UiT The Arctic University of Norway

DTE-2502 Neural Networks: Support Vector Machines

Kalyan Ram Ayyalasomayajula, PhD

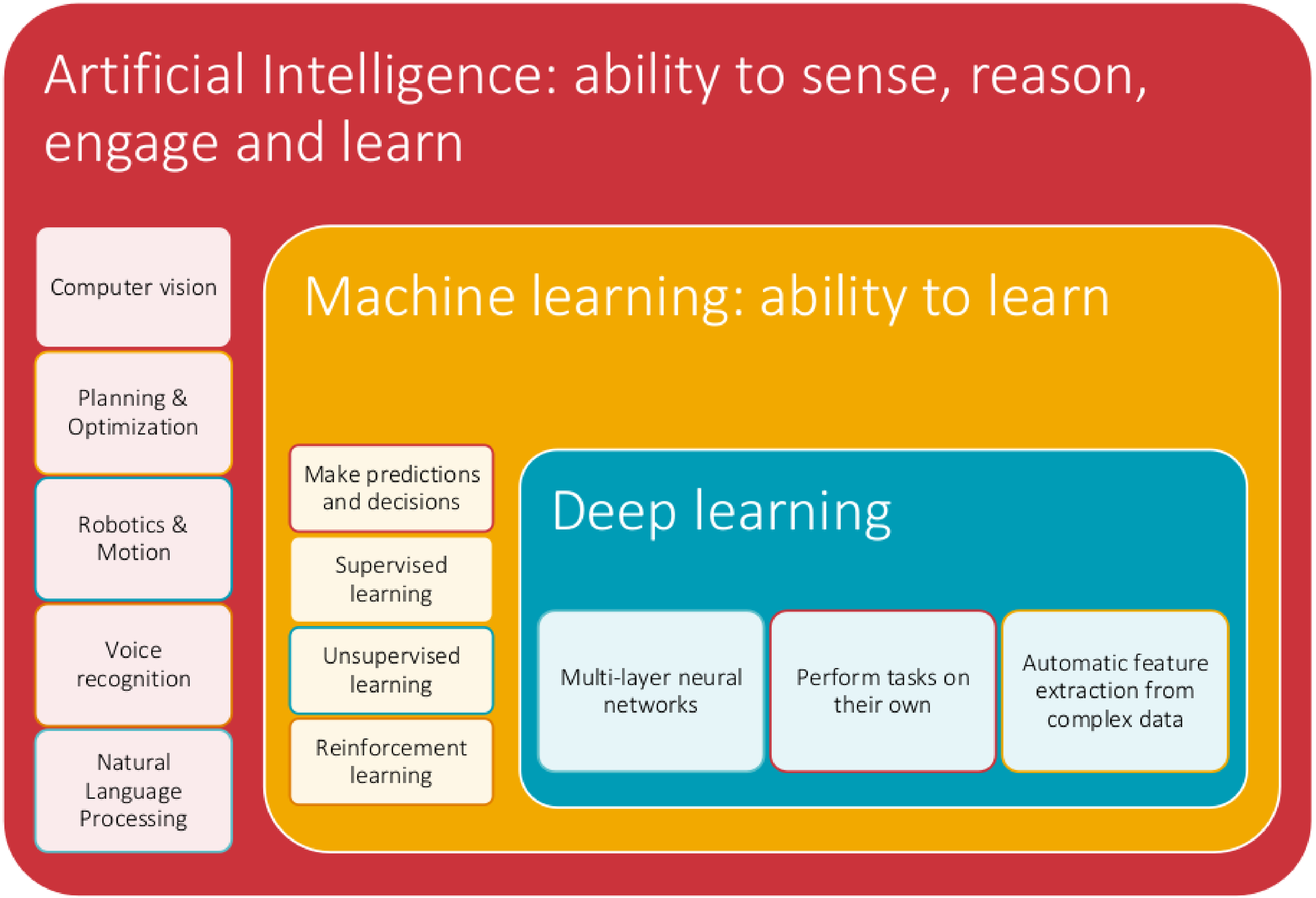
Associate professor, UiT Narvik

Email: kay001@post.uit.no

Overview

- Introduction to AI, ML and Deep learning
- Regression vs Classification
- Recap of linear algebra
- SVM: Theory
- SVM: Implementation

AI, ML and Deep learning

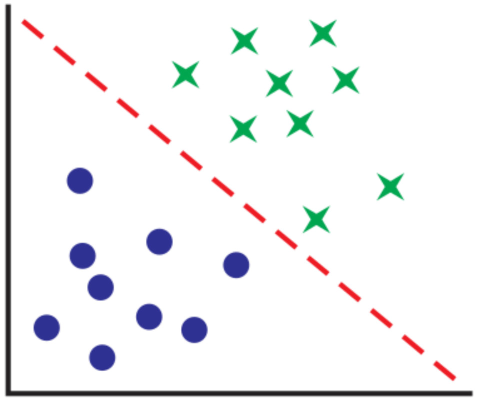


Artificial Intelligence is the theory and development of computer systems able to perform tasks normally requiring human intelligence, such as visual perception, speech recognition, decision-making, and translation between languages.

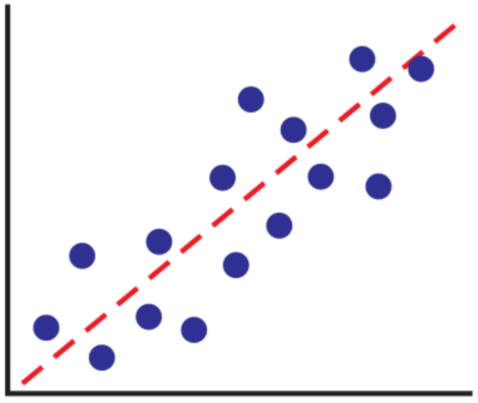
Artificial neural networks are machine learning techniques that simulate the mechanism of learning in biological organisms.

Regression vs Classification

Classification



Regression



- Regression: Curve fitting
- Classification: Decision making

Regression vs Classification

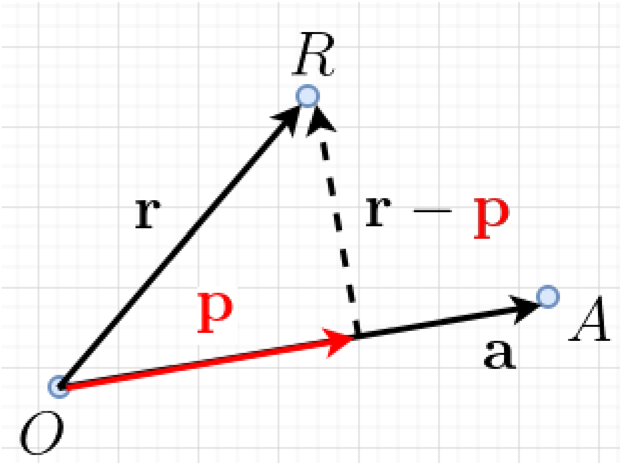
- Regression
- Output variable is continuous nature or real value.
- Find the best fit curve, which can predict the output more accurately
- Eg: Weather Prediction, Stock price prediction etc

- Classification
- Output variable is discrete value.
- Find the best decision boundary, which can seperate different classes in data.
- Eg: Spam emails, Face recognition etc

Support Vector Machine classifier

- SVM is a popular linear method prior to the deep learning trend.
- We might be interested in analysing data with a linear method before trying out advance non-linear methods.
- They are still very popular in fields where model explainability is very important.

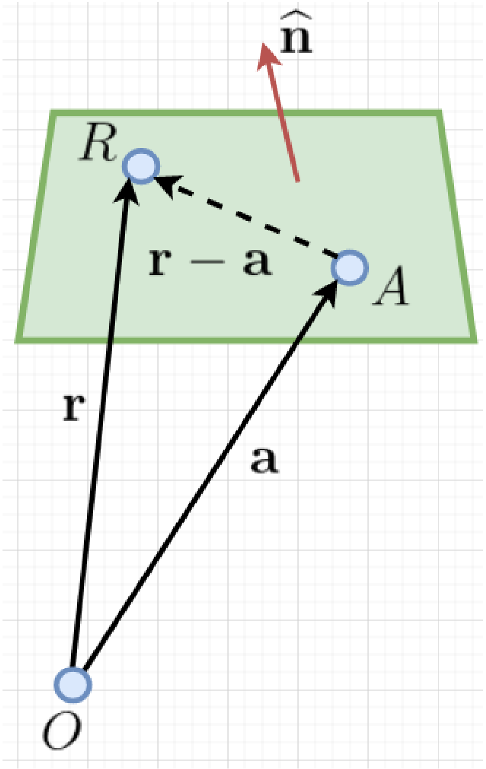
Recap of vector identities - Projections



$$\mathbf{p} = (\mathbf{r} \cdot \mathbf{a})\mathbf{u}$$

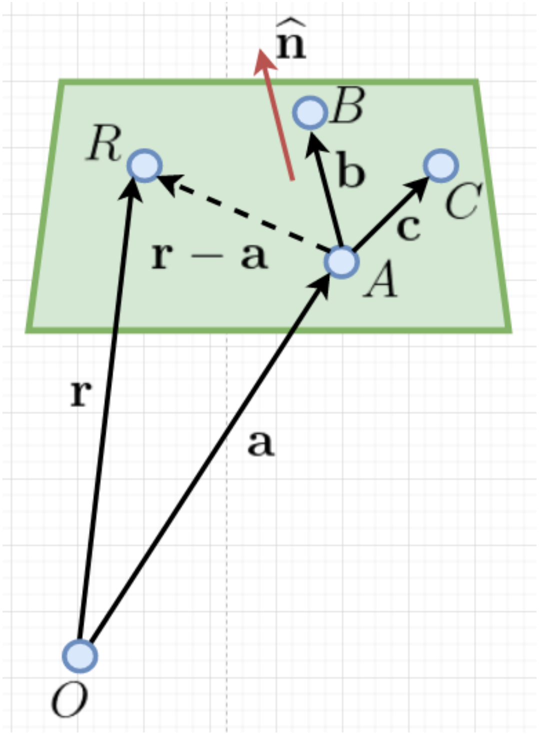
$$\mathbf{u} = \frac{\mathbf{a}}{|\mathbf{a}|}$$

Recap of vector identities - Plane



$$(\mathbf{r} - \mathbf{a}) \cdot \hat{\mathbf{n}} = 0$$

Recap of vector identities - Plane



$$\hat{\mathbf{n}} = \frac{\mathbf{b} \times \mathbf{c}}{|\mathbf{b} \times \mathbf{c}|}$$

Reason for vector notation

- Generalizable to higher dimensional spaces
- Notation and concepts will remain the same as in 2D/3D

Equation of a hyperplane

- A plane in 2D:

$$y = ax + b$$

- We can see it as

$$\mathbf{w}^T \cdot \mathbf{x} = 0; \text{where} \quad \mathbf{w} = \begin{pmatrix} 1 \\ -a \\ -b \end{pmatrix}, \mathbf{x} = \begin{pmatrix} y \\ x \\ 1 \end{pmatrix}$$

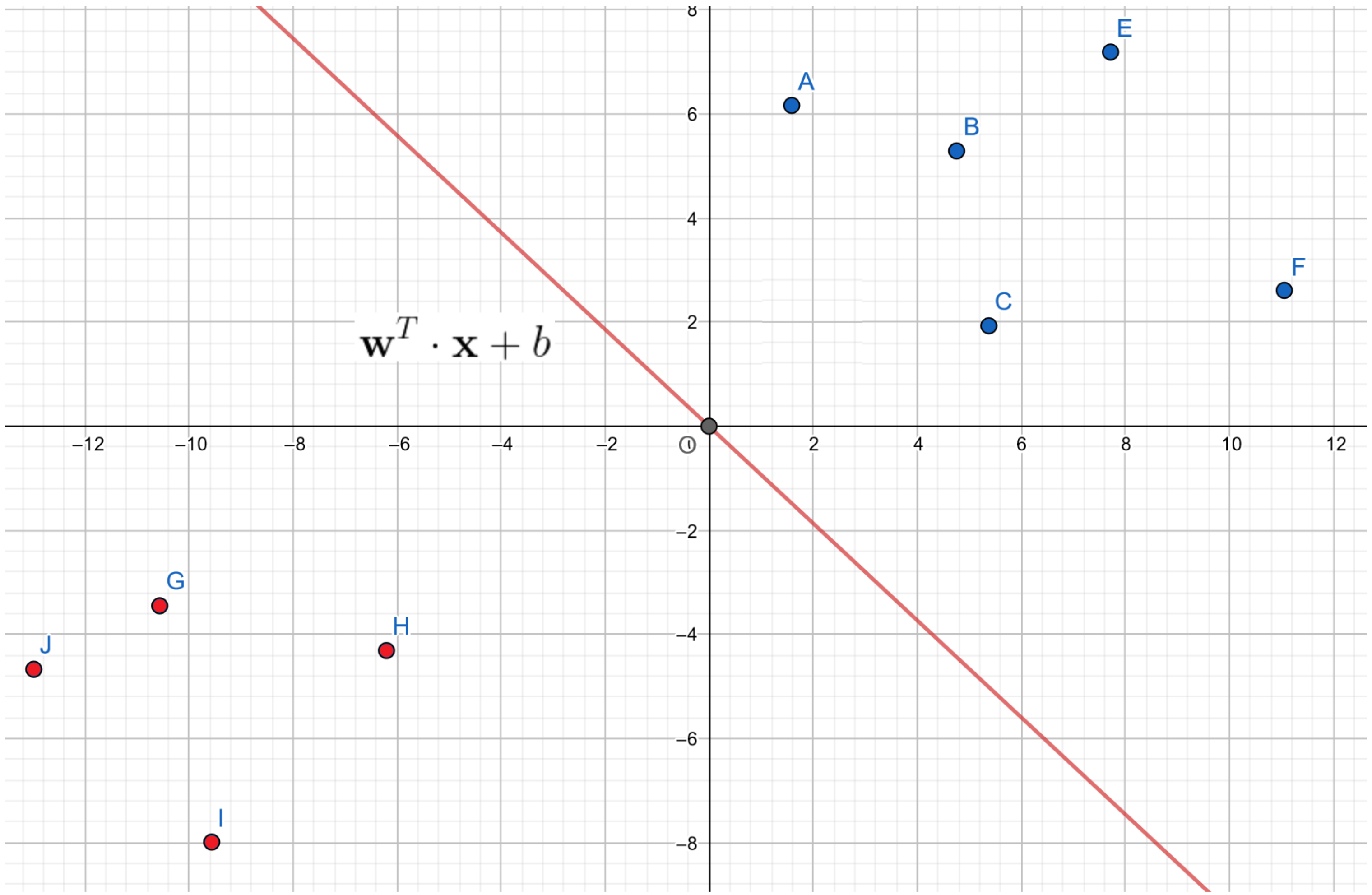
- $\widehat{\mathbf{n}} \cdot (\mathbf{x} - \mathbf{y})$ drawing analogy from the plane equation $\mathbf{w}^T$ is the ***normal***

- $\mathbf{w}^T \cdot \mathbf{x} = 0$ is called the ***hyperplane***

What is a Support Vector Machine

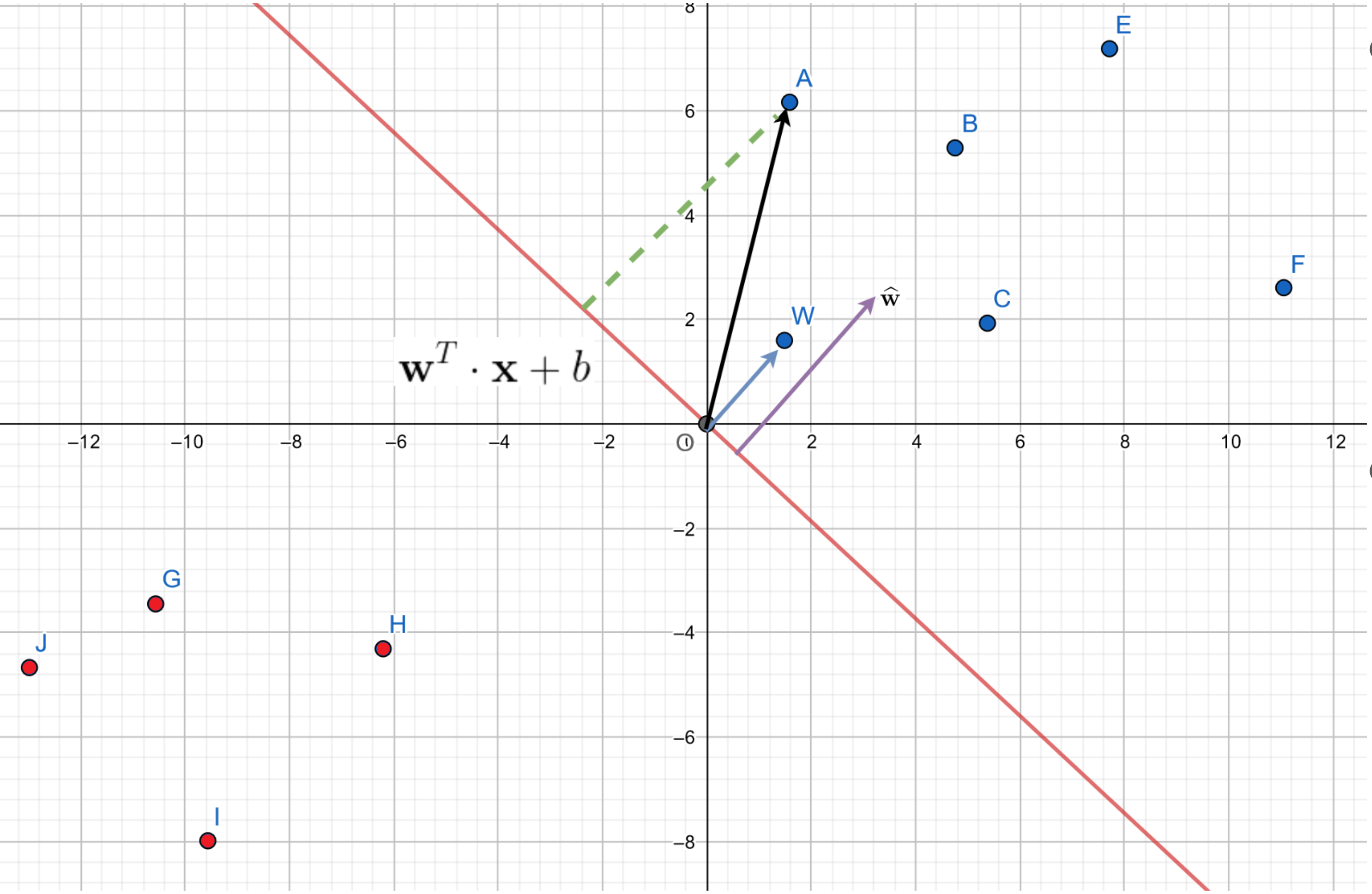
- SVM is an algorithm that can learn a linear model from a set of **labelled data** available to **train** the model. Eg (classification, regression (SVR))
- We suppose that the data we want to classify can be separated into classes by a line
- We know that a line can be represented by the equation $y = \mathbf{w}^T \cdot \mathbf{x} + b$
- We know that there is an infinity of possible lines obtained by changing the value of $\mathbf{w}$ and b
- We use an algorithm to determine which are the values of $\mathbf{w}$ and b giving the "best" line separating the data.

Distance to SVM hyperplane (1)



- Can we apply SVM here?

Distance to SVM hyperplane (2)



- Distance from point *A* to the hyperplane

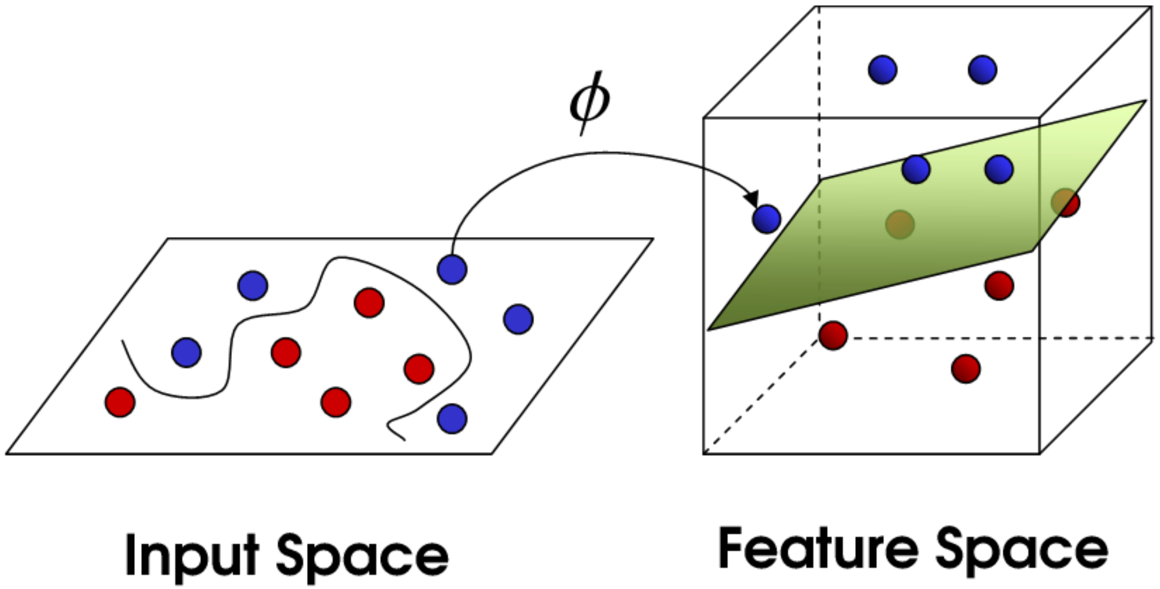
$$|\mathbf{p}| = \widehat{\mathbf{w}} \cdot \mathbf{a}$$

- We can find a hyperplane that can has the maximum margin between the two classes. (Maximal margin classifier)

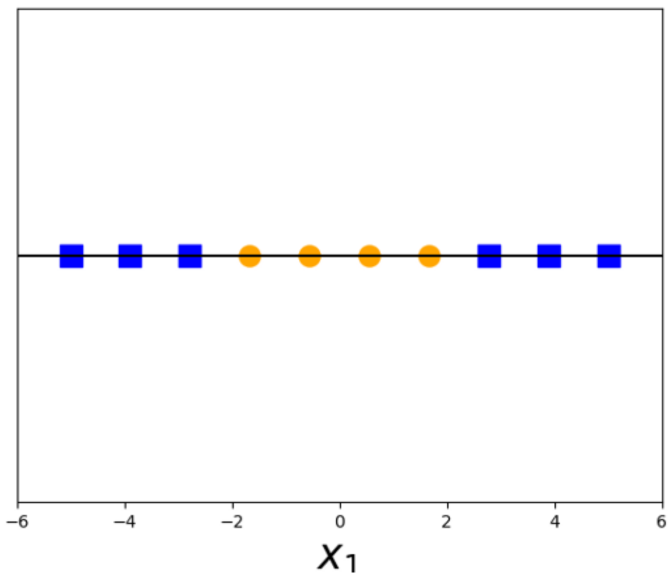
What if the data we want to classify can**not** be separated into classes by a line

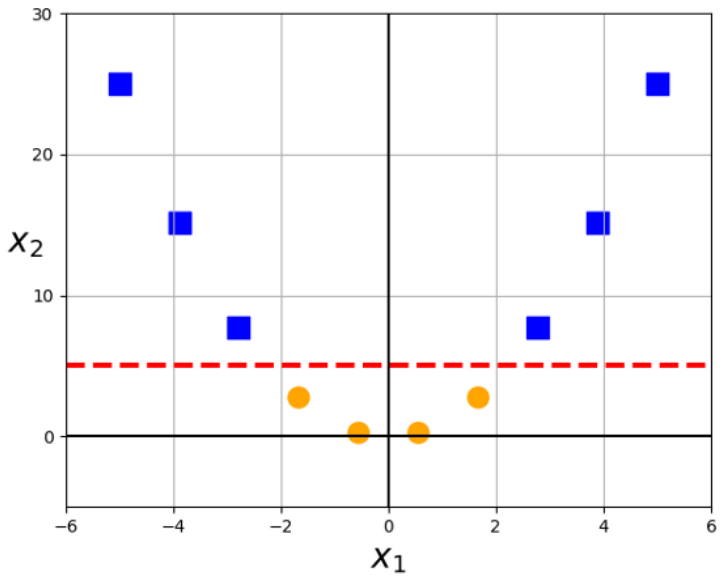
Kernel trick for SVMs

- If the data is not linearly separable in the original input, space then we apply transformations to the data, which map the data from the original space into a *higher dimensional feature space*.
- The goal is that after the transformation to the higher dimensional space, the classes are now linearly separable in this higher dimensional feature space.



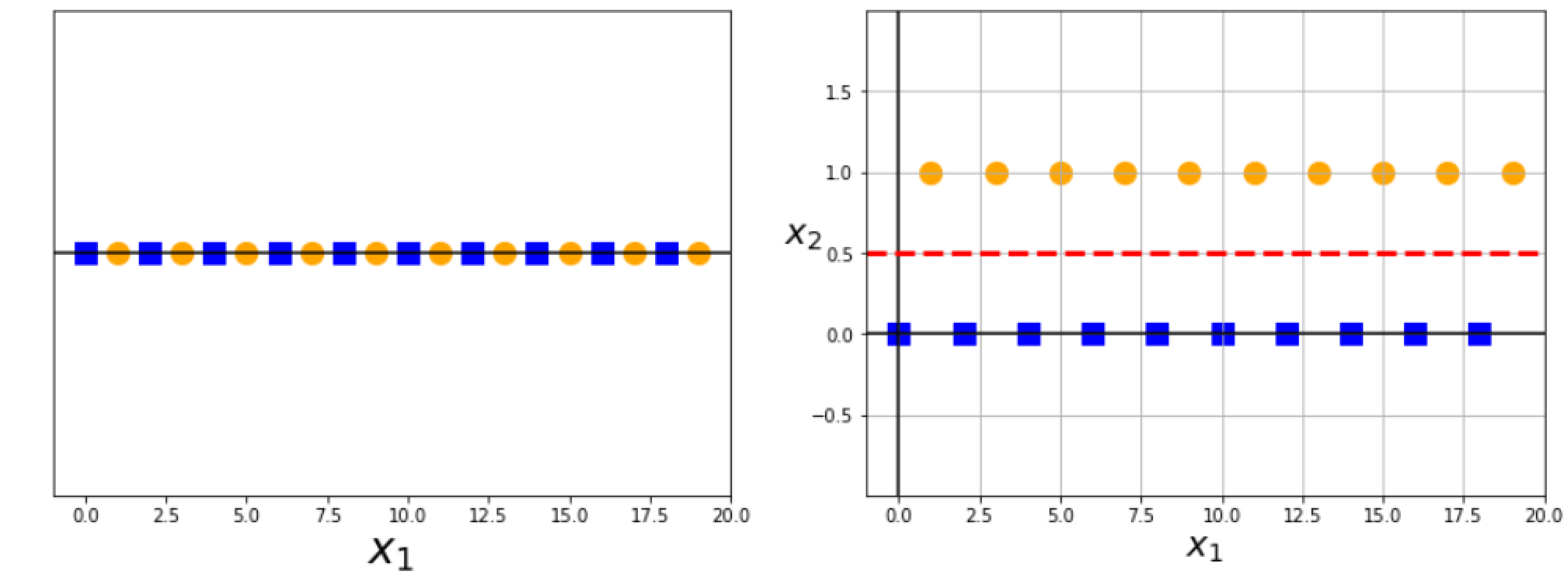
Kernel trick for SVMs (Eg1)





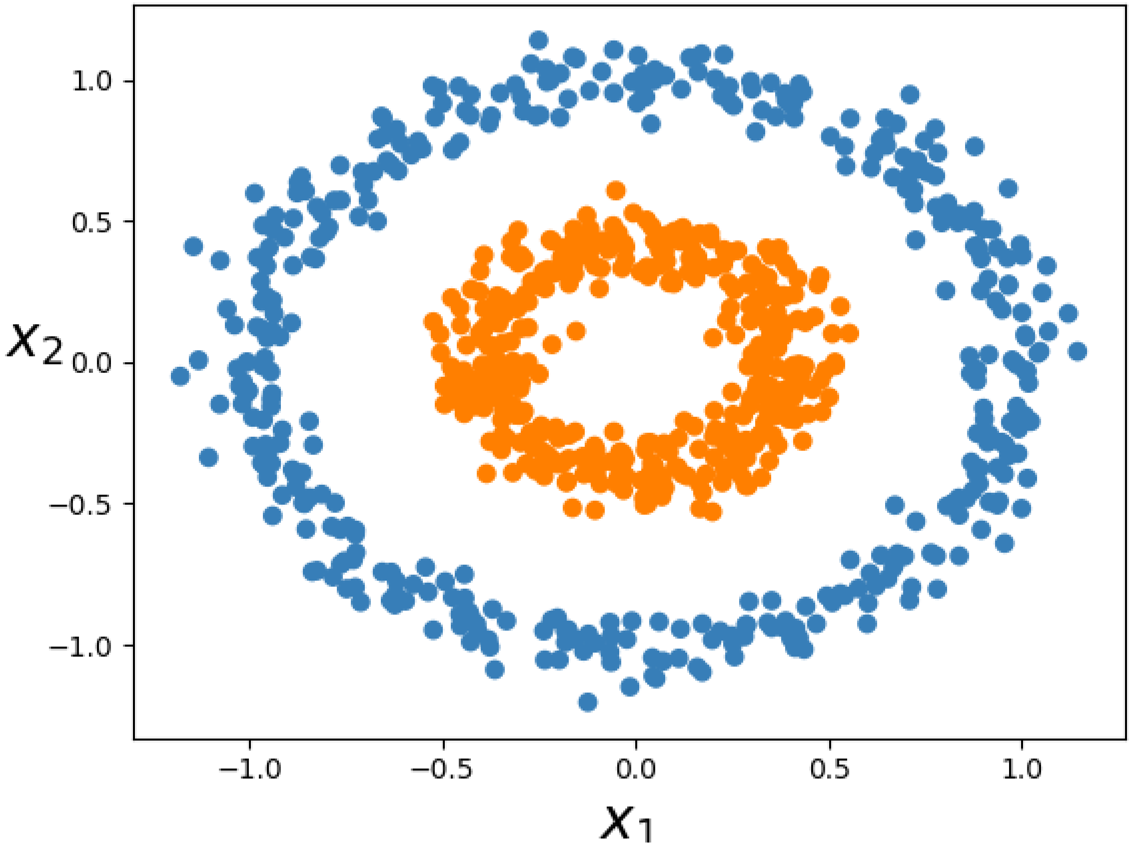
$$\phi(x) = x^2$$

Kernel trick for SVMs (Eg2)



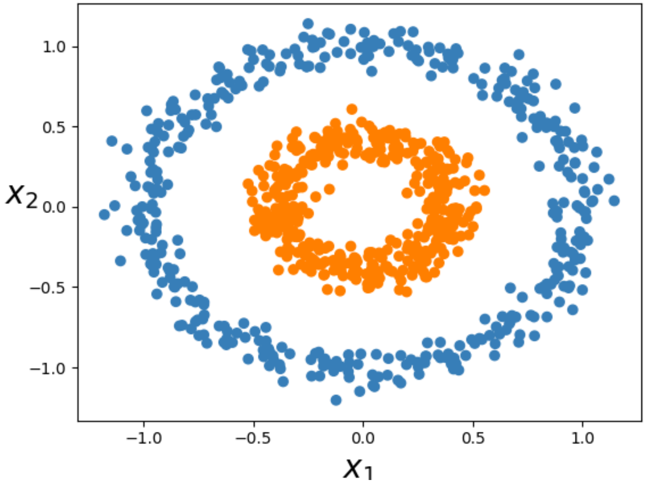
$$\phi(x) = x \bmod 2$$

Kernel trick for SVMs (Eg3)

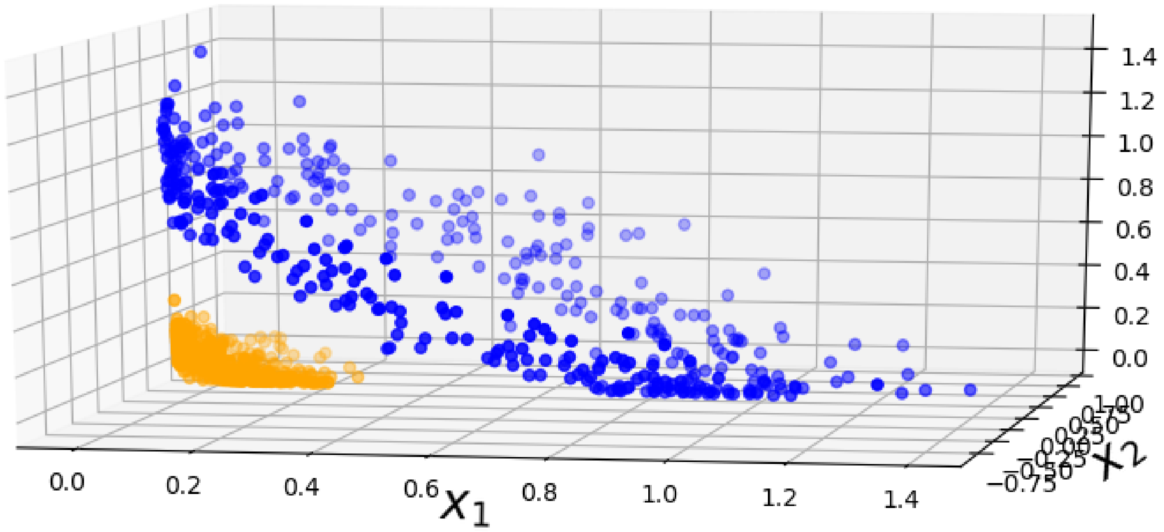


$$\phi(\mathbf{x}) = \phi\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{pmatrix}$$

Kernel trick for SVMs (Eg3)



$$\phi(\mathbf{x}) = \phi\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{pmatrix}$$



Kernel definition

- Kernel is defined as a function that acts on the input vectors in the original space and returns the **dot product** of the vectors in feature space.
- Formally

$$\mathbf{x}, \mathbf{y} \in X \text{ and } \phi : X \to \mathbb{R}^n$$
$$\text{then } k(\mathbf{x}, \mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle$$

- Let’s apply this definition to **Eg3**

Kernel definition to Eg3

- Let $\mathbf{a}, \mathbf{b} \in \mathbb{R}^2$; $\mathbf{a}^T = (a_1, a_2), \mathbf{b}^T = (b_1, b_2)$

.

$$\phi(\mathbf{x}) = \begin{pmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{pmatrix}$$

$$\langle \phi(\mathbf{a}), \phi(\mathbf{b}) \rangle = \phi(\mathbf{a})^T \cdot \phi(\mathbf{b})$$

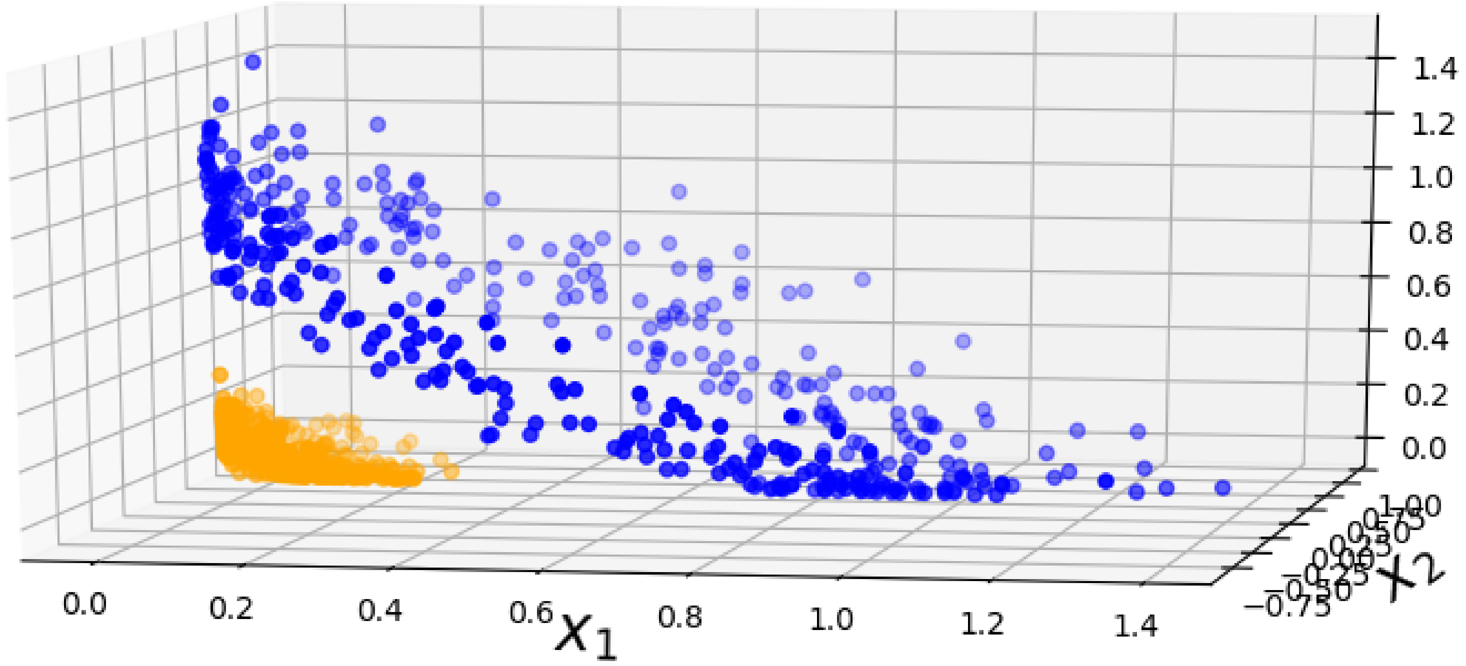
$$= \begin{pmatrix} a_1^2 & \sqrt{2}a_1a_2 & a_2^2 \end{pmatrix} \cdot \begin{pmatrix} b_1^2 \\ \sqrt{2}b_1b_2 \\ b_2^2 \end{pmatrix}$$

$$= (a_1^2b_1^2 + 2a_1b_1a_2b_2 + a_2^2b_2^2) = (a_1b_1 + a_2b_2)^2$$

$$= \left[\begin{pmatrix} a_1 & a_2 \end{pmatrix} \cdot \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \right]^2 = \left[\mathbf{a}^T \cdot \mathbf{b} \right]^2$$

Kernel trick for SVMs (Eg3)

- Squared vector norm separation



References

- https://www.svm-tutorial.com/2017/02/svms-overview-support-vector-machines/
- https://www.svm-tutorial.com/2014/11/svm-understanding-math-part-2/
- https://towardsdatascience.com/the-kernel-trick-c98cdbcaeb3f
- https://medium.com/@zxr.nju/what-is-the-kernel-trick-why-is-it-important-98a98db0961d
- https://www.baeldung.com/cs/svm-hard-margin-vs-soft-margin