**PS5841**

# Data Science in Finance & Insurance

# Support Vector Machines

Yubo Wang

Autumn 2021

COLUMBIA UNIVERSITY

# SVC

$$L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{k=1}^{N} \alpha_i \alpha_k y_i y_k \boldsymbol{x}_i^T \boldsymbol{x}_k$$

$$\hat{f}(\boldsymbol{x}) = \boldsymbol{x}^T \widehat{\boldsymbol{\beta}} + \hat{\beta}_0 = \sum_{i=1}^{N} \hat{\alpha}_i \, y_i \boldsymbol{x}^T \boldsymbol{x}_i + \hat{\beta}_0$$

$$\boldsymbol{\beta} = \sum_{i=1}^{N} \alpha_i y_i \boldsymbol{x}_i$$

COLUMBIA UNIVERSITY

# SVC with Kernel Function (1)

- Generalize the inner products to kernel functions

$$
\begin{aligned}
L_D &= \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{k=1}^{N} \alpha_i \alpha_k y_i y_k \boldsymbol{x}_i^T \boldsymbol{x}_k \\
&= \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{k=1}^{N} \alpha_i \alpha_k y_i y_k < \boldsymbol{x}_i, \boldsymbol{x}_k > \\
\rightarrow L_D &= \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{k=1}^{N} \alpha_i \alpha_k y_i y_k < h(\boldsymbol{x}_i), h(\boldsymbol{x}_k) > \\
&= \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{k=1}^{N} \alpha_i \alpha_k y_i y_k K(\boldsymbol{x}_i, \boldsymbol{x}_k)
\end{aligned}
$$

- For the linear kernel function, $K(\boldsymbol{x}, \boldsymbol{x}') =< h(\boldsymbol{x}), h(\boldsymbol{x}') > = \boldsymbol{x}^T \boldsymbol{x}'$

COLUMBIA UNIVERSITY

# SVC with Kernel Function (2)

- Generalize the inner products to kernel functions

$$\hat{f}(\boldsymbol{x}) = \boldsymbol{x}^T\widehat{\boldsymbol{\beta}} + \hat{\beta}_0 = \sum_{i=1}^{N} \hat{\alpha}_i\, y_i \boldsymbol{x}^T \boldsymbol{x}_i + \hat{\beta}_0 = \sum_{i=1}^{N} \hat{\alpha}_i\, y_i < \boldsymbol{x}, \boldsymbol{x}_i > +\hat{\beta}_0$$

$$\rightarrow \hat{f}(\boldsymbol{x}) = \sum_{i=1}^{N} \hat{\alpha}_i\, y_i < h(\boldsymbol{x}), h(\boldsymbol{x}_i) > +\hat{\beta}_0$$

$$= \sum_{i=1}^{N} \hat{\alpha}_i\, y_i K(\boldsymbol{x}, \boldsymbol{x}_i) + \hat{\beta}_0$$

- For the linear kernel function, $K(\boldsymbol{x}, \boldsymbol{x}') =< h(\boldsymbol{x}), h(\boldsymbol{x}') > = \boldsymbol{x}^T \boldsymbol{x}'$

COLUMBIA UNIVERSITY

# Feature Space Expansion

- A kernel function can expand the feature space.
- Example – from 2D to 6D

$$\boldsymbol{x} = (x_1, x_2)^T \rightarrow h(\boldsymbol{x}) = (h_1(\boldsymbol{x}), \dots, h_6(\boldsymbol{x}))^T$$

$$K(\boldsymbol{x}_i, \boldsymbol{x}_k) = (1 + < \boldsymbol{x}_i, \boldsymbol{x}_k >)^2$$

COLUMBIA UNIVERSITY

# Example – from 2D to 6D (1)

$$\boldsymbol{x} = (x_1, x_2)^T \rightarrow h(\boldsymbol{x}) = (h_1(\boldsymbol{x}), \ldots, h_6(\boldsymbol{x}))^T$$

$$\begin{gathered} K(\boldsymbol{x}_i, \boldsymbol{x}_k) = (1 + < \boldsymbol{x}_i, \boldsymbol{x}_k >)^2 \\ = (1 + x_{i1}x_{k1} + x_{i2}x_{k2})^2 \\ = 1 + (x_{i1}x_{k1})^2 + (x_{i2}x_{k2})^2 \\ +2x_{i1}x_{k1} + 2x_{i2}x_{k2} + 2x_{i1}x_{k1}x_{i2}x_{k2} \end{gathered}$$

COLUMBIA UNIVERSITY

# Example – from 2D to 6D (2)

$$K(\boldsymbol{x}_i, \boldsymbol{x}_k) = 1 + (x_{i1}x_{k1})^2 + (x_{i2}x_{k2})^2$$
$$+2x_{i1}x_{k1} + 2x_{i2}x_{k2} + 2x_{i1}x_{k1}x_{i2}x_{k2} =< h(\boldsymbol{x}_i), h(\boldsymbol{x}_k) >$$

- $h_1(\boldsymbol{x}) = 1 \rightarrow h_1(\boldsymbol{x}_i)h_1(\boldsymbol{x}_k) = 1$
- $h_2(\boldsymbol{x}) = x_1^2 \rightarrow h_2(\boldsymbol{x}_i)h_2(\boldsymbol{x}_k) = (x_{i1}x_{k1})^2$
- $h_3(\boldsymbol{x}) = x_2^2 \rightarrow h_3(\boldsymbol{x}_i)h_3(\boldsymbol{x}_k) = (x_{i2}x_{k2})^2$
- $h_4(\boldsymbol{x}) = \sqrt{2}x_1 \rightarrow h_4(\boldsymbol{x}_i)h_4(\boldsymbol{x}_k) = 2x_{i1}x_{k1}$
- $h_5(\boldsymbol{x}) = \sqrt{2}x_2 \rightarrow h_5(\boldsymbol{x}_i)h_5(\boldsymbol{x}_k) = 2x_{i2}x_{k2}$
- $h_6(\boldsymbol{x}) = \sqrt{2}x_1x_2 \rightarrow h_6(\boldsymbol{x}_i)h_6(\boldsymbol{x}_k) = 2x_{i1}x_{k1}x_{i2}x_{k2}$

COLUMBIA UNIVERSITY

# Support Vector Machine

- The support vector machine is an extension of the support vector classifier, expanding the feature space using kernels
- Linear $K(\boldsymbol{x}, \boldsymbol{x}') = <\boldsymbol{x}, \boldsymbol{x}'> = \boldsymbol{x}^T \boldsymbol{x}'$
- Polynomial $K(\boldsymbol{x}, \boldsymbol{x}') = (1 + <\boldsymbol{x}, \boldsymbol{x}'>)^d$
- Radial basis $K(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\gamma \|\boldsymbol{x} - \boldsymbol{x}'\|^2\right)$
- Neural Network

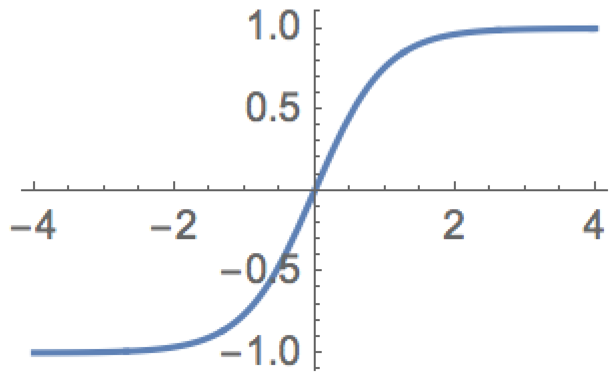
$$K(\boldsymbol{x}, \boldsymbol{x}') = \tanh(\kappa_1 <\boldsymbol{x}, \boldsymbol{x}'> + \kappa_2)$$

COLUMBIA UNIVERSITY

# tanh

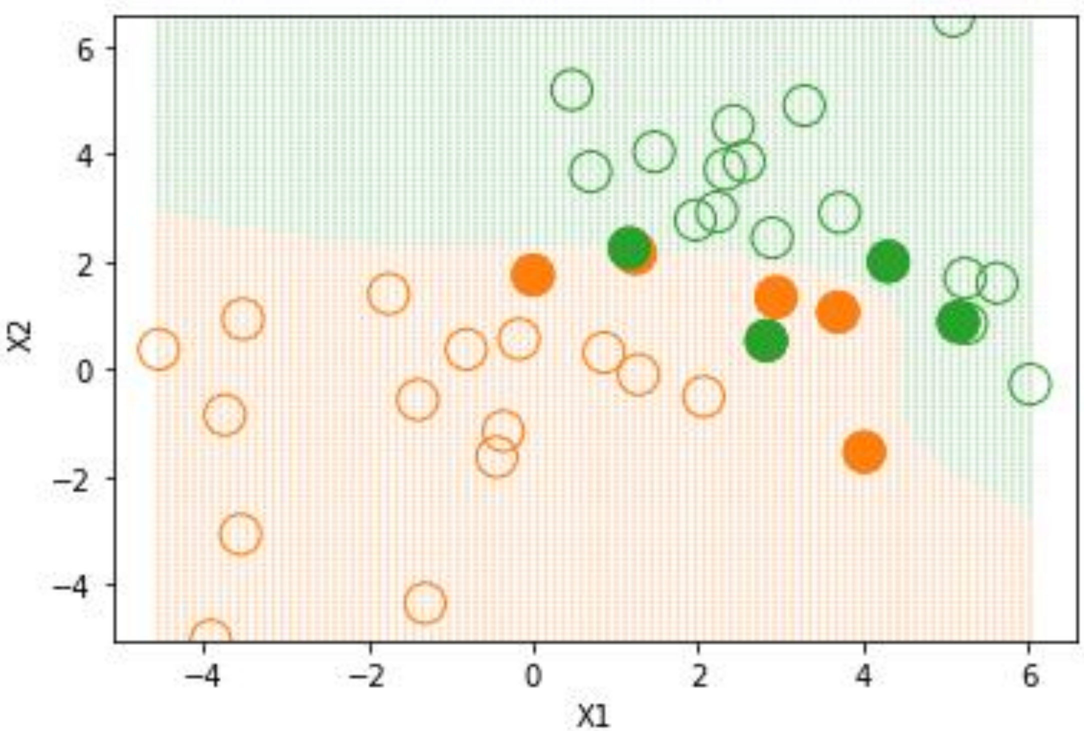
- Neural Network

$$K(\boldsymbol{x}, \boldsymbol{x}') = \tanh(\kappa_1 < \boldsymbol{x}, \boldsymbol{x}' > +\kappa_2)$$



# Decision Boundary (SVM) non-separable case

- Poly3, C=1



# That was


Columbia University