Exercise for Machine Learning (SS 20)

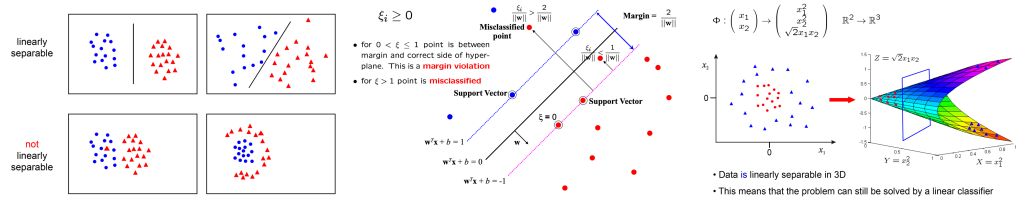
Assignment 6: Support Vector Machine

Prof. Dr. Steffen Staab, steffen.staab@ipvs.uni-stuttgart.de
Alex Baier, alex.baier@ipvs.uni-stuttgart.de
Janik Hager, janik-manel.hager@ipvs.uni-stuttgart.de
Ramin Hedeshy, ramin.hedeshy@ipvs.uni-stuttgart.de
Analytic Computing, IPVS, University of Stuttgart

Submit your solution in Ilias as either PDF for theory assignments or Jupyter notebook for practical assignments.
Mention the names of all group members and their immatriculation numbers in the file.
Submission is possible until the following Monday, 15.06.2020, at 14:00.

1 Concepts

Explain the following terms and how they are related to SVM in your own words and with (visual) examples:



(a) Linear separability (Zisserman 2015)

(b) Slack variables (Zisserman 2015)

(c) Kernel functions (Zisserman 2015)

Figure 1: Solution for Task 1.

1. Linear separability

   Solution: A data set is linearly separable if a linear function (also called discriminant) $\hat{f}(x)$ exists (linear in features) that separates the (two) classes of the data set, i.e. all data points of the one class are on one side (e.g. $\hat{f}(x) < 0$) of the line/plane/hyperplane spanned by $\hat{f}(x) = 0$ while all the data points belonging to the other class are on the other side (e.g. $\hat{f}(x) > 0$) of the line/plane/hyperplane.

2. Slack variables

   Solution: Slack variables ξ_i soften the objective up by allowing small violations of constraints (e.g. missclassifications or margin violations where a point is on the correct side of the hyperplane but between the margin and the hyperplane). This allows to find solutions for not linearly separable data sets.

3. Kernel functions

   Solution: Kernel functions measure the similarity of two data points x and x' (basically comparing them) by expressing how correlated their respective function outputs y and y' are. They help us to work directly on the data points without using the feature space first. Each kernel corresponds to a specific feature choice (although not always that obvious) and vice versa as follows: $k(x, x') = \phi(x)^\top \phi(x')$.

2 Perceptron

1. Define the classification function for the perceptron classifier.

   Solution: $\hat{f}(x) = \begin{cases} 1 & \text{if } w^\top x + b > 0 \\ -1 & \text{if } w^\top x + b < 0 \end{cases}$

2. The dataset for the OR function is given by:

$$X = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}$$

$$y = \begin{bmatrix} -1 & 1 & 1 & 1 \end{bmatrix}^\top$$

   Given the initial weights of $w = \begin{bmatrix} 1 & -1 & 0.5 \end{bmatrix}$, where w_3 is the bias. Perform the perceptron algorithm (slide 10) with $\alpha = 0.6$ until all data points are correctly classified. Show your computations for each training step. (Note: In the case of $w \cdot x = 0$ output 1.)

   Solution:

   $\odot$ denotes the point-wise multiplication. Append a one for every data point to include the bias: $X = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

   1. iteration:

   $\hat{y} = \hat{f}(X) = Xw = \begin{bmatrix} 0.5 & -0.5 & 1.5 & 0.5 \end{bmatrix}^\top$, $\hat{y} \odot y = \begin{bmatrix} -0.5 & -0.5 & 1.5 & 0.5 \end{bmatrix}^\top$

   Fix first data point: $w_{new} = w_{old} - 0.6\ x_1\ \text{sign}(\hat{f}(x_1)) = \begin{bmatrix} 1 & -1 & -0.1 \end{bmatrix}^\top$

   2. iteration:

   $\hat{y} = \hat{f}(X) = Xw = \begin{bmatrix} -0.1 & -1.1 & 0.9 & -0.1 \end{bmatrix}^\top$, $\hat{y} \odot y = \begin{bmatrix} 0.1 & -1.1 & 0.9 & -0.1 \end{bmatrix}^\top$

   Fix second data point: $w_{new} = w_{old} - 0.6\ x_2\ \text{sign}(\hat{f}(x_2)) = \begin{bmatrix} 1 & -0.4 & 0.5 \end{bmatrix}^\top$

   3. iteration:

   $\hat{y} = \hat{f}(X) = Xw = \begin{bmatrix} 0.5 & 0.1 & 1.5 & 1.1 \end{bmatrix}^\top$, $\hat{y} \odot y = \begin{bmatrix} -0.5 & 0.1 & 1.5 & 1.1 \end{bmatrix}^\top$

   Fix first data point: $w_{new} = w_{old} - 0.6\ x_1\ \text{sign}(\hat{f}(x_1)) = \begin{bmatrix} 1 & -0.4 & -0.1 \end{bmatrix}^\top$

   4. iteration:

   $\hat{y} = \hat{f}(X) = Xw = \begin{bmatrix} -0.1 & -0.5 & 0.9 & 0.5 \end{bmatrix}^\top$, $\hat{y} \odot y = \begin{bmatrix} 0.1 & -0.5 & 0.9 & 0.5 \end{bmatrix}^\top$

Fix second data point: $w_{new} = w_{old} - 0.6\ x_2\ \text{sign}(\hat{f}(x_2)) = \begin{bmatrix}1 & 0.2 & 0.5\end{bmatrix}^\top$

5. iteration:

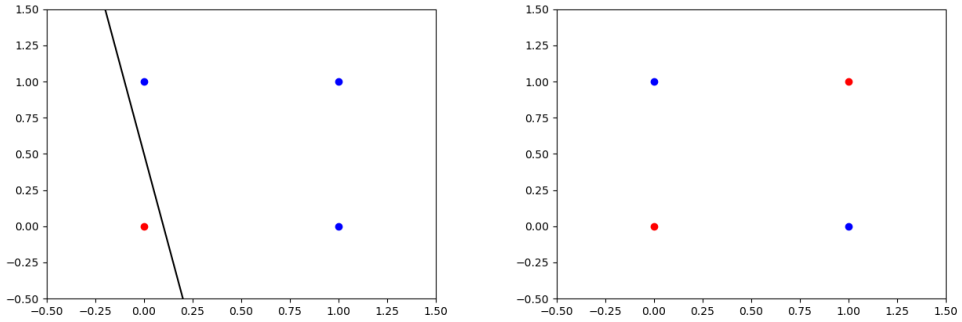
$\hat{y} = \hat{f}(X) = Xw = \begin{bmatrix}0.5 & 0.7 & 1.5 & 1.7\end{bmatrix}^\top$, $\hat{y} \odot y = \begin{bmatrix}-0.5 & 0.7 & 1.5 & 1.7\end{bmatrix}^\top$

Fix first data point: $w_{new} = w_{old} - 0.6\ x_1\ \text{sign}(\hat{f}(x_1)) = \begin{bmatrix}1 & 0.2 & -0.1\end{bmatrix}^\top$

6. iteration:

$\hat{y} = \hat{f}(X) = Xw = \begin{bmatrix}-0.1 & 0.1 & 0.9 & 1.1\end{bmatrix}^\top$, $\hat{y} \odot y = \begin{bmatrix}0.1 & 0.1 & 0.9 & 1.1\end{bmatrix}^\top$

We are done since $\forall x_i : \hat{y}_i \odot y_i > 0$. Final $w = \begin{bmatrix}1 & 0.2 & -0.1\end{bmatrix}^\top$.



(a) Resulting discriminant function for OR. (Task 2.2)

(b) No linear separability for XOR. (Task 2.3)

Figure 2: Solution for Task 2.

3. Prove that the XOR function cannot be represented by a (linear) perceptron.

   Solution: As can be seen in Figure 2b, this data set is not linearly separable.

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:

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

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

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

$$
\begin{aligned}
k(x, x') &= e^{-\frac{\|x-x'\|^2}{2\sigma^2}} \\
&= e^{-\frac{(x-x')^\top (x-x')}{2\sigma^2}} \\
&= e^{-\frac{x^\top x - 2x^\top x' + x'^\top x'}{2\sigma^2}} \\
&= e^{-\frac{\|x\|^2 + \|x'\|^2}{2\sigma^2} + \frac{2x^\top x'}{2\sigma^2}} \\
&= e^{-\frac{\|x\|^2 + \|x'\|^2}{2\sigma^2}} \; e^{\frac{x^\top x'}{\sigma^2}}
\end{aligned}
$$

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:

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

The numerator $(x^\top x')^n$ 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.