Support Vector Machines

Joseph M. Ingenito

University of Denver

Winter 2023

Table of Contents

Supervised Learning Models

Definition

Supervised Learning is defined by it's use of labeled datasets that train alogrithms to classify data or predict outcomes accurately.

Supervised Learning Models

Definition

Supervised Learning is defined by it's use of labeled datasets that train alogrithms to classify data or predict outcomes accurately.

Types of Models:

- Neural Networks
- Naive Bayes Classifiers
- K-Nearest Neighbors
- Regression
- Support Vector Machines (SVM)

Neural Network Comparison

Advantages:

- We can constrain the size of the network and number of layers, controlling the dimensionality of the model.
- Calculate predictions very quickly since the number of matrix multiplications is fixed by the number of layers.

Neural Network Comparison

Advantages:

- We can constrain the size of the network and number of layers, controlling the dimensionality of the model.
- Calculate predictions very quickly since the number of matrix multiplications is fixed by the number of layers.

Disadvantages:

- Longer training time.
- Non-guaranteed convergence due to local minima.
- Fixed size (now a disadvantage), since in the real world the actual problem could be more complex than anticipated.

SVM Goals

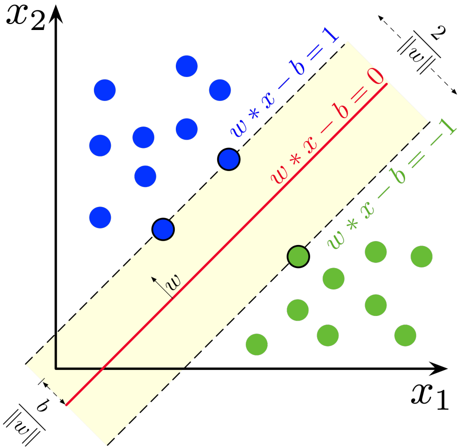


Figure: Goal of the SVM

Widest Street Approach

Goal: Maximize the width of the street $\frac{2}{||W||}$, or equivalently, minimize the ℓ^2-norm $||W||$.

Widest Street Approach

Goal: Maximize the width of the street $\frac{2}{||W||}$, or equivalently, minimize the ℓ^2-norm $||W||$.

Mathematical Conveniences:

- Minimizing $||W||$ is equivalent to minimizing $\frac{1}{2}||W||^2$.
- Introduce a new variable $y_i = \begin{cases} 1, \text{if } W \cdot X_i + b \geq 1 \\ -1, \text{if } W \cdot X_i + b \leq -1. \end{cases}$

Widest Street Approach

Goal: Maximize the width of the street $\frac{2}{||W||}$, or equivalently, minimize the ℓ^2-norm $||W||$.

Mathematical Conveniences:

- Minimizing $||W||$ is equivalent to minimizing $\frac{1}{2}||W||^2$.
- Introduce a new variable $y_i = \begin{cases} 1, \text{if } W \cdot X_i + b \geq 1 \\ -1, \text{if } W \cdot X_i + b \leq -1. \end{cases}$

Combine constraints into one condition:

$$y_i(W \cdot X_i + b) \geq 1 \iff y_i(W \cdot X_i + b) - 1 \geq 0.$$

Lagrange Mulitpliers

Goal from Lagrange: Maximize over all α_i,

$$\mathcal{L}(X) = \frac{1}{2}||W||^2 - \sum_{i=1}^{N} \alpha_i (y_i(W \cdot X_i + b) - 1)$$

Lagrange Mulitpliers

Goal from Lagrange: Maximize over all α_i,

$$\mathcal{L}(X) = \frac{1}{2}||W||^2 - \sum_{i=1}^{N} \alpha_i(y_i(W \cdot X_i + b) - 1)$$

Take partial derivatives:

$$\frac{\partial}{\partial W}\mathcal{L} = W - \sum_i \alpha_i y_i X_i = 0 \iff W = \sum_i \alpha_i y_i X_i$$

$$\frac{\partial}{\partial b}\mathcal{L} = -\sum_i \alpha_i y_i = 0 \iff \sum_i \alpha_i y_i = 0.$$

Lagrange Continued

Represent the optimization problem in terms of the dot product of input vectors:

$$\begin{aligned}
\max \mathcal{L} &= \frac{1}{2}||W||^2 - \sum_i \alpha_i (y_i(W \cdot X_i + b) - 1) \\
&= \frac{1}{2} W^T W - W^T \sum_i \alpha_i y_i X_i - b \sum_i \alpha_i y_i + \sum_i \alpha_i \\
&= \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j X_i \cdot X_j
\end{aligned}$$

Lagrange Continued

Represent the optimization problem in terms of the dot product of input vectors:

$$\begin{aligned}
\max\mathcal{L} &= \frac{1}{2}||W||^2 - \sum_i \alpha_i(y_i(W \cdot X_i + b) - 1) \\
&= \frac{1}{2}W^TW - W^T\sum_i \alpha_i y_i X_i - b\sum_i \alpha_i y_i + \sum_i \alpha_i \\
&= \sum_i \alpha_i - \frac{1}{2}\sum_i\sum_j \alpha_i\alpha_j y_i y_j X_i \cdot X_j
\end{aligned}$$

Decision Rule: Given a vector U, we classify U according to the rule,

$$\text{sign}(W \cdot U + b) = \text{sign}(\sum_i \alpha_i y_i X_i \cdot U + b)$$

Support Vectors

Still need to optimize

$$\sum_i \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j X_i \cdot X_j$$

Support Vectors

Still need to optimize

$$\sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j X_i \cdot X_j$$

This is a convex optimization problem, thus no risk of local maxima.
"Sequential Minimal Optimization" is used in practice.

Support Vectors

Still need to optimize

$$\sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j X_i \cdot X_j$$

This is a convex optimization problem, thus no risk of local maxima. "Sequential Minimal Optimization" is used in practice.

Result: most $\alpha_i = 0$ except for a small amount, referred to as the "support vectors". The decision rule simplifies:

$$\text{sign}\left(\sum_{i \in SV} \alpha_i y_i X_i \cdot U + b\right).$$

Non-Linearly Seperable Case

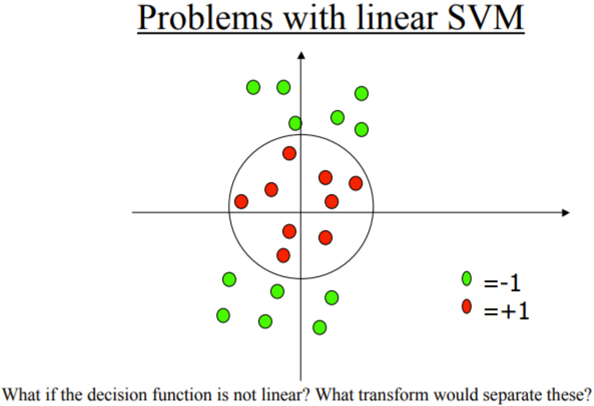


Figure: Non-linearly separable data.

Project to Higher Dimensions

Luckily we only need to define the dot product in the higher dimensional space!

Project to Higher Dimensions

Luckily we only need to define the dot product in the higher dimensional space!

Suppose φ is some projection, then we just need a function K such that $K(X, Y) = \varphi(X) \cdot \varphi(Y)$, called the Kernel.

Project to Higher Dimensions

Luckily we only need to define the dot product in the higher dimensional space!

Suppose φ is some projection, then we just need a function K such that $K(X, Y) = \varphi(X) \cdot \varphi(Y)$, called the Kernel.

We really only care about the Kernel since

$$\text{sign}\left(\sum_{i \in SV} \alpha_i y_i K(X_i, U) + b\right),$$

is the new decision rule.

Popular Kernels

Linear:

$$K(X, Y) = X \cdot Y + 1$$

Popular Kernels

Linear:

$$K(X, Y) = X \cdot Y + 1$$

Polynomial:

$$K(X, Y) = (X \cdot Y + 1)^d$$

Popular Kernels

Linear:

$$K(X, Y) = X \cdot Y + 1$$

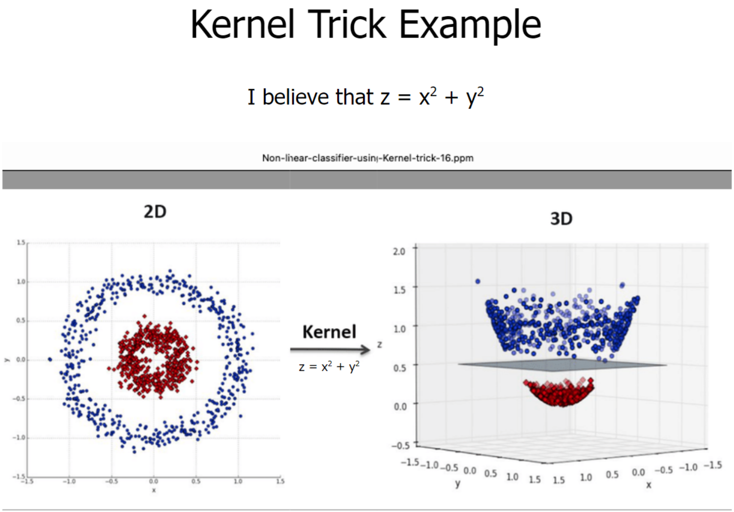
Polynomial:

$$K(X, Y) = (X \cdot Y + 1)^d$$

Radial Basis Function (Standard in Practice):

$$K(X, Y) = \exp\left(-\frac{||X - Y||^2}{2\sigma^2}\right) = \exp(-\gamma||X - Y||^2)$$

Kernel Trick Visualized



Jos Luis Rojo-lvarez; Manel Martnez-Ramn; Jordi Muoz-Mar; Gustau Camps-Valls, "Support Vector Machine and Kernel Classification Algorithms," in Digital Signal Processing with Kernel Methods , , IEEE, 2018, pp.433-502.