Mathematics for Machine Learning

Prof Willie Brink

Applied Mathematics, Stellenbosch University

Lecture 11: Classification with SVMs

Contents of the module

Introduction

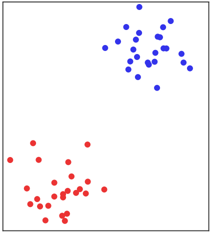
Consider the following binary classification problem.

Given training data $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N)$ consisting of input vectors $\boldsymbol{x}_n \in \mathbb{R}^D$ and corresponding labels $y_n \in \{-1, 1\}$, learn a classifier $f(\boldsymbol{x})$ such that

$$f(\boldsymbol{x}_n) \begin{cases} > 0, & \text{if } y_n = 1 \\ < 0, & \text{if } y_n = -1 \end{cases} \qquad \text{or equivalently: } \; y_n f(\boldsymbol{x}_n) > 0$$

We'll consider linear classifiers of the form $f(\boldsymbol{x}) = \boldsymbol{w}^\mathsf{T}\boldsymbol{x} + b$.

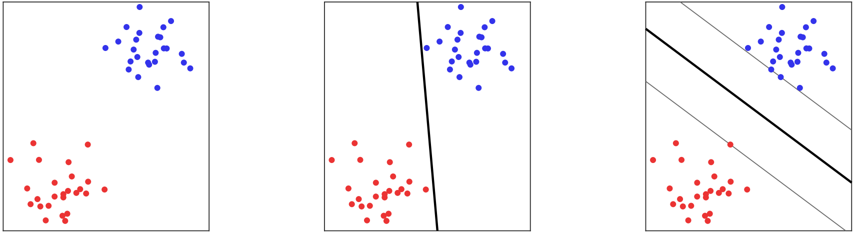
For $D = 2$, $f(\boldsymbol{x}) = 0$ represents a line that splits $\mathbb{R}^2$ into two regions: where $f(\boldsymbol{x}) > 0$, and where $f(\boldsymbol{x}) < 0$.



12.1 Separating hyperplanes

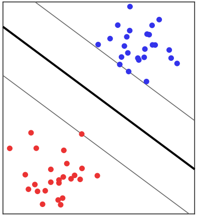
A linear classifier $f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b$ leads to a linear decision boundary, $f(\boldsymbol{x}) = 0$, requiring the two classes to be linearly separable.

Parameters $\boldsymbol{w}$ and b influence the position and orientation of the decision boundary. What would be the best choice for these parameters?



The maximum margin solution might be most stable under perturbations of the input.

We want to maximise the margin around the decision boundary.
Training samples touching the margin are called support vectors.



Note: $\boldsymbol{w}^\mathsf{T}\boldsymbol{x} + b = 0$ and $c(\boldsymbol{w}^\mathsf{T}\boldsymbol{x} + b) = 0$ define the same line (or hyperplane in D dimensions).

We choose the scale of $\boldsymbol{w}$ and b such that $\boldsymbol{w}^\mathsf{T}\boldsymbol{x}_+ + b = 1$ and $\boldsymbol{w}^\mathsf{T}\boldsymbol{x}_- + b = -1$ for positive and negative support vectors, respectively.

With this normalisation, the width of the margin turns out to be $2/\|\boldsymbol{w}\|$.

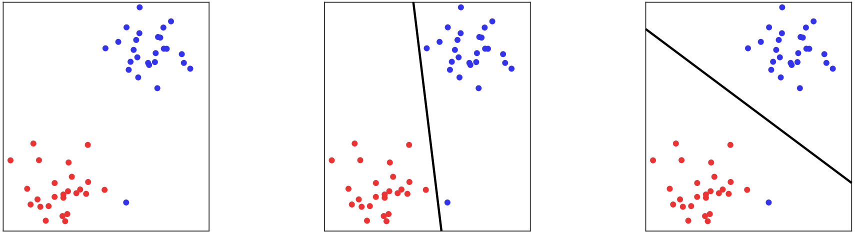
Learning the classifier is done through solving the following optimisation problem:

$$\max_{\boldsymbol{w},b} \frac{2}{\|\boldsymbol{w}\|} \quad \text{subject to} \quad \boldsymbol{w}^\mathsf{T}\boldsymbol{x}_n + b \begin{cases} \geq 1, & \text{if } y_n = 1 \\ \leq -1, & \text{if } y_n = -1 \end{cases} \qquad n = 1, \ldots, N$$

12.2 Primal support vector machine

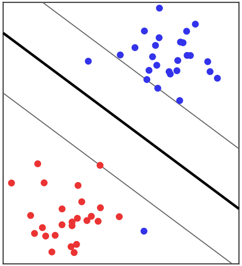
Equivalent problem: $\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2$ subject to $y_n(\boldsymbol{w}^\mathsf{T}\boldsymbol{x}_n + b) \geq 1 \quad n = 1, \ldots, N$

Now, should the classifier fit the training data *perfectly*?



There could be a trade-off between the margin size and the number of mistakes made on the training data.

We assign a slack variable ξ_n to every $\boldsymbol{x}_n$ in the training set.



- if $\xi_n \leq 0$, $\boldsymbol{x}_n$ is classified correctly and is outside the margin
- if $0 < \xi_n < 1$, $\boldsymbol{x}_n$ is classified correctly but violates the margin
- if $\xi_n \geq 1$, $\boldsymbol{x}_n$ is misclassified

The optimisation problem becomes:

$$\min_{\boldsymbol{w},b,\xi_1,\ldots,\xi_N} \tfrac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{n=1}^{N}\xi_n \quad \text{subject to} \quad y_n(\boldsymbol{w}^\mathsf{T}\boldsymbol{x}_n + b) \geq 1 - \xi_n \quad n = 1,\ldots,N$$

The parameter C is a regulariser.

- a small C allows constraints to be easily ignored $\implies$ large margin
- a large C makes constraints hard to ignore $\implies$ narrow margin

Multi-class SVM

SVMs find a maximum margin solution to separate **two** classes in $\mathbb{R}^D$, and there is no definitive multi-class SVM formulation.

One-vs-rest

Training: learn an SVM for each class vs the others.

Prediction: apply each SVM to the test sample, and assign class according to the SVM that returns the highest decision score.

One-vs-one

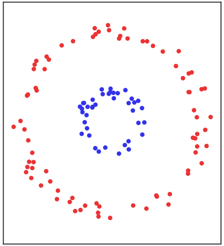
Training: learn an SVM for each pair of classes.

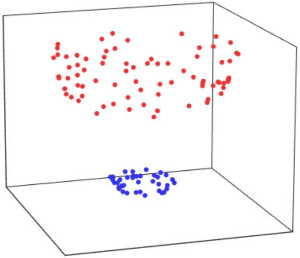
Prediction: each SVM votes on a class to assign to the test sample, and we pick the class with most votes (breaking ties with decision scores).

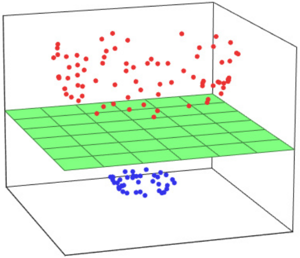
12.4 Kernels

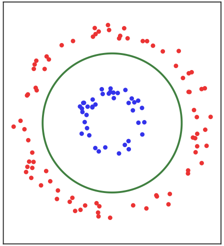
What if the classes are not even almost linearly separable?

Map feature vectors to another space, where the classes are linearly separable!









Choose a kernel function $k : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ for which there exists a function $\phi : \mathbb{R}^D \to \mathbb{R}^{D'}$ such that $k(\boldsymbol{a}, \boldsymbol{b}) = \phi(\boldsymbol{a})^\mathsf{T} \phi(\boldsymbol{b})$.

When we train an SVM, or predict with an SVM, we don't need to know what ϕ is!

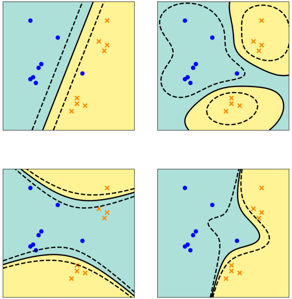
We're never interested in the D'-dimensional version of a vector, only in dot products of two of those vectors, so we just apply the kernel function without ever specifying ϕ.

Examples (where γ, r, d are hyperparameters):

RBF kernel: $k(\boldsymbol{a}, \boldsymbol{b}) = e^{-\gamma \|\boldsymbol{a}-\boldsymbol{b}\|^2}$

Polynomial: $k(\boldsymbol{a}, \boldsymbol{b}) = (\gamma(\boldsymbol{a}^\mathsf{T} \boldsymbol{b}) + r)^d$

Sigmoidal: $k(\boldsymbol{a}, \boldsymbol{b}) = \tanh(\boldsymbol{a}^\mathsf{T} \boldsymbol{b} + r)$



The choice of a kernel and tuning of its hyperparameters can be done through validation.