AI Strategy and Digital transformation
4. Support Vector Machine (and Regression)

Piotr Wójcik
University of Warsaw (Poland)
pwojcik@wne.uw.edu.pl

January 2025

Support vector machine – introduction

- **support vector machine** (**SVM**) is designed as a **classification** tool
- was invented by Cortes and Vapnik (1995) and has since grown significantly in popularity
- the concept of **SVM** is similar to the discriminant analysis
- the aim is to find in a multidimensional space a **hyperplane separating observations** from different groups
- **SVM** can handle any number of data dimensions
- similar approach can be used in regression tasks and is in this case called **support vector regression** (SVR)

Hyperplane

- formally a hyperlane in p dimensions is defined as a **flat subspace** having $p - 1$ dimensions
- the general formula defining the hyperplane is:

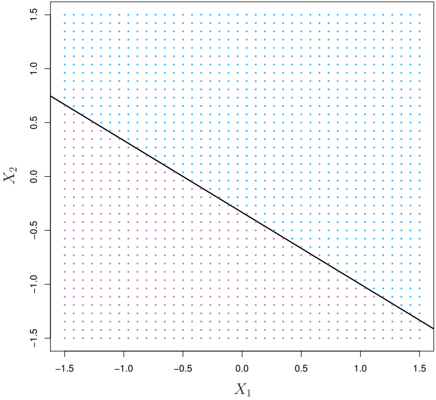
$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p = 0$$

- for a two-dimensional space ($p = 2$), the hyperplane is a flat space of the order of 1, or **a straight line**
- for a three-dimensional space ($p = 3$), the hyperplane is a flat space of order 2, i.e. **two-dimensional surface**, etc.
- if $\beta_0 = 0$, the hyperplane plane goes through a point $[0, 0, \ldots, 0]$

Hyperplane – cont'd

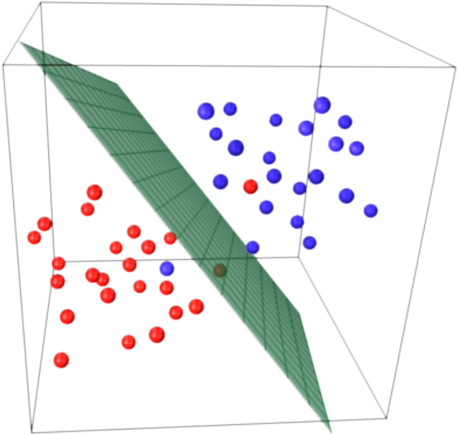
- if for a point with coordinates $X = (X_1, X_2, \ldots, X_p)$ the above equation is satisfied, it is located **on the hyperplane**
- if $\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p > 0$, the point is located **on one side of the hyperplane**
- while if $\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p < 0$ the point is located **on the other side of the hyperplane**
- one can therefore treat the hyperplane as the mechanism of **dividing of space into two parts**
- determining on which side of the hyperplane the point lies requires only checking the sign of the expression $\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p$

Hyperplane $1 + 2X_1 + 3X_2 = 0$ in two dimensions

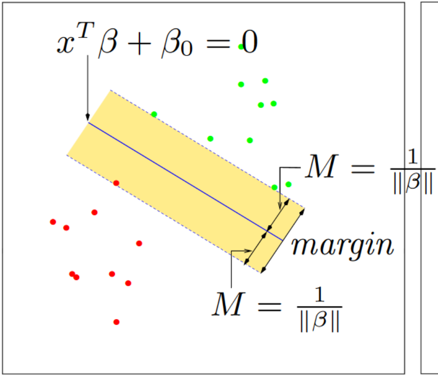


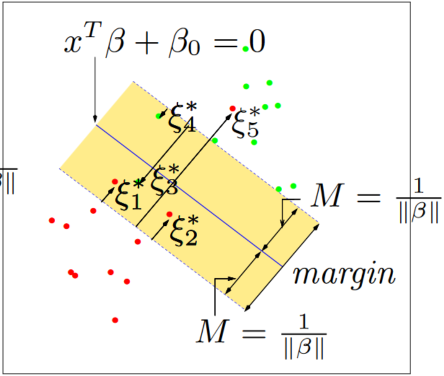
Source: James et al. (2017), p. 339

Two dimensional hyperplane in three dimensions



Maximum margin classifier vs Support vector classifier





Source: Hastie et al. (2009), p. 418

Penalty for wrong classifications

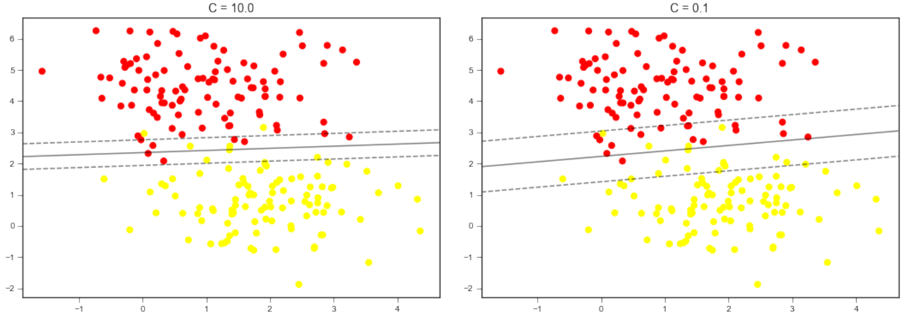
- in the case of the **soft margin** for the optimization problem, the parameter `C` is added, which determines the weight (**penalty**) attached to incorrect classifications:

$$\min_{\beta_0,\beta} \left(\frac{1}{2}||\beta||^2 + C\sum_{i=1}^{n}\xi_i\right)$$

such that: $\forall i = 1, 2, \ldots, n : y_i(\beta_0 + x_i^T\beta) \geq 1 - \xi_i$,

- in practice, the selection of the optimal value of C is done with the **cross-validation** of the model
- C is responsible for the problem of **bias-variance trade-off**

Sample impact of C on classification results

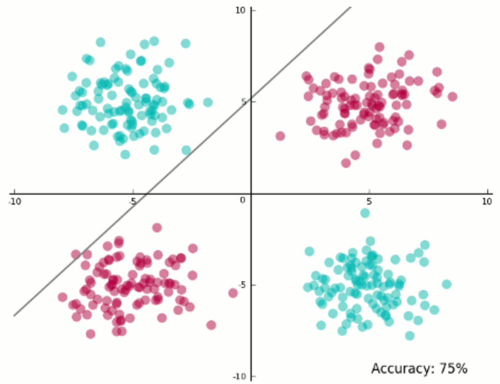


Support vector classifier – support vectors

- **support vectors** are the observations that affect the position of the hyperplane separating the groups
- in the case of this classifier these are all observations located **at the edge of the margin** of the hyperplane and also those located **on the wrong side of the margin**
- other observations do not affect the result of the classification
- this means that the support vector classifier is **not influenced** by the exact location of observations **on the right side of the margins**

Nonlinear boundaries between groups

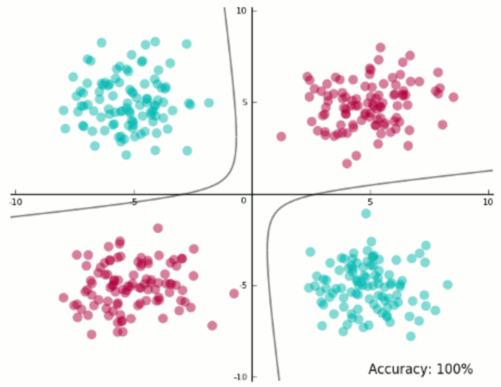
- the boundary between two groups in the data **does not have to be linear**
- in this case the **support vector classifier** or any other linear classifier will be useless



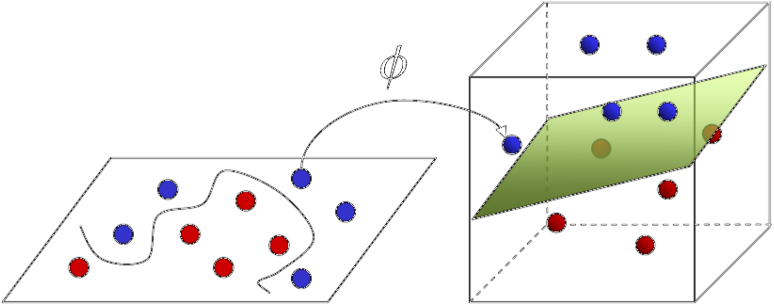
Nonlinear boundaries between groups – cont'd

- the solution to this problem could be the manual **extension of the set of model features** by adding successive powers of explanatory variables or generally their non-linear transformations, e.g. $(x+y)^2 = x^2 + 2xy + y^2$
- in this way the problem is moved **from two-dimensional space to three-dimensional space**
- **intuition**: if separation of the groups using a **hyperplane** between groups is not possible in p dimensions, try to map data to **more dimensions**, where **separation with a hyperplane will be possible**

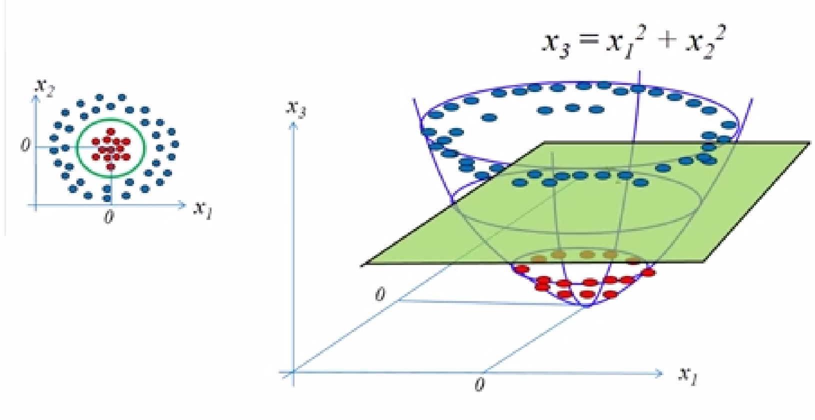
SVM – mapping data into more dimensions



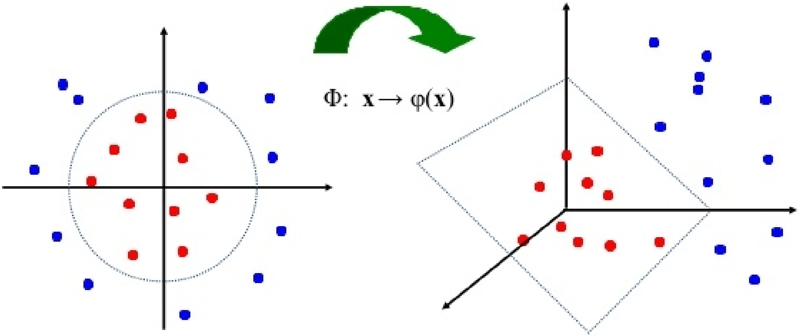
Mapping data into more dimensions – example 1



Mapping data into more dimensions – example 2



Mapping data into more dimensions – example 3



Support vector machine (SVM)

- manual extension of the set of features is limited only by the analyst's creativity
- however, the increase in the number of variables results in the **increase of the computational complexity** of the optimization problem
- the solution to this problem is the method known as the **support vector machine**
- the support vector machine (**SVM**) is an extension of the idea of **support vector classifier**
- it extends the set of analyzed features in a specific way – by **indirectly** mapping the data to a **more dimensional space** using a selected **kernel function**

Support vector machine (SVM)

- having more dimensions (features) one can better separate data
- however, with the increase in the number of features, the number of model parameters also increases which has impact on the risk of **overfitting**
- in the case of SVM, the so-called **kernel trick** is used
- it consists in transforming data in such a way as if one added new variables (data dimensions) to the model, but without physically generating new columns in the data
- this allows to achieve **an analogous effect**, like extending the set of features, but is **much less computationally intensive**

Kernel function

- the transformation applied is called a **kernel function**
- it is a function of two variables $k(x, z)$, such that $k(x, z) > 0$ and $k(x, z) = k(z, x)$
- it can therefore be identified with the **measure of similarity**
- the formula from the optimization problem $y_i(\beta_0 + x_i^T \beta) \geq 1$ can be alternatively defined using the **scalar product** (**dot product**) of the weight vector β and features' values x: $y_i(\beta_0 + \langle x_i, \beta \rangle) \geq 1$
- suppose one wants to apply an additional **transformation** $h()$: $y_i(\beta_0 + \langle h(x_i), h(\beta) \rangle) \geq 1$,

Kernel trick

- kernel function is **positive semi-definite**, if for any n, any $x_1, x_2, \ldots, x_n$ and any real values $c_1, c_2, \ldots, c_n$:

$$\sum_{i=1}^{n} \sum_{j=1}^{n} c_i c_j k(x_i, x_j) \geq 0$$

- if a kernel function is **positive semi-definite**, then there exists such $\phi()$, that

$$k(x, \beta) = \langle \phi(x), \phi(\beta) \rangle$$

- therefore using the kernel function gives **the same effect as using a scalar product on an extended feature space**
- one does not have to define the function $\phi()$ – the only thing needed is the selection of the **appropriate kernel function**

Kernel trick – example

- suppose one has two X variables, and the kernel function is simply a quadratic function
- then one can write:

$$
\begin{aligned}
k(x,\beta) &= (x'\beta)^2 \\
&= x_1^2\beta_1^2 + 2x_1\beta_1x_2\beta_2 + x_2^2\beta_2^2 \\
&= (x_1^2, \sqrt{2}x_1x_2, x_2^2)(\beta_1^2, \sqrt{2}\beta_1\beta_2, \beta_2^2)' \\
&= \phi(x)'\phi(\beta) = \langle\phi(x), \phi(\beta)\rangle
\end{aligned}
$$

- the effect is the same as if one used the **squares of both variables** and their **interaction** in the model

Common kernel functions

- linear kernel:

$$K(x,z) = x'z + 1$$

- polynomial kernel:

$$K(x,z) = (s * x'z + 1)^d$$

- Gaussian kernel:

$$K(x,z) = exp(-\frac{||x-z||^2}{2\sigma^2})$$

- Gaussian radial basis function (RBF):

$$K(x,z) = exp(-\gamma||x-z||^2)$$

Common kernel functions – cont'd

- Laplace RBF kernel:

$$K(x, z) = exp(-\frac{||x - z||}{\sigma})$$

- Hyperbolic tangent kernel:

$$K(x, z) = tanh(\kappa x'z + c)$$

- Sigmoid kernel:

$$K(x, z) = tanh(\alpha x'z + c)$$

Common kernel functions – cont'd

Jądro liniowe (ang. *linear kernel*):

$$k(\mathbf{x}, \mathbf{z}) = \mathbf{x}^{\mathrm{T}}\mathbf{z}$$

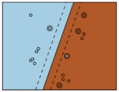
Jądro wielomianowe (ang. *polynomial kernel*):

$$k(\mathbf{x}, \mathbf{z}) = (\mathbf{x}^{\mathrm{T}}\mathbf{z} + 1)^{M}$$

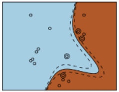
Jądro gaussowskie (ang. *gaussian kernel*):

$$k(\mathbf{x}, \mathbf{z}) = \exp\left\{-\frac{||\mathbf{x} - \mathbf{z}||^2}{2\sigma^2}\right\}$$

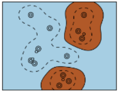
Linear



Poly

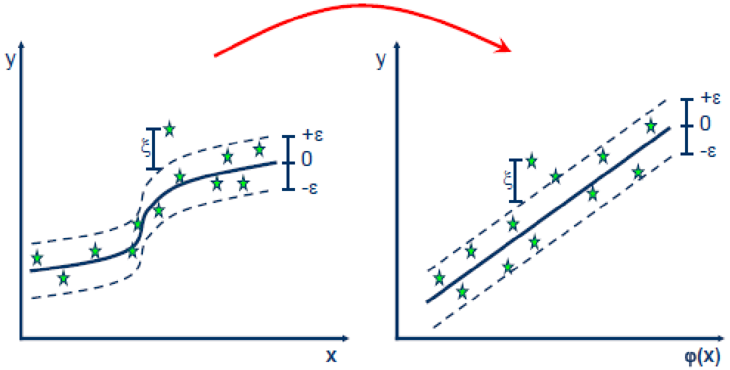


Gaussian



Support Vector Regression (SVR)

- a similar approach can be used to solve the regression problem
- **support vector regression** (SVR) adapts the hyperplane to the data in such a way that as many data points as possible are at a distance from it not greater than ϵ



SMV – practical exercises in python

Thank you for your attention