| | | |
|---|---|---|
| **Name, Vorname:** Wang, Jiahui | **Fach:** B.Sc.Mathematik | **Martriknr.**2992080 |
| **Name, Vorname:** Zhao, Jiaqi | **Fach:** M.Sc.Autonome System | **Martriknr.** 3470190 |
| **Name, Vorname:** Wang, Ziyin | **Fach:** M.Sc.Informatik | **Martriknr.** 3435397 |



# Assignment 6

## Task 1: Support Vector Machines (SVM) Concepts

1. Linear separability:
   The two classes of a given data set can totally linear separately by a linear function(discriminant) $\hat{f}(x) = w^\top x + b$, namely the data of one class totally on one side of discriminant ($\hat{f}(x) < 0$), the data of another class totally on another side ($\hat{f}(x) > 0$). In 2D space the discriminant is a line ($\hat{f}(x) = 0$), in 3D space the discriminant is a plane and in n-dim space the discriminant is a hyperplane. Examples1 are following:

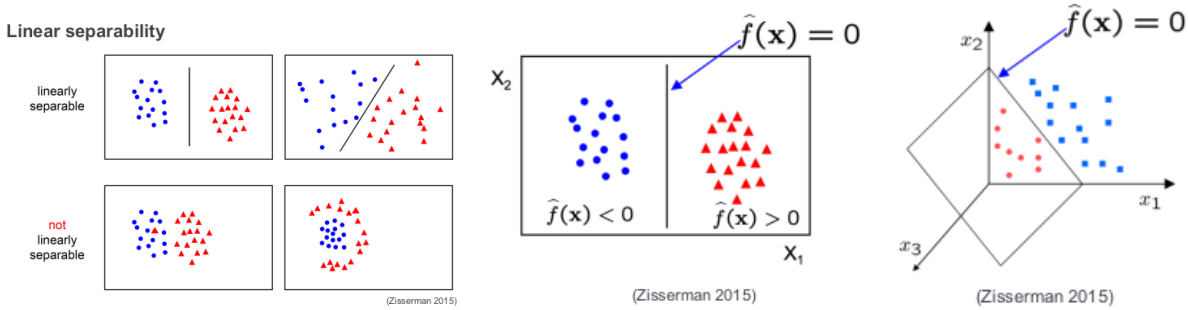


Abbildung 1: The examples of linear separability

2. Slack variables:
   If a noise point exists in the data set, and for this noise, SVM wants to make the correct classification, then the discriminant will be close to another class, which means, points can be linearly separated, but with a very narrow margin, so it does not correspond the constraint of a large margin. For solving this problem, SVM builds soft margin classifier by using slack variables $\xi_i$, this allows that some samples could not satisfy the large margin constraint.And it can revise optimization problem, avoid overfitting. Examples2 are following:

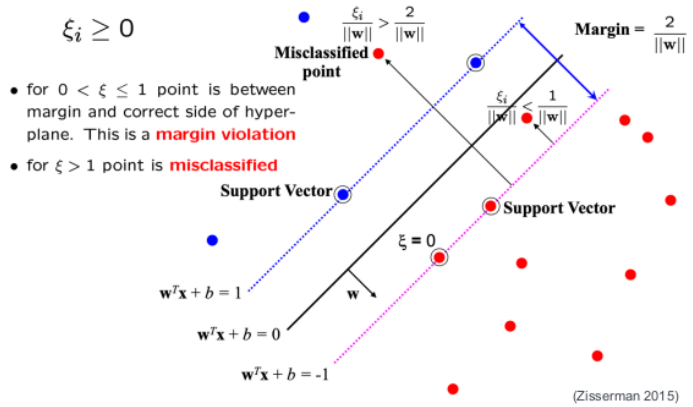


Abbildung 2: The examples of slack variable

**3.** Kernel functions:

The function of the kernel function ($k(x_i, x_j) = \phi(x_j)^\top \phi(x_i)$) is to imply a mapping from low-dimensional space to high-dimensional space, and this mapping can turn the two classes of linearly inseparable points in the low-dimensional space into linearly separable. Examples3 are following:

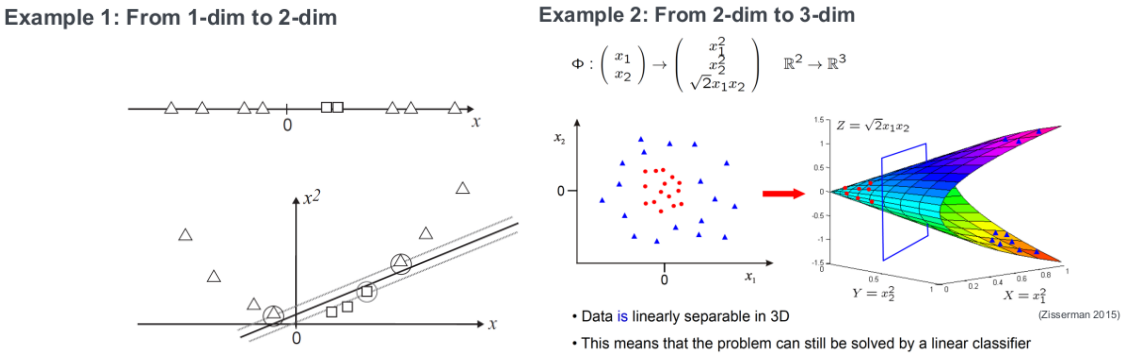


Abbildung 3: The examples of kernel function

## Task 2: Perceptron

**1.** The classification function for the perceptron classifer is:

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

**2.** First we append 1 for every data point with the bias:

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

then we should perform the first iteration:

$$\hat{y} = \hat{f}(X) = Xw = \begin{bmatrix} 0.5 \\ -0.5 \\ 1.5 \\ 0.5 \end{bmatrix}$$

$$\hat{y} \odot y = \begin{bmatrix} -0.5 \\ -0.5 \\ 1.5 \\ 0.5 \end{bmatrix}$$

Fixing the first data point, we get $w_n = w_o - \alpha sign(\hat{f}(x_1)) = \begin{bmatrix} 1 \\ -1 \\ -0.1 \end{bmatrix}$.

Then we perform the second iteration:

$$\hat{y} = \hat{f}(X) = Xw = \begin{bmatrix} -0.1 \\ 1.1 \\ 0.9 \\ -0.1 \end{bmatrix}$$

$$\hat{y} \odot y = \begin{bmatrix} -0.1 \\ 1.1 \\ 0.9 \\ -0.1 \end{bmatrix}$$

Fixing the second data point, we get $w_n = w_o - \alpha sign(\hat{f}(x_1)) = \begin{bmatrix} 1 \\ -0.4 \\ 0.5 \end{bmatrix}$.

Then we perform the third iteration:

$$\hat{y} = \hat{f}(X) = Xw = \begin{bmatrix} 0.5 \\ 0.1 \\ 1.5 \\ 1.1 \end{bmatrix}$$

$$\hat{y} \odot y = \begin{bmatrix} -0.5 \\ 0.1 \\ 1.5 \\ 1.1 \end{bmatrix}$$

Fixing the first data point, we get $w_n = w_o - \alpha sign(\hat{f}(x_1)) = \begin{bmatrix} 1 \\ -0.4 \\ -0.1 \end{bmatrix}$.

Then we perform the fourth iteration:

$$\hat{y} = \hat{f}(X) = Xw = \begin{bmatrix} -0.1 \\ -0.5 \\ 0.9 \\ 0.5 \end{bmatrix}$$

$$\hat{y} \odot y = \begin{bmatrix} -0.1 \\ -0.5 \\ 0.9 \\ 0.5 \end{bmatrix}$$

Fixing the second data point, we get $w_n = w_o - \alpha sign(\hat{f}(x_1)) = \begin{bmatrix} 1 \\ 0.2 \\ 0.5 \end{bmatrix}$.

Then we perform the fifth iteration:

$$\hat{y} = \hat{f}(X) = Xw = \begin{bmatrix} 0.5 \\ 0.7 \\ 1.5 \\ 1.7 \end{bmatrix}$$

$$\hat{y} \odot y = \begin{bmatrix} -0.5 \\ 0.7 \\ 1.5 \\ 1.7 \end{bmatrix}$$

Fixing the first data point, we get $w_n = w_o - \alpha sign(\hat{f}(x_1)) = \begin{bmatrix} 1 \\ 0.2 \\ -0.1 \end{bmatrix}$.

Then we perform the sixth iteration:

$$\hat{y} = \hat{f}(X) = Xw = \begin{bmatrix} -0.1 \\ 0.1 \\ 0.9 \\ 1.1 \end{bmatrix}$$

$$\hat{y} \odot y = \begin{bmatrix} 0.1 \\ 0.1 \\ 0.9 \\ 1.1 \end{bmatrix}$$

Now we can find that for all x, $\hat{y} \odot y > 0$, so we can get the final $w = \begin{bmatrix} 1 \\ 0.2 \\ -0.1 \end{bmatrix}$

## Task 3: Polynomial Kernel

First we calculate the scalar product of $(\phi(x_i), \phi(x_j))$:

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

so it only uses the scalar product of two-dimensional vectors instead of mapping of the two-dimensional vector to three dimensions.

## Task 4: Gaussian Kernel

Gaussian kernel is:

$$\begin{aligned}k(x, \acute{x}) &= e^{-\frac{||x-\acute{x}||^2}{2\sigma^2}} \\ &= e^{-\frac{(x-\acute{x})^\mathsf{T}(x-\acute{x})}{2\sigma^2}} \\ &= e^{-\frac{x^\mathsf{T}x-2x^\mathsf{T}\acute{x}+\acute{x}^\mathsf{T}\acute{x}}{2\sigma^2}} \\ &= e^{-\frac{||x||^2+||\acute{x}||^2}{2\sigma^2}} e^{\frac{x^\mathsf{T}\acute{x}}{\sigma^2}}\end{aligned}$$

We can use the Taylor expansion over $e^x$ for $e^{\frac{x^\mathsf{T}\acute{x}}{\sigma^2}}$:

$$\begin{aligned}k(x, \acute{x}) &= \underbrace{e^{-\frac{||x||^2+||\acute{x}||^2}{2\sigma^2}}}_{const.=a} e^{\frac{x^\mathsf{T}\acute{x}}{\sigma^2}} \\ &= a\sum_{n=0}^{\infty}\frac{(\frac{x^\mathsf{T}\acute{x}}{\sigma^2})^2}{n!} \\ &= a\sum_{n=0}^{\infty}\frac{(x^\mathsf{T}\acute{x})^2}{\sigma^{2n}n!}\end{aligned}$$

We can find that after the Taylor expansion in the equation, $(x^\mathsf{T}\acute{x})^2$ is a Polynomial Kernel wie in Task 3.
So we can come to a conclusion that the Gaussian kernel consists of an infinite sum of the polynomial kernels of $x, \acute{x}$, in other words, it projects to an infinite dimensional feature space.