(3)}}^T x_2 = 0.6 > 0$, the predicted target value of x_2 is 1. Therefore, it is still misclassified.

Iteration 4: $w^{(4)} = w^{(3)} + 0.1 x_2 t_2$

$$w^{(4)} = \begin{bmatrix} 1.9 \\ 2.1 \end{bmatrix} + (0.1) * \begin{bmatrix} -3 \\ 3 \end{bmatrix} * (-1) = \begin{bmatrix} 2.2 \\ 1.8 \end{bmatrix}$$

Given that ${w^{(4)}}^T x_2 = -1.2 < 0$, the predicted target value of x_2 is -1. Therefore, it is still classified correctly. Finally, we will check the classification result of all the points using the new weight vector $w^{(4)}$.

Point	Actual Target Value	Predicted Target Value	Classification Result
x_1: (-4,1)	t_1:-1	$({w^{(4)}}^T x_1 = \text{-7} < 0) = \text{-1}$	Classified Correctly
x_2: (-3,3)	t_2:-1	$({w^{(4)}}^T x_2 = \text{-1.2} < 0) = \text{-1}$	Classified Correctly
x_3: (2,3)	t_3:1	$({w^{(4)}}^T x_3 = 9.8 > 0) = 1$	Classified Correctly
x_4: (4,1)	t_4:1	$({w^{(4)}}^T x_4 = 10.6 > 0) = 1$	Classified Correctly

Based on the above results, all the points are classified correctly. Therefore, the perceptron algorithm stops.

Question 3: Polynomial Kernel

The second-order polynomial kernel for a two-dimensional vector $x_i = \begin{bmatrix} x_{i1} & x_{i2} \end{bmatrix}^\top$ is defined as:

$$\phi(x_i) = \begin{bmatrix} x_{i1}^2 \\ \sqrt{2}x_{i1}x_{i2} \\ x_{i2}^2 \end{bmatrix}$$

Show that the mapping of the two-dimensional vector to three dimensions is not necessary for calculating the scalar product $\langle \phi(x_i), \phi(x_j) \rangle$. (Note: Transform the equation such that it only uses the scalar product of two-dimensional vectors.)

Solution:

$$\phi(x_i)^T \phi(x_j) = \begin{bmatrix} x_{i1}^2 \\ \sqrt{2}x_{i1}x_{i2} \\ x_{i2}^2 \end{bmatrix}^T \begin{bmatrix} x_{j1}^2 \\ \sqrt{2}x_{j1}x_{j2} \\ x_{j2}^2 \end{bmatrix} = x_{i1}^2 x_{j1}^2 + 2x_{i1}x_{i2}x_{j1}x_{j2} + x_{i2}^2 x_{j2}^2 = (x_{i1}x_{j1} + x_{i2}x_{j2})^2 = (x_i^T x_j)^2$$

As you can see, we can directly use the two data points without mapping them to three dimensions beforehand.

Question 4: Gaussian Kernel

For all students other than B.Sc. Data Science.

Slide 69 mentions that the Gaussian kernel, also called Radial Basis Function (RBF), projects to an infinite dimensional feature space. Give an intuition on why this is the case and prove it. (Note: Use the Taylor expansion over e^x to show that the Gaussian kernel is an infinite sum over the polynomial kernels.)

Solution:

Definition of Gaussian kernel: $k(x, x') = e^{-\frac{\|x-x'\|^2}{2\sigma^2}}$

$$\begin{aligned} k(x, x') &= e^{-\frac{\|x-x'\|^2}{2\sigma^2}} \\ &= e^{-\frac{(x-x')^T(x-x')}{2\sigma^2}} \\ &= e^{-\frac{x^Tx-2x^Tx'+x'^Tx'}{2\sigma^2}} \\ &= e^{-\frac{\|x\|^2+\|x'\|^2}{2\sigma^2}+\frac{2x^Tx'}{2\sigma^2}} \\ &= e^{-\frac{\|x\|^2+\|x'\|^2}{2\sigma^2}} \quad e^{\frac{x^Tx'}{\sigma^2}} \end{aligned}$$

We fix the first part as a constant ($c := e^{-\frac{\|x\|^2+\|x'\|^2}{2\sigma^2}}$) while we use the Taylor expansion over e^x ($e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}$) for the second part:

$$\begin{aligned} k(x, x') &= c \; e^{\frac{x^Tx'}{\sigma^2}} \\ &= c \sum_{n=0}^{\infty} \frac{(\frac{x^Tx'}{\sigma^2})^n}{n!} \\ &= c \sum_{n=0}^{\infty} \frac{(x^Tx')^n}{\sigma^{2n} n!} \end{aligned}$$

The numerator $(x^Tx')^n$ is the polynomial kernel. Therefore, the Gaussian kernel consists of an infinite sum over polynomial kernels of x and x', leading to an infinite dimensional feature space.

In the above equation, the term $e^{-\frac{\|x\|^2+\|x'\|^2}{2\sigma^2}}$ does not depend on the specific data points x and x', but only on the Euclidean norms of those vectors. Thus, it can be treated as a constant.