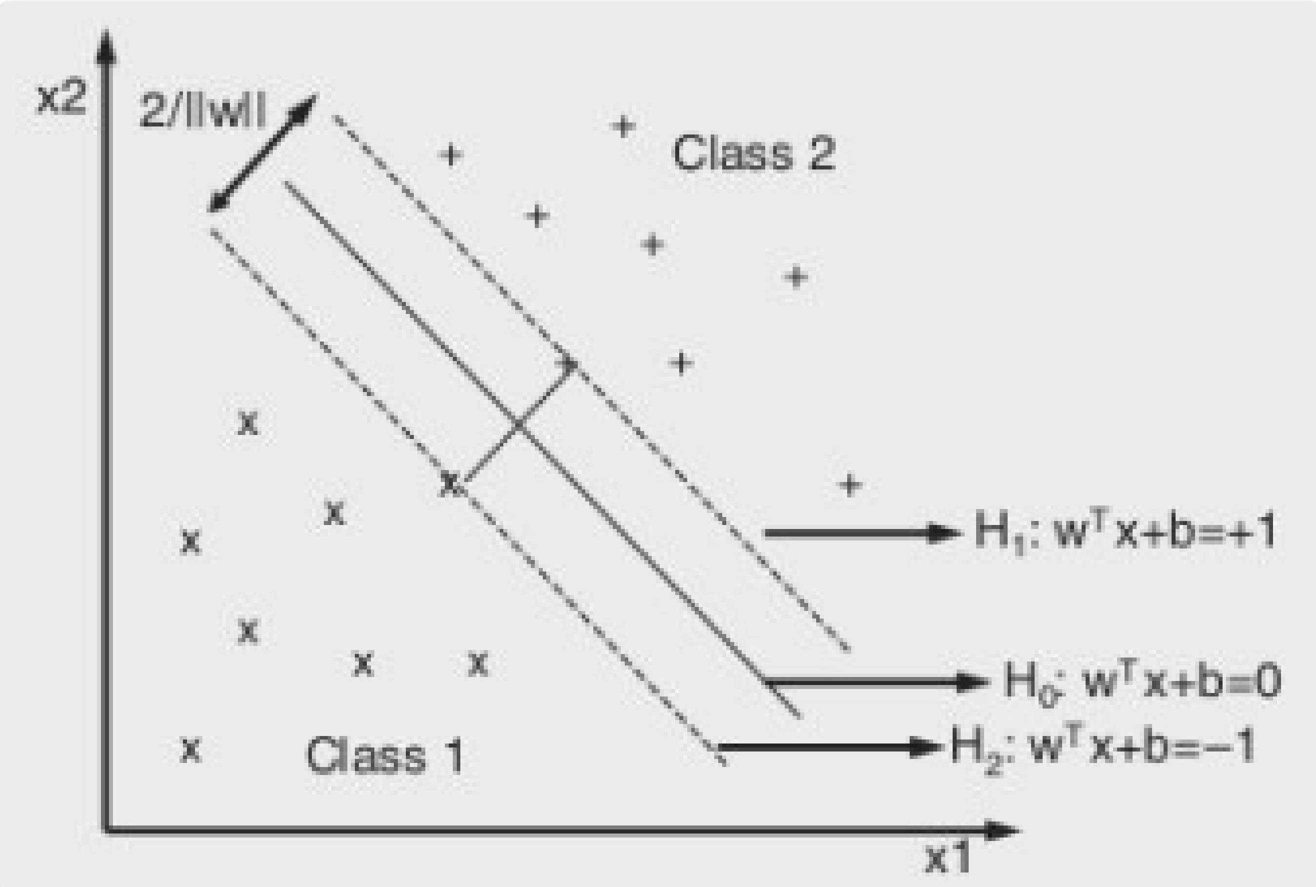
3.9 Support Vector Machine

3.9.1 Intro

> 🔑 #DEF *Support vector machines (SVMs) deal with the shortcomings of neural networks.*

The origins of classification SVMs date back to the early days of **linear programming**.

3.9.2 The Linear Separable Case #LinearSeparableCase



SVMs aim at maximizing this margin in order to pull both classes as far apart as possible.

🎯 #DEF #SupportVector **Support Vector:** Training points that lie on hyperplane H1 or hyperplane H2 in the figure.
🎯 #DEF #Hyperplane **Classification Hyperplane:** In figure, H0.

==> New observations are checked whether they are situated above H0 in which case the prediction is +1 or below (prediction –1).

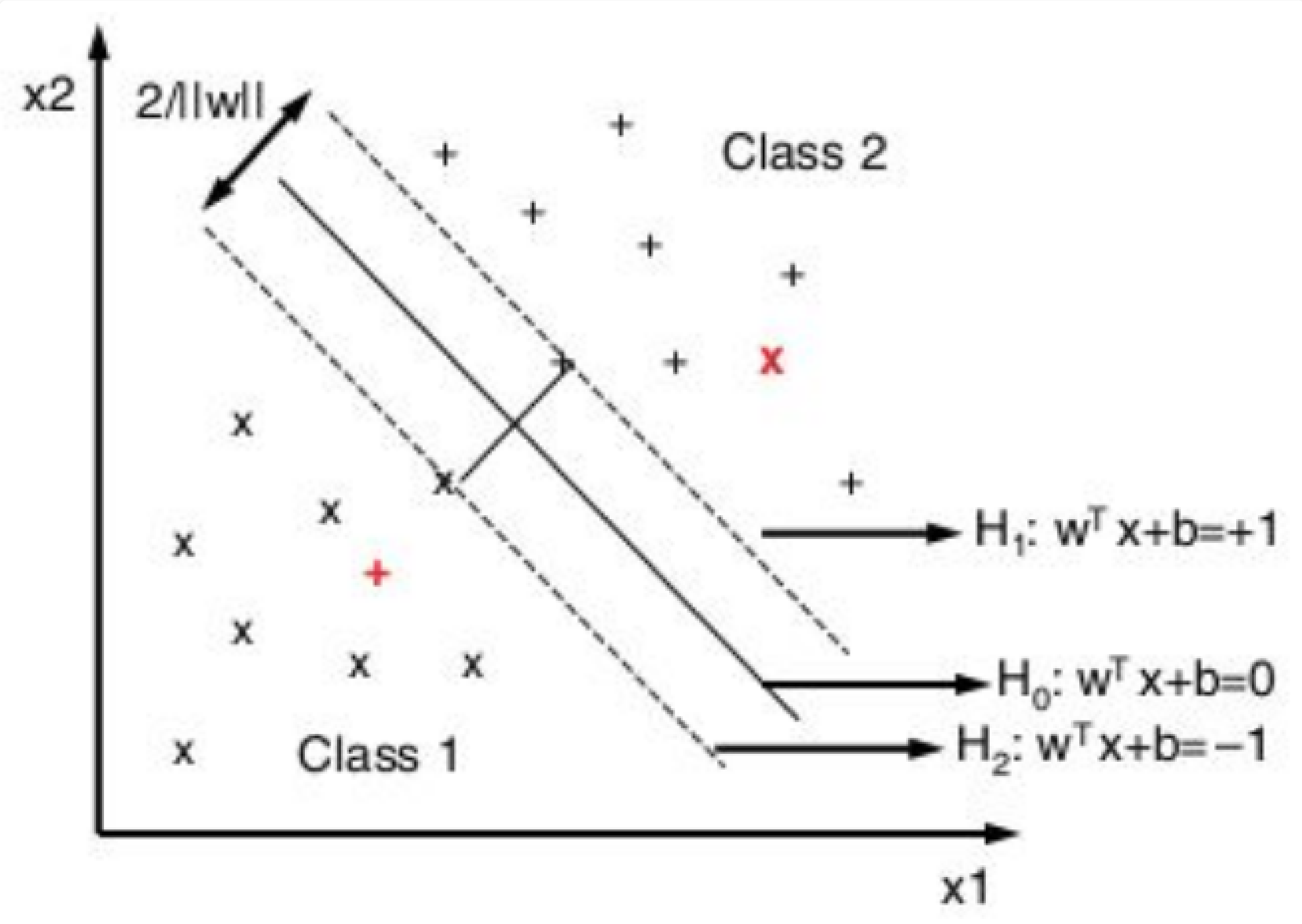
To do this we need to minimize the following linear programming problem:

$min : \frac{1}{2} \cdot \sum_{i=1}^{N} w_i^2$
$subject\ to :\ y_k(w^T x_k + b) \geq 1$, $k = 1,\ ...\ ,n$

This is a convex optimization problem with no local minima and only one global minimum.

3.9.3 The Linear Nonseparable Case #LinearNonSeparableCase

We have **overlapping class distributions** ==> the SVM classifier can be extended with error terms.



We now have to minimize the following linear programming problem:

$min : \frac{1}{2} \cdot \sum_{i=1}^{N} w_i^2 + C \cdot \sum$
$subject\ to :\ y_k(w^T x_k + b) \geq 1 - e_k$, $k = 1,\ ...\ ,n$
$e_k \geq 0$

where:

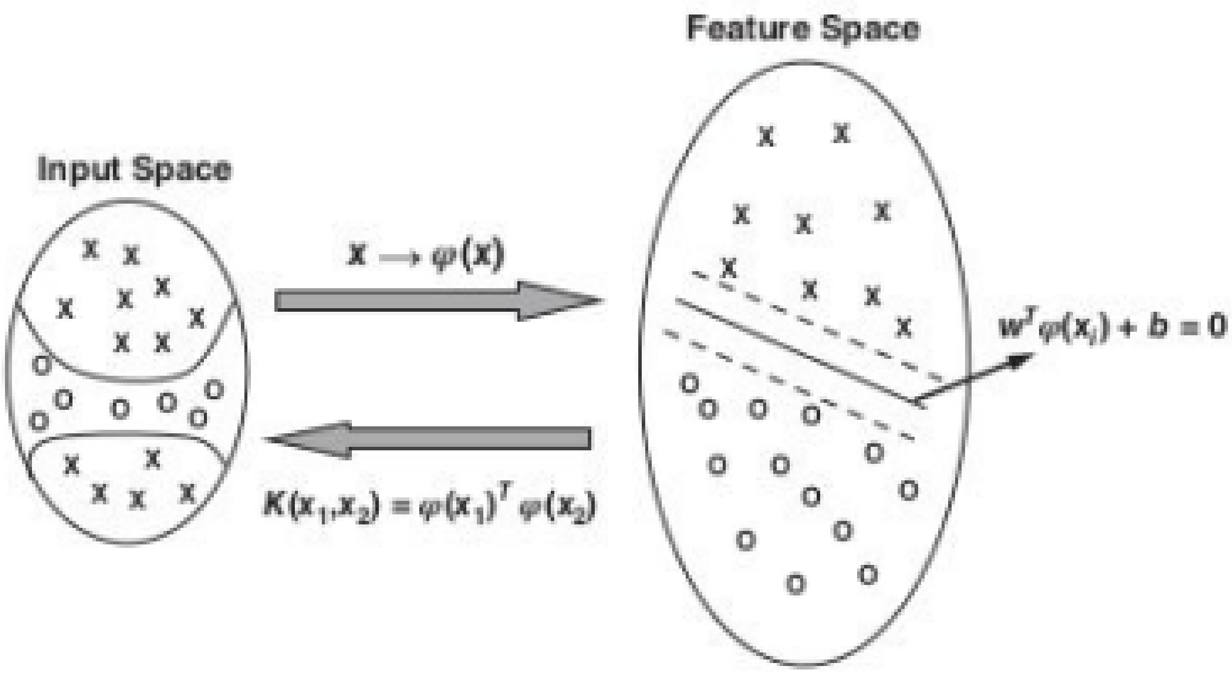
- e_k: error variables that allow misclassifications;
- C: hyperparameter in the objective function that balances the importance of maximizing the margin versus minimizing the error on the data.

> 🔑 ***Key concept:***
>
> - *A high value of C implies a higher risk of overfitting.*

> - *A low value of C implies a lower risk of overfitting.*

3.9.4 The Nonlinear SVM Classifier #NonlinearSVMClassifier

Nonlinear SVM classifier will map the input data to a higher dimensional feature space using some mapping $\varphi(x)$.



3.9.5 Tuning of the Hyperparameters #HyperparameterTuning

Five steps:

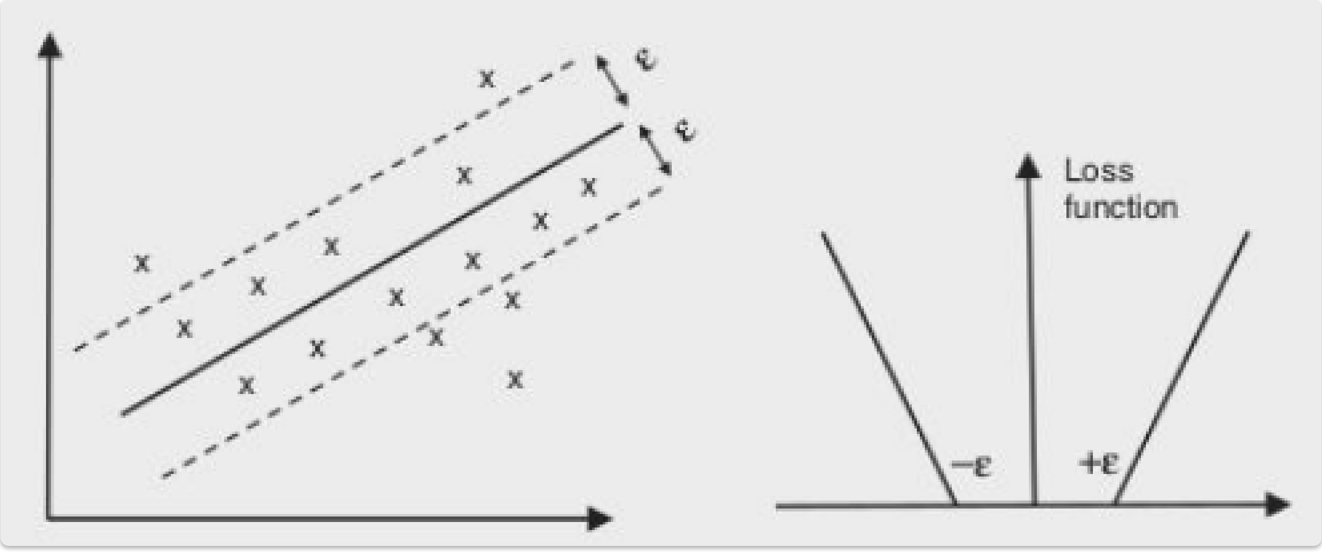
1. Partition the data into 40% training, 30% validation and 30% test data;
2. Build a SVM classifier for each parameter combination from the set;
3. Choose the combination with the best validation set performance;
4. Build a SVM classifier with the optimal parameter combination on *combined training + validation dataset*;
5. Calculate the performance of the estimated classifier on the test set.

3.9.6 SVMs for Regression #SVMRegression

SVMs can also be used for regression applications with a continuous target.

This is done by finding a function $f(x)$, as flat as possible, which has at most ε deviation from the actual targets. The loss function:

- Tolerates errors less than ε.
- Penalizes errors higher than ε.



3.9.7 Transparency #SVMTransparency

A black-box approach can be complex in settings where interpretability is important.

SVMs have a **universal approximation property**:

- They do not require tuning of the number of hidden neurons
- Are characterized by convex optimization.

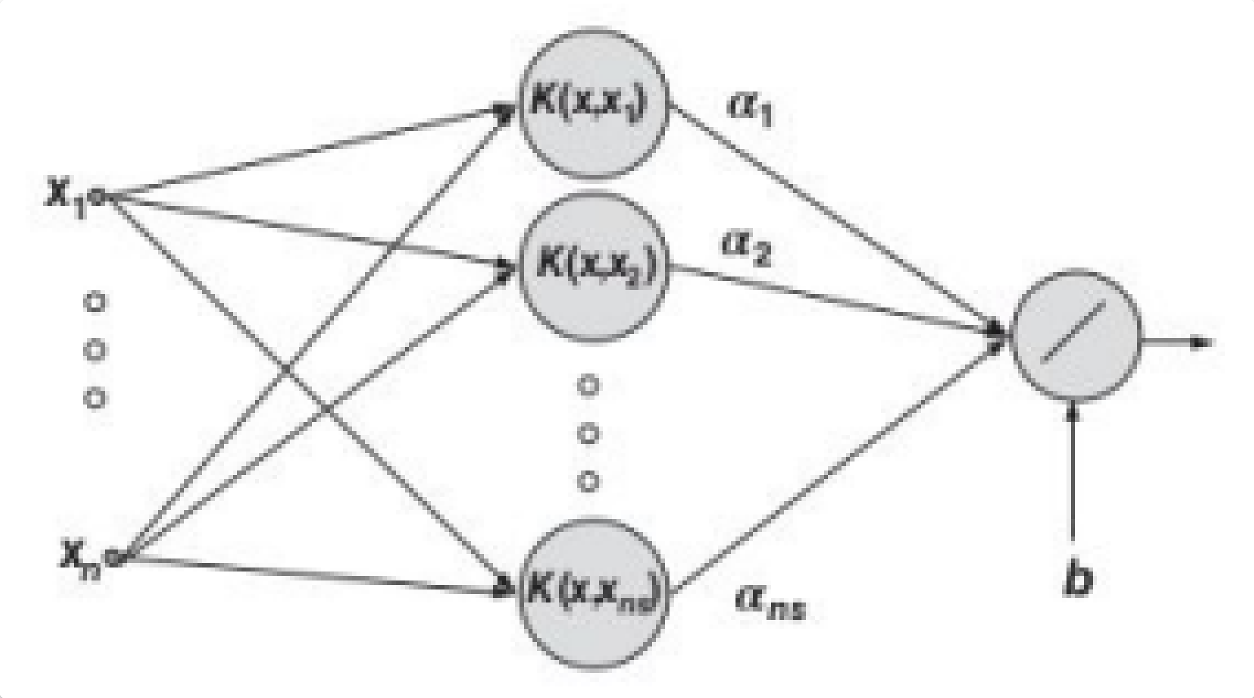
Variable selection can be performed using the backward variable selection procedure:

- **Advantage:**
  🟢 Reduces the number of variables;
- **Disadvantage:**
  🔴 Does not provide any additional insight into the workings of the SVM.

3.9.7.1 Rule-Based Approach #RuleBasedApproach

In order to better understand the inner workings of the SVM, a rule-based approach can be utilized (decompositional. We have that:

1. The SVM can be represented as a neural network:



2. The hidden layer uses kernel activation functions:
==> The number of hidden neurons now corresponds to the number of support vectors and follows automatically from the optimization.
3. The output layer uses a linear activation function.

3.9.7.2 Pedagogical Approach #PedagogicalApproach

To understand the inner workings of the SVM, a pedagogical approach can be also used. It can be easily combined with SVMs since it considers the underlying model as a black box.

It works like this:

1. SVM is first used to construct a data set with SVM predictions for each of the observations;
2. This data set is then given to a decision tree algorithm to build a decision tree;
3. Additional training set observations can be generated to facilitate the tree construction process.

3.9.7.3 Two-Stage Models #TwoStageModels

Two-stage Models can be used to provide more comprehensibility.
A simple model (e.g., linear or logistic regression) is estimated first, followed by an SVM to correct the errors of the latter.

Next chapter: Ensemble Methods