Life Sciences and
Facility Management

Institute of
Computational Life Sciences

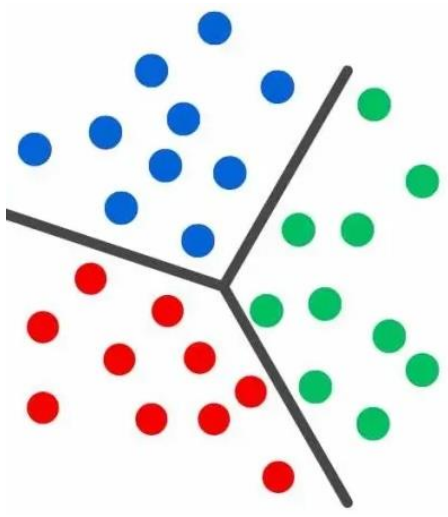
Support Vector Machines

Machine Learning

Norman Juchler

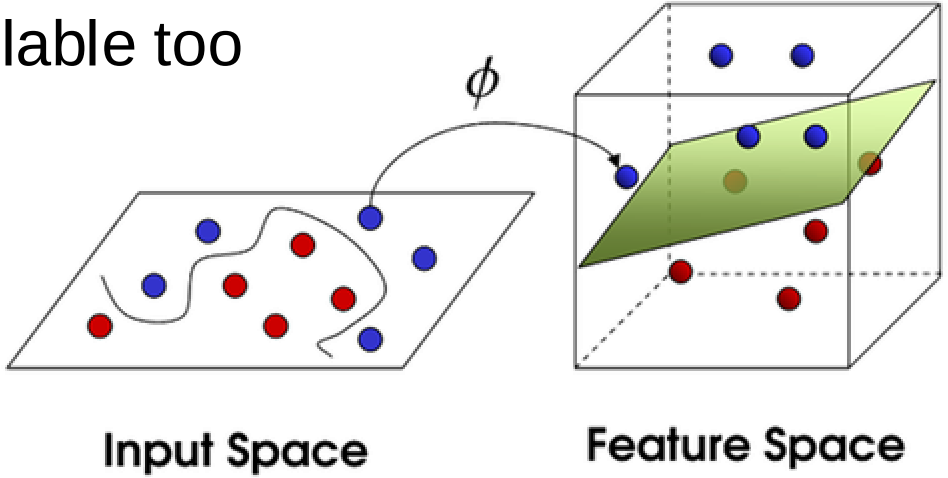
Learning objectives

- Explain the core concepts of SVMs
- Differentiate between linear and non-linear SVMs
- Be aware of its strengths and limitations



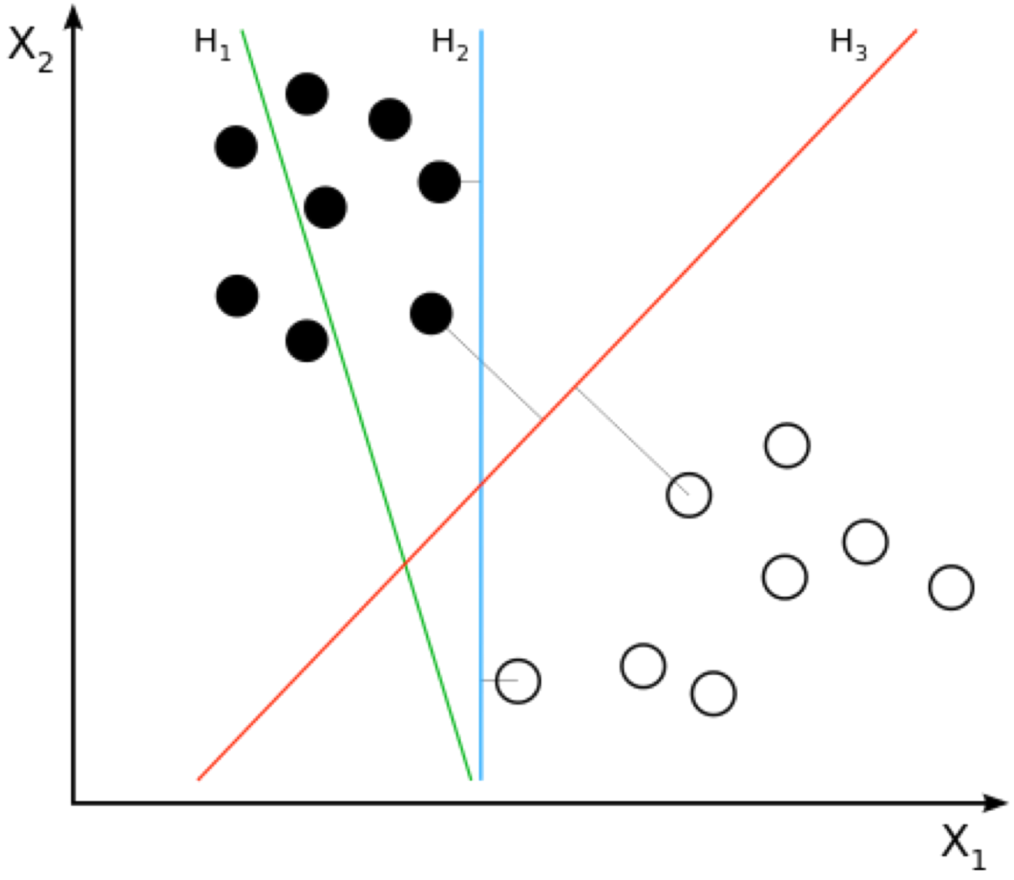
Context

- Supervised learning
  - (Mainly) classification
  - (...but also) regression
- Proposed by Vapnik and Chervonenkis in 70s/80s/90s
- Robust method with a solid theoretical foundation
- Can be seen as an extension of linear classifiers
- Basic form is linear, but nonlinear variants are available too



Motivation

Which of the classifiers H1, H2, H3 is the best?



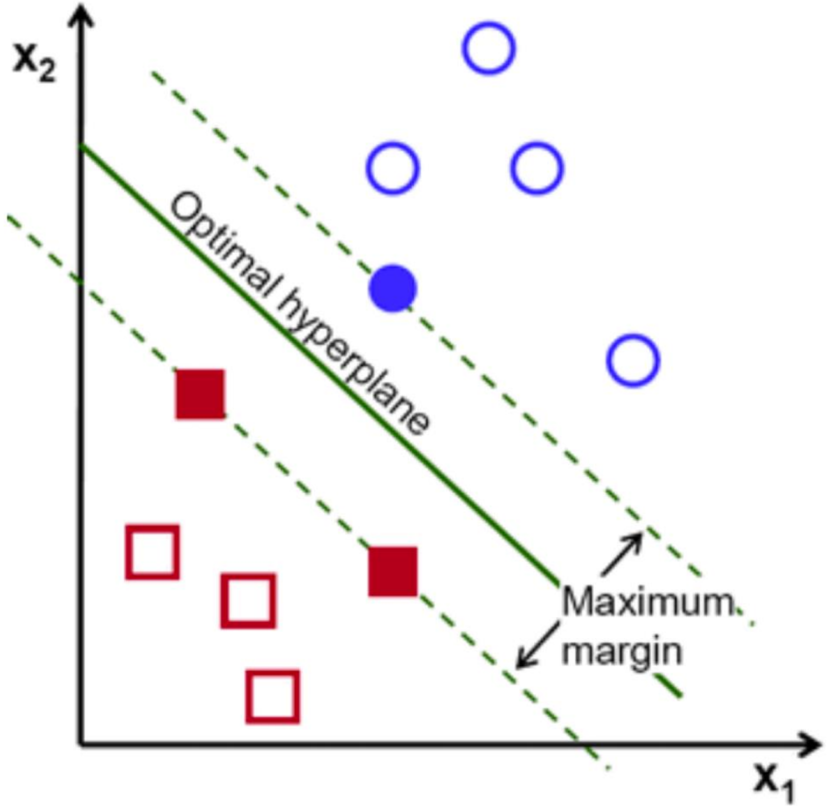
Key concepts: Support vectors and margin

- Linear classifier: Find **hyperplane** that best separates the classes in the feature space
- Any hyperplane ca be expressed as

$$\mathbf{w}^T\mathbf{x} - b = 0$$

- Geometric interpretation: **w** is the normal of the decision boundary / hyperplane!

- **<u>Definition</u>**: The nearest points of each class to a hyperplane are called **support vectors**
- **<u>Key idea</u>**: SVMs find the hyperplane with maximal distance (**margin**) to the support vectors.



Key concepts: Hard margin vs. soft margin

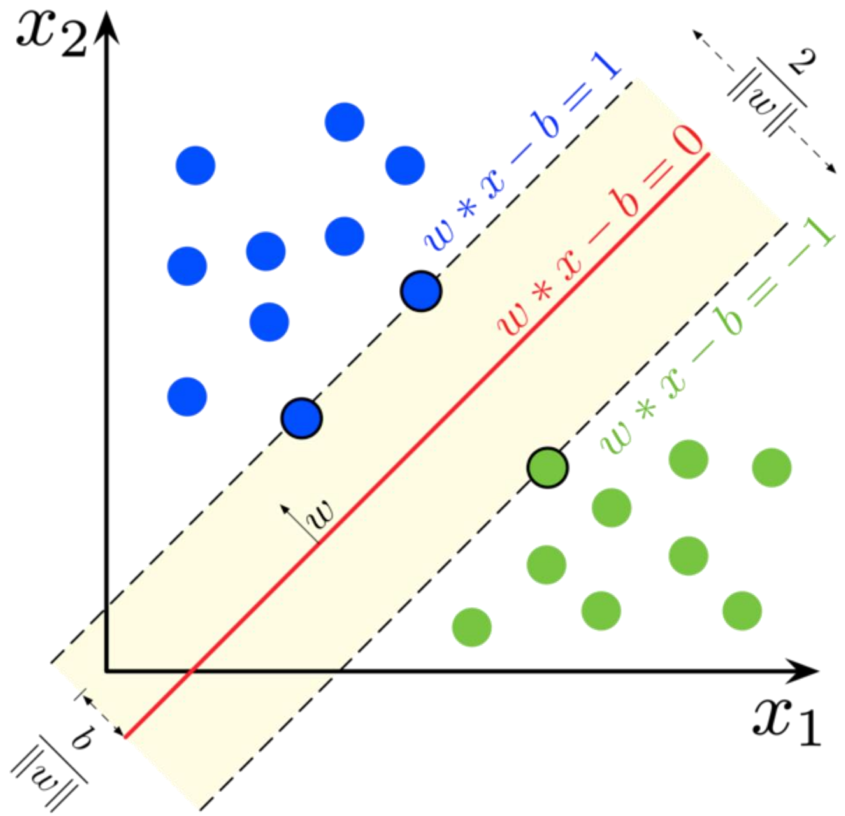
- Let's try to formulate the optimization criterion!

- **Observations**:
  - The vector **w** is responsible for the orientation of the hyperplane (it's its normal vector)
  - It turns out: the margin width is 2/||w|| (see here)
  - This will be the boundary condition:

$$\mathbf{w}^T\mathbf{x}_i - b \geq 1, \quad \text{if } y_i = c_1$$
$$\mathbf{w}^T\mathbf{x}_i - b \leq -1, \quad \text{if } y_i = c_2$$

  - Bias term b determines the position of the plane
  - Trick: Let's use here the class labels {-1, 1}! Why? Because we can rewrite this:

$$y_i(\mathbf{w}^T\mathbf{x}_i - b) \geq 1 \quad \forall i$$

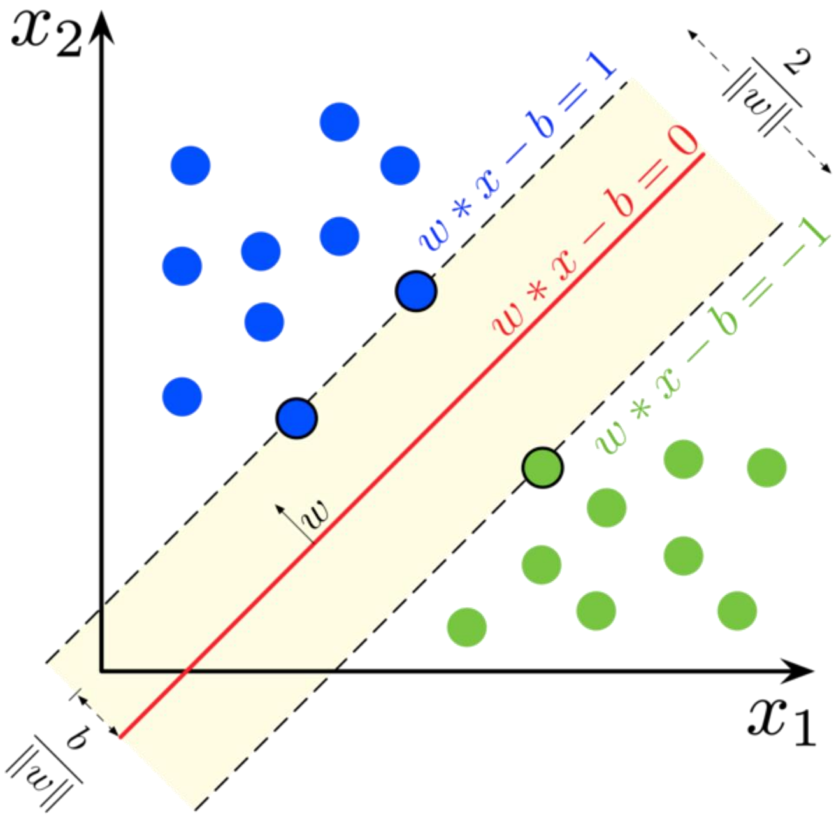


Key concepts: Hard margin vs. soft margin

- Optimization with “**hard margins**”:

$$\min_{\mathbf{w},b} \|\mathbf{w}\|^2 \quad \text{subject to:}$$

$$y_i(\mathbf{w}^T\mathbf{x}_i - b) \geq 1 \quad \forall i$$



- This criterion ensures all data points are correctly classified and lie outside the margin.
- **Problem**: This works only perfectly separable dataset!

Key concepts: Hard margin vs. soft margin

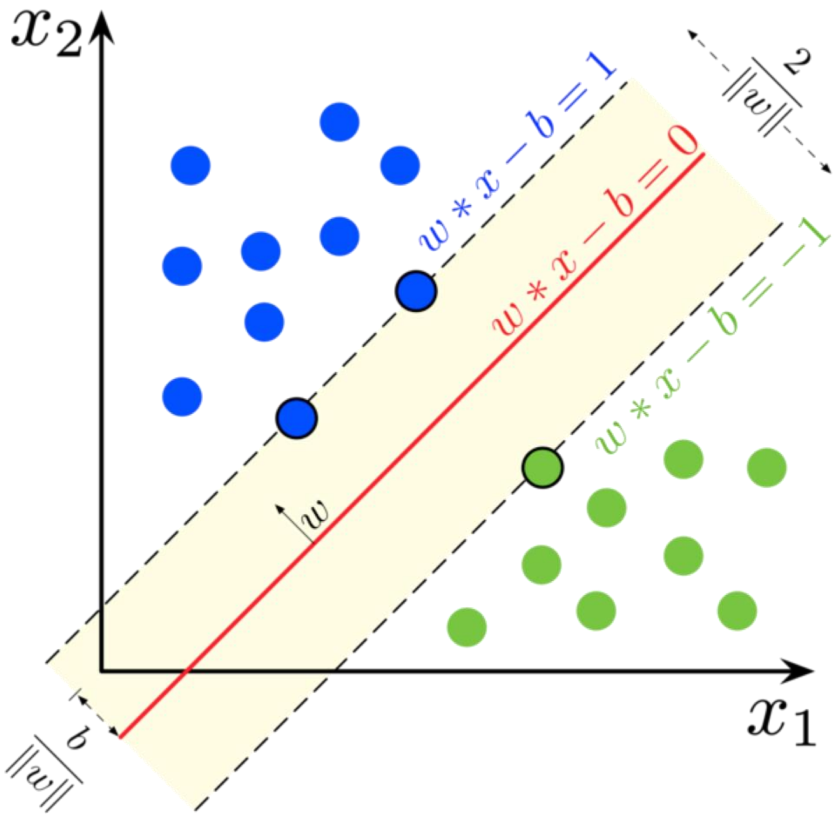
- Optimization with “**soft margins**”:

$$\min_{\mathbf{w},b,\xi} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i \text{subject to:}$$

$$y_i(\mathbf{w}^T\mathbf{x}_i - b) \geq 1 - \xi_i \quad \forall i$$

$$\xi \geq 0 \quad \forall i$$

- This formulation introduces the **slack variable** ξ_i
  - Extent to which the i-th data point violates the margin
- Optimization relies on regularization parameter C:
  - Large C: Penalizes classification errors heavily, leading to a smaller margin and less tolerance for violations.
  - Small C: Penalizes errors less, allowing for a larger margin and more violations.



Key concepts: Hard margin vs. soft margin

- Optimization with “**soft margins**”:

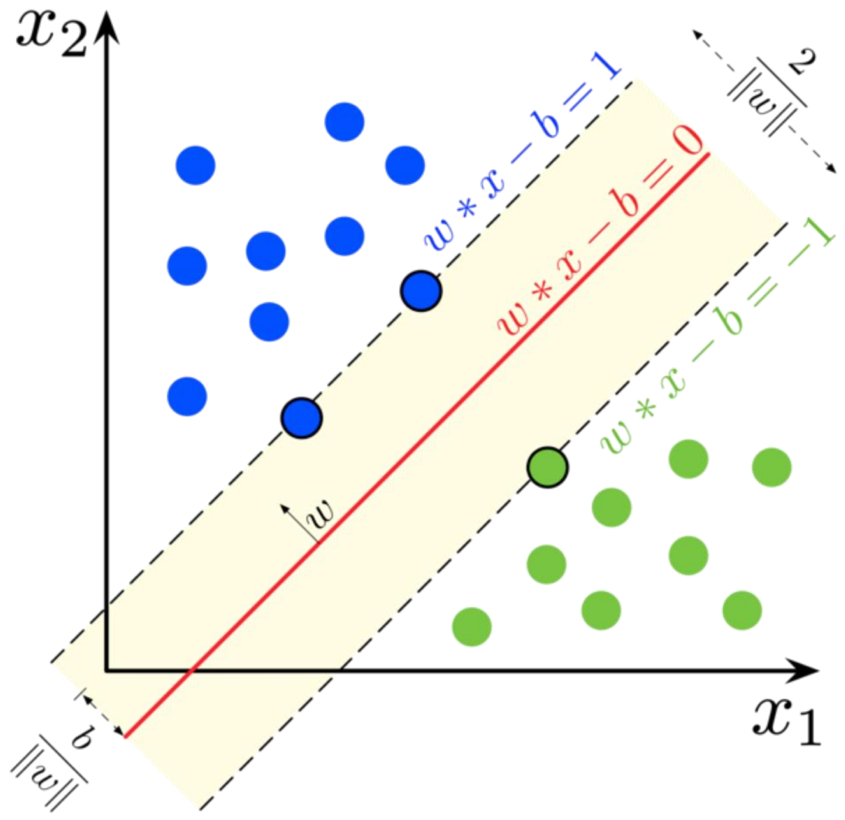
$$\min_{\mathbf{w},b,\xi} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i \text{subject to}$$
$$y_i(\mathbf{w}^T\mathbf{x}_i - b) \geq 1 - \xi_i \quad \forall i$$
$$\xi \geq 0 \quad \forall i$$

- This formulation introduces the **slack variable** ξ_i
  - Extent to which the i-th data point violates the margin
- Equivalent formulation:

$$\|\mathbf{w}\|^2 + \mathbf{C}\left[\frac{1}{n}\sum_{i=1}^{n}\max\left(0, 1 - y_i(\mathbf{w}^T\mathbf{x}_i - b)\right)\right]$$

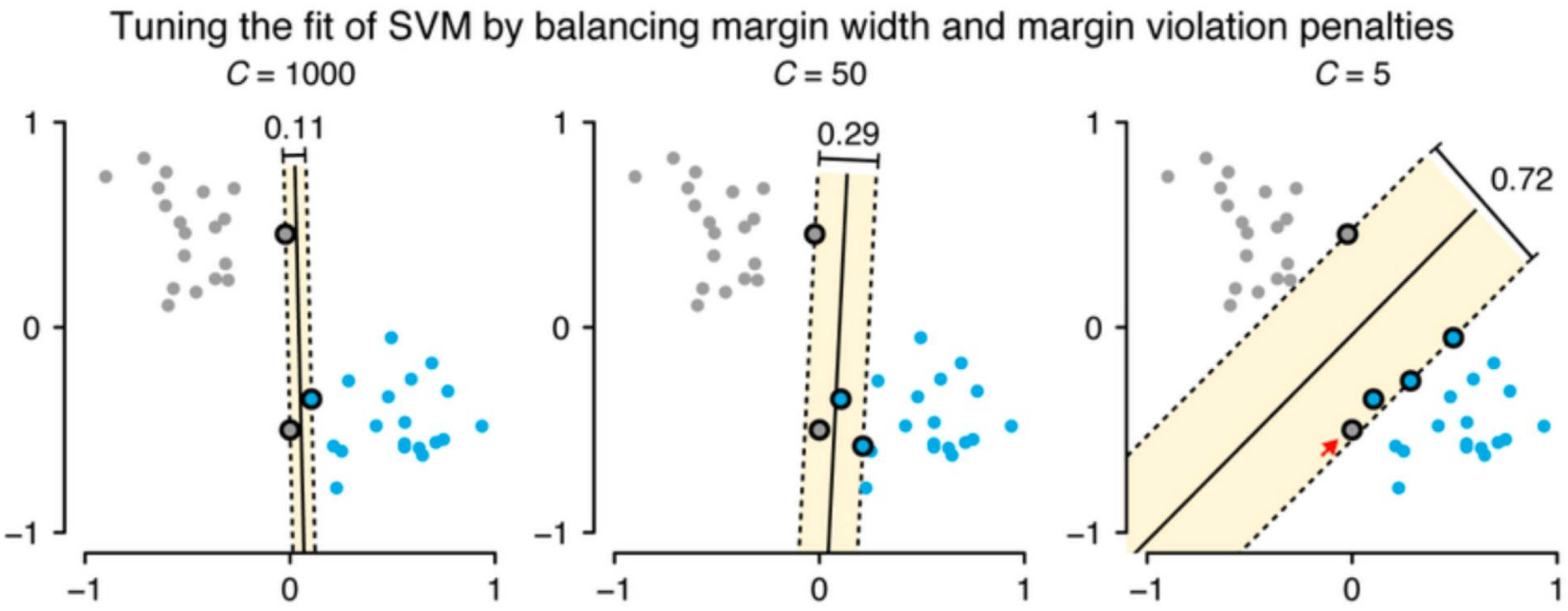
Equals 0 for all points that satisfy the hard margin constraint
Otherwise measures extend of margin violation

Weighting between hard and soft margin constraint



Working of SVM

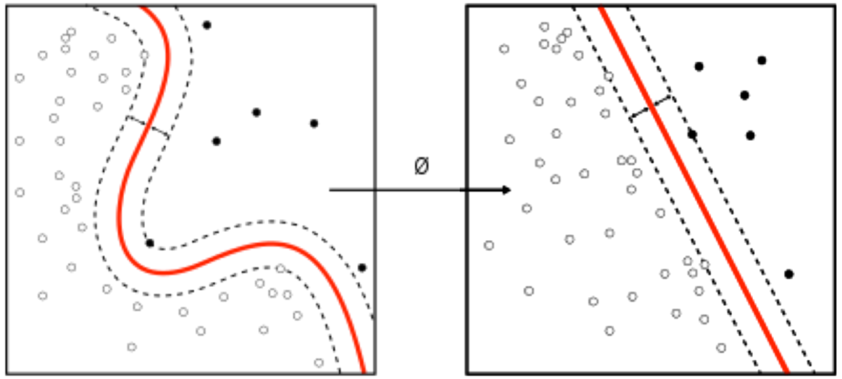
- Finding a separating line without margin-violating points is not always possible.



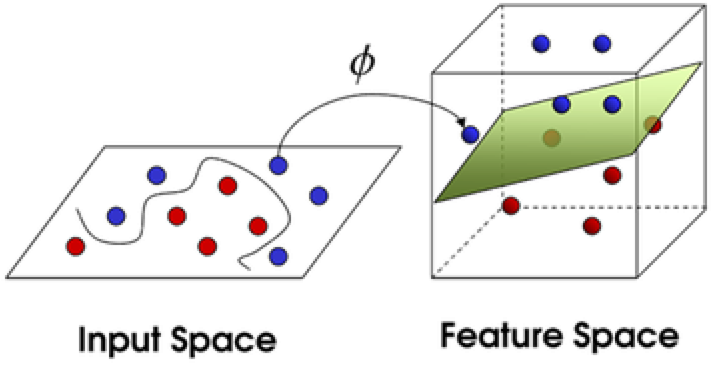
- Smaller values of C: More margin violating points, making the model more robust to outliers and increasing margin width.
- Larger values of C: Decrease the number of misclassified (training) points.

Linear versus nonlinear SVM

- Linear models
  - When applied directly in feature space, the above algorithm can only describe linear classification boundaries.



- Non-linear models
  - Idea: Transform the features X to another space first
  - Then fit the maximum-margin hyperplane in this transformed feature space.



The kernel trick

- **Idea**: It can be useful to transform the data into a higher-dimensional space!
- This can make the data better linearly separable.

$$
\begin{aligned}
\phi(\mathbf{a})^\top \phi(\mathbf{b}) &= \begin{pmatrix} a_1{}^2 \\ \sqrt{2}\, a_1 a_2 \\ a_2{}^2 \end{pmatrix}^\top \begin{pmatrix} b_1{}^2 \\ \sqrt{2}\, b_1 b_2 \\ b_2{}^2 \end{pmatrix} = a_1{}^2 b_1{}^2 + 2 a_1 b_1 a_2 b_2 + a_2{}^2 b_2{}^2 \\
&= (a_1 b_1 + a_2 b_2)^2 = \left(\begin{pmatrix} a_1 \\ a_2 \end{pmatrix}^\top \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \right)^2 = (\mathbf{a}^\top \mathbf{b})^2
\end{aligned}
$$

- The ”trick“ is that for a suitable transformation function φ, we can calculate the result of the kernel function in the original space, without having to actually perform the transformation to the higher-dimensional space.

Nonlinear classification

- Kernels enable SVMs to learn non-linear separation function.

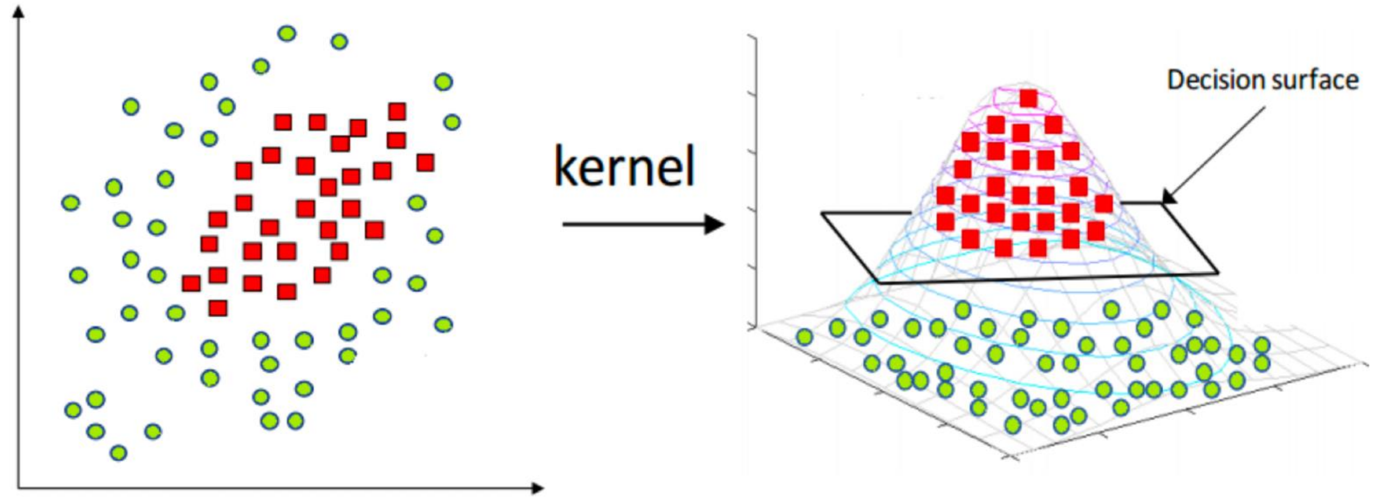
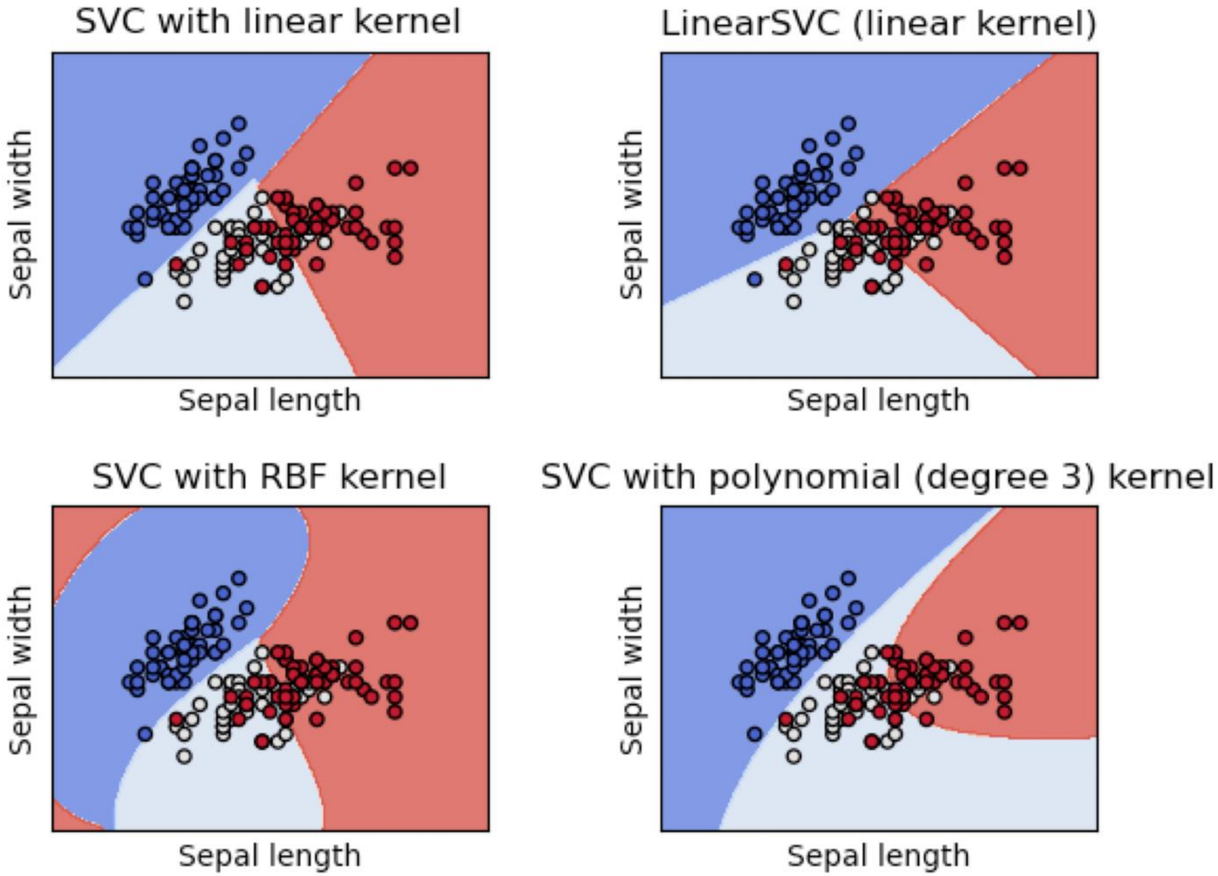


Figure 25: Example of how a transformation to a higher-dimensional space may improve data separability.

Effect of different kernels



Summary

SVMs in a nutshell

- SVMs are supervised machine learning algorithms for classification and regression tasks
- Key idea: Maximize the margin between data points of different classes to ensure better generalization.
- Basic SVMs are linear, but nonlinear cases can be handled with kernel functions
- Support vectors: Are the critical data points closest to the hyperplane
- Advantages:
  - Effective in high-dimensional spaces
  - Robust to overfitting (when properly tuned)
- Limitations:
  - Can be computationally expensive, especially with large datasets
  - Requires careful selection of hyperparameters and kernel type

Further ~~reading~~ watching

- StatQuest:
  - Support Vector Machines (main ideas) (20min)
  - Support Vector Machines (polynomial kernel) (7min)
  - Support Vector Machines (RBF kernel) (15min)
- Interactive demo: Link