Support Vector Machines

MATH4069 Group Project - presentation
8th Dec 2023

Group G:

Jake Dorman, Timi Folaranmi, Rishabh Agarwal, Anas Almhmadi

Overview

In this talk, we will cover:

- The mathematical intuition behind SVMs,
- Using kernels in SVMs to tackle nonlinear data,
- How SVMs compare to other models.

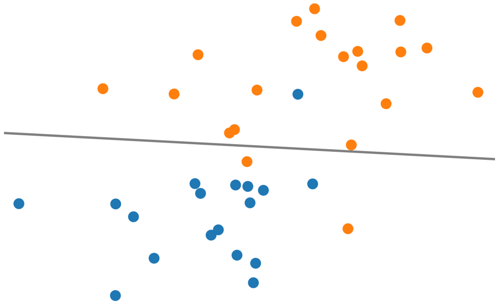
By the end of this talk, you should know:

- What type of problems SVMs can be used to solve,
- That SVMs work by maximising the margin,
- What kernels are, and how they can be used with SVMs to deal with more complex data.

What is the problem?

SVMs are used to solve **binary classification problems**:

- Input data points: $\mathbf{x}^{(i)} = (x_1^{(i)}, \ldots x_p^{(i)})$,
- Labels: $y^{(i)} \in \{-1, 1\}$.



We aim to build a **hyperplane** that best separates the data.

The Fully Separable Case

The equation for a hyperplane is

$$\boldsymbol{\theta}^\top \mathbf{x} + \theta_0 = 0.$$

The optimal hyperplane maximises the distance m to the nearest points, known as the **margin**.

$$\max_{\boldsymbol{\theta}, \theta_0} m.$$

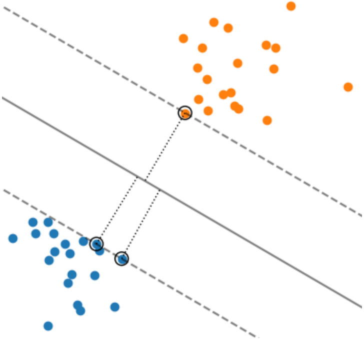


Figure: SVM hyperplane separating fully separable data.

The Fully Separable Case

The equation for a hyperplane is

$$\boldsymbol{\theta}^\top \mathbf{x} + \theta_0 = 0.$$

The optimal hyperplane maximises the distance m to the nearest points, known as the **margin**.

$$\max_{\boldsymbol{\theta}, \theta_0} m.$$

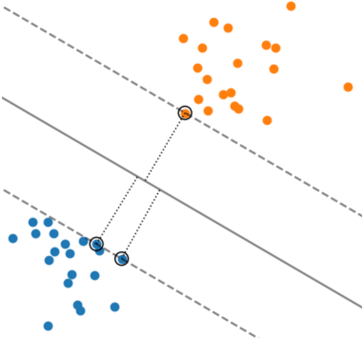


Figure: SVM hyperplane separating fully separable data.

The optimal hyperplane only depends on the points on the margin, known as the **support vectors**.

The Non-Separable Case

Introduce **slack variables** $\xi^{(i)}$ that determine how much point i is over the margin.

We now want to **maximise** the margin and **minimise** the slack, or:

$$\min_{\boldsymbol{\theta},\theta_0} \left[\frac{1}{m} + C \sum_{i=1}^{N} \xi^{(i)} \right]$$

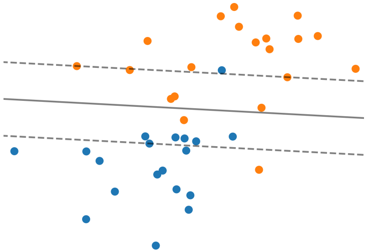


Figure: SVM hyperplane separating non separable data.

The Non-Separable Case

Introduce **slack variables** $\xi^{(i)}$ that determine how much point i is over the margin.

We now want to **maximise** the margin and **minimise** the slack, or:

$$\min_{\boldsymbol{\theta},\theta_0} \left[\frac{1}{m} + C \sum_{i=1}^{N} \xi^{(i)} \right]$$

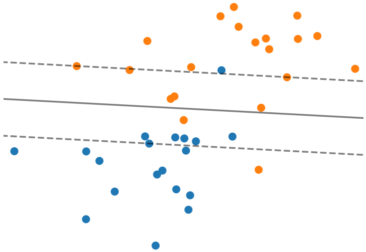


Figure: SVM hyperplane separating non separable data.

The **support vectors** are all points on or over the margin.

The Solution

The optimisation problem is equivalent to:

$$\min_{\alpha^{(i)}} \left[\sum_{i=1}^{N} \alpha^{(i)} - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha^{(i)} \alpha^{(j)} y^{(i)} y^{(j)} (\mathbf{x}^{(i)})^\top \mathbf{x}^{(j)} \right]$$

$$\text{subject to } 0 \leq \alpha^{(i)} \leq C$$

which can be solved numerically using the Sequential Minimal Optimisation (SMO) algorithm.

The Solution

The optimisation problem is equivalent to:

$$\min_{\alpha^{(i)}} \left[\sum_{i=1}^{N} \alpha^{(i)} - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha^{(i)} \alpha^{(j)} y^{(i)} y^{(j)} (\mathbf{x}^{(i)})^\top \mathbf{x}^{(j)} \right]$$

$$\text{subject to } 0 \leq \alpha^{(i)} \leq C$$

which can be solved numerically using the Sequential Minimal Optimisation (SMO) algorithm.

$\boldsymbol{\theta}$ and θ_0 are:

$$\boldsymbol{\theta}^* = \sum_{i=1}^{n} \alpha^{(i)} y^{(i)} \mathbf{x}^{(i)} \qquad \theta_0^* = y^{(s)} - \boldsymbol{\theta}^{*\top} \mathbf{x}^{(s)}.$$

where s is any of the support vectors.

The problem: Non - linear data

Sometimes the data points exhibit a non linear relationship...

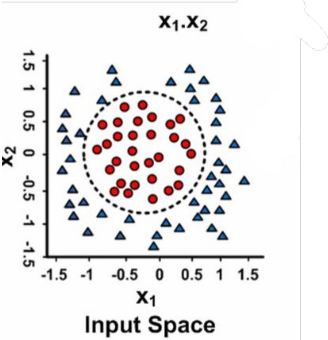


Figure: Muhammad Awais Bin Altaf, DOI: 10.1109/TBCAS.2014.2386891

The problem: Non - linear data

Sometimes the data points exhibit a non linear relationship...

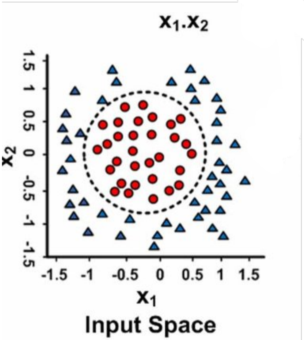


Figure: Muhammad Awais Bin Altaf, DOI: 10.1109/TB-CAS.2014.2386891

We need a way of classifying these data points, since they are **not linearly separable**.

The solution: Kernels

Kernels are functions which can help with transforming the data points into a space where we can separate them. The general form for a Kernel function K is:

$$K(\mathbf{x}, \mathbf{y}) = \Phi(\mathbf{x}) \cdot \Phi(\mathbf{y})$$

- $\Phi(\mathbf{x})$ and $\Phi(\mathbf{y})$ denotes the transformation. This is called the **Kernel trick**.

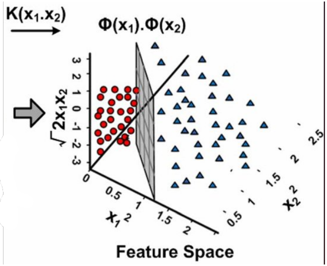


Figure: Muhammad Awais Bin Altaf, DOI: 10.1109/TBCAS.2014.2386891

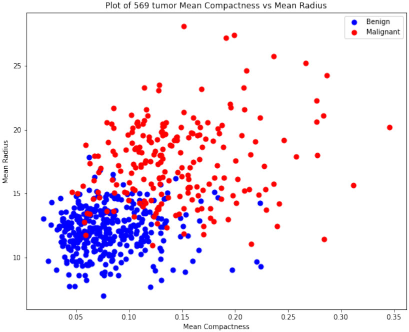
Common kernels

Examples of kernel functions typically used in practice include, for feature vectors $\mathbf{x}$ and $\mathbf{y}$:

- Polynomial Kernel:
  - $K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + c)^d$, where c is a constant and d is the polynomial degree
- Gaussian Radial Basis Function (RBF) Kernel:
  - $K(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}\right) = \exp\left(-\gamma\|\mathbf{x} - \mathbf{y}\|^2\right)$, where $\sigma > 0$
- Sigmoid Kernel:
  - $K(\mathbf{x}, \mathbf{y}) = \tanh(\alpha\mathbf{x} \cdot \mathbf{y} + c)$, where α is a scaling parameter and c is constant

Example: Breast Cancer Dataset

We applied various classification techniques to a real life dataset:



- Features of cell nuclei from biopsy of suspected cancer tumor
- Data points - 569
- Features - 30 - Radius, Texture, Area, etc.

Results: Breast Cancer Dataset

Table: Model Metrics

Table: Linear

Model	Accuracy
Naïve Bayes	0.929
LDA	0.956
Logistic	0.972
SVM (Linear)	0.975

Table: Non-linear

Model	Accuracy
QDA	0.965
SVM (Polynomial)	0.965
SVM (RBF)	0.979

- Used 5 fold cross validation to compute the model accuracy
- Among linear models, SVM with linear kernel and regularization of $C = 0.1$ gives the best accuracy of **0.975**
- SVM with RBF kernel and $C = 2$ and $\gamma = 0.04$ had the highest accuracy of **0.979**

Conclusion

You should now know:

- SVMs are a method to solve **binary classification problems**, by building a hyperplane to separate the classes
- The optimal hyperplane is the one that **maximises the margin** and **minimises the slack**
- (Non - linear) Kernels allow us to separate non - linear data
- Application of various models on a real dataset of breast cancer patients showed SVM with RBF kernels with the best performance