Support Vector Machine

Jordi Villà i Freixa

Universitat de Vic - Universitat Central de Catalunya
Study Abroad

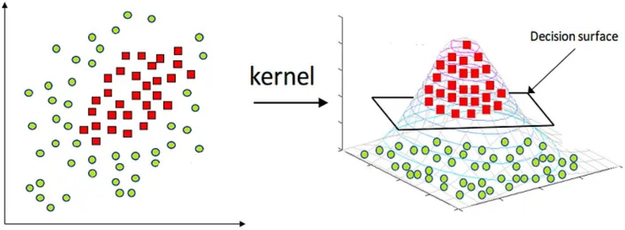
jordi.villa@uvic.cat

course 2023-2024

Índex

1. Introduction
2. Maximal MArgin Classifier
3. Support Vector Classifier
4. Support vector machines
5. Bibliography

Support-vector machine and kermel functions

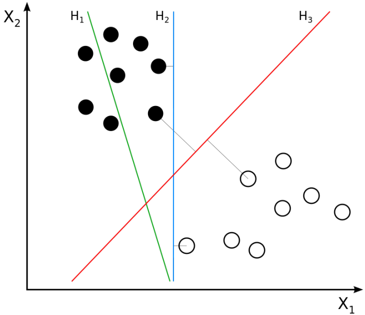


Support-vector machines (SVMs) are a set of algorithms of supervised learning developed by Vladimir Vapnik and his team at the AT&T Bell labs.

Let us take a labelled dataset with two classes or categories. Our goal is to predict the category of a new data point. Intuitively, an SVM is a model that represents the points of a dataset in space, separating the two classes into two subespaces that are as large as possible by means of a hyperplane that depends on the position of the closest points between the two classes (the support vector).

Example in 2 dimensions

The SVM algorithm looks for a hyperplane, in this case a straight line, that separates the two data categories. But which? The one that provides a larger margin between the elements of the two categories.



Support vectors

are the data points, which are closest to the hyperplane. These points will define the separating line better by calculating margins. These points are more relevant to the construction of the classifier.

A **hyperplane** is a decision plane which separates between a set of objects having different class memberships. If we have p-dimensional space, a hyperplane is a flat subspace with dimension $p-1$. In $\mathbb{R}^3$, the hyperplane is a line $\beta_0 + \beta_1 X_1 + \beta_2 X_2 = 0$. In $\mathbb{R}^p$,

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p = 0 \tag{1}$$

A **margin** is a gap between the two lines on the closest class points. This is calculated as the perpendicular distance from the line to support vectors or closest points. If the margin is larger in between the classes, then it is considered a good margin, a smaller margin is a bad margin.

Hyperplane

Following the definition of the hyperplane, if a point $X = (X_1, X_2, \dots, X_p)^T$ in $\mathbb{R}^p$ satisfies 1, then X lies on the hyperplane. Now, suppose that X does not satisfy (9.2); rather,

$$\begin{aligned} \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p &> 0 \\ \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p &< 0 \end{aligned}$$

then X lies above or below the hyperplane. One can easily determine on which side of the hyperplane a point lies by simply calculating the sign of the left hand side of 1.

Hyperplane

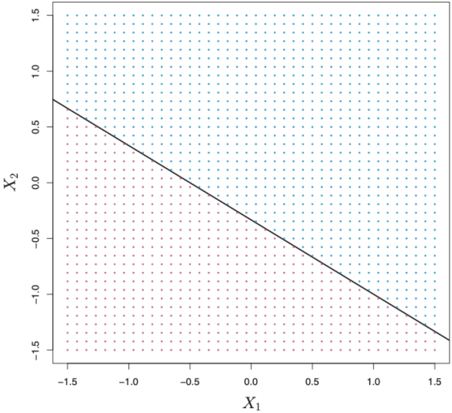


Figure 1: Hyperplane in 2D $1 + 2X_1 + 3X_2 = 0$. The blue and purple regions show points above and below the hyperplane (adapted from [1]).

Hyperplane

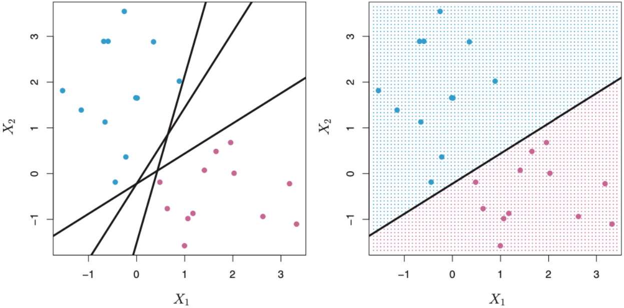


Figure 2: If our data can be perfectly separated using a hyperplane, then there will in fact exist an infinite number of such hyperplanes. [1].

Maximal Margin Classifier

In order to construct a classifier based upon a separating hyperplane, we must have a reasonable way to decide which of the infinite possible separating hyperplanes to use.

A natural choice is the maximal margin hyperplane (also known as the optimal separating hyperplane), which is the separating hyperplane that is farthest from the training observations.[1]

Support vectors

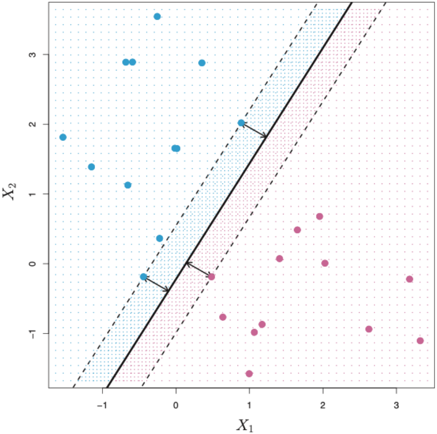


Figure 3: Support vectors [1].

Maximal Margin Classifier

- The maximal margin hyperplane is the separating hyperplane for which the margin is largest.
- We hope that a classifier that has a large margin on the training data will also have a large margin on the test data, and hence will classify the test observations correctly.
- The maximal margin hyperplane represents the mid-line of the widest “slab” that we can insert between the two classes.
- Leads to overfitting when p is large.
- Very sensitive to the training data. It will perform perfectly on the training data set, but maybe poorly on the unseen data. Though it is elegant and simple, this classifier unfortunately cannot be applied to most data sets, since it requires that the classes be separable by a **linear boundary**.

Building a MMC

$$
\begin{array}{rl}
\text{maximize} & M = M(\beta_0, \beta_1, \ldots, \beta_p) \\
\text{subject to} & \begin{array}{l} \sum_{j=1}^{p} \beta_j^2 = 1 \\ y_i(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}) \geq M \; \forall \; i = 1, \ldots, n \end{array}
\end{array}
$$

Sensitivity of the hyperplane

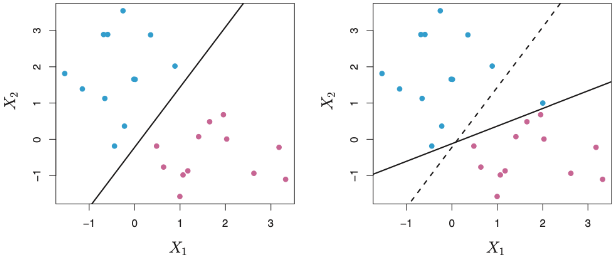
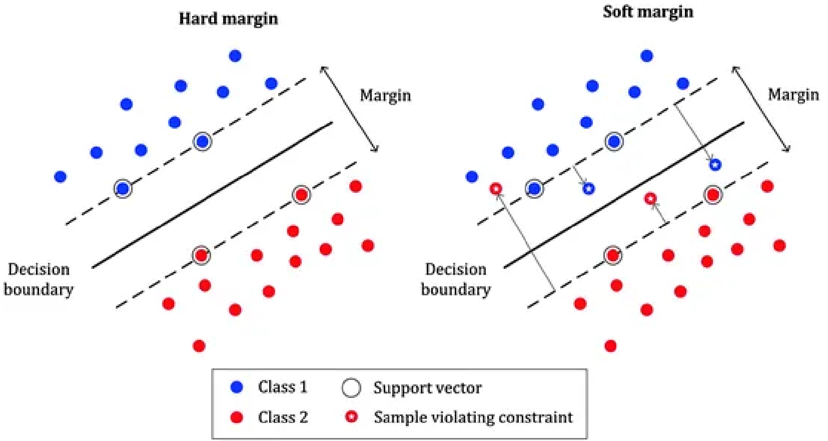


Figure 4: The addition of a single point leads to a dramatic shift in the maximal margin hyperplane shown as a solid line[1].

Soft Margin Classifier

- We need a classifier with greater robustness to individual observations, and better classification of most of the training observations. The generalization of the maximal margin classifier to the non-separable case is known as the support vector classifier or soft margin classifier.
- It is **still linear** a modification of the Maximal-Margin Classifier to relax the margin to handle noisy class boundaries in real data.
- It allows certain points to be deliberately misclassified. By doing this, it is able to classify most of the points correctly in the unseen data and is also more robust.
- An observation can be not only on the wrong side of the margin, but also on the wrong side of the hyperplane. Observations on the wrong side of the hyperplane correspond to training observations that are misclassified by the support vector classifier.

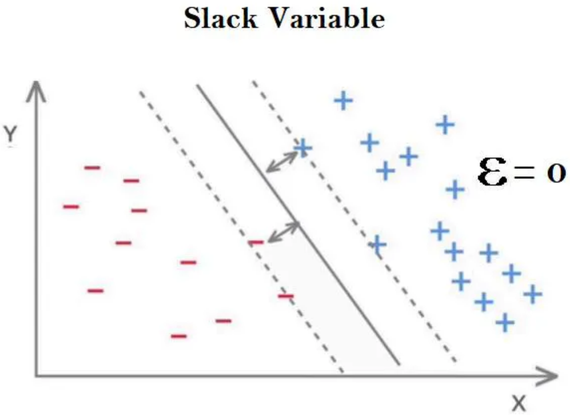

Hard margin
Soft margin
Margin
Margin
Decision boundary
Decision boundary
Class 1
Class 2
Support vector
Sample violating constraint

Building a SVC

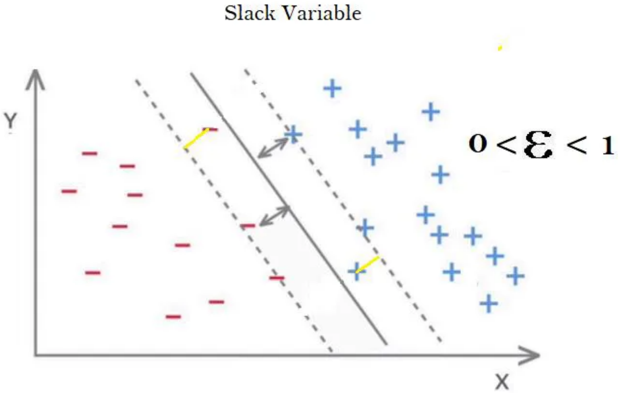
$$
\begin{array}{ll}
\text{maximize} & M = M(\beta_0, \beta_1, \ldots, \beta_p, \varepsilon_1, \ldots, \varepsilon_n) \\
 & \sum_{j=1}^{p} \beta_j^2 = 1 \\
\text{subject to} & y_i(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}) \geq M(1 - \varepsilon_i) \ \forall \ i = 1, \ldots, n \\
 & \varepsilon_i \geq 0, \ \sim_{i=1}^{n} \leq C
\end{array}
$$

Slack variable ε_i

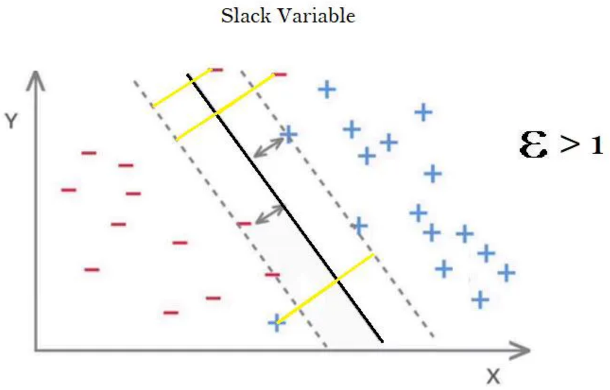
- Slack variables are introduced to allow certain constraints to be violated. That is, certain training points will be allowed to be within the margin:
  - If $\varepsilon_i = 0$ then the ith slack variable observation is on the correct side of the margin
  - If $\varepsilon_i > 0$ then the ith observation is on the wrong side of the margin (it has violated the margin)
  - If $\varepsilon_i > 1$ then it is on the wrong side of the hyperplane.
- We want the number of points within the margin to be as small as possible, and of course we want their penetration of the margin to be as small as possible.


Slack Variable
Y
X
$\varepsilon = 0$

But if you draw a Support Vector Classifier in such a way that it only violates the margin, i.e. $0 < \varepsilon < 1$, the observations classify correctly as shown in figure below.



If the data points violate the hyperplane, i.e. $\varepsilon(\epsilon) > 1$, then the observation is on the wrong side of the hyperplane.



Cost of missclassification

- Cost of misclassification, C, is greater than or equal to the summation of all the epsilons of each data point.
- When C is large, you allow a larger number of data points to be misclassified or to violate the margin.
- When C is small, the margin is narrow and there are few misclassifications.

Non-linear boundaries

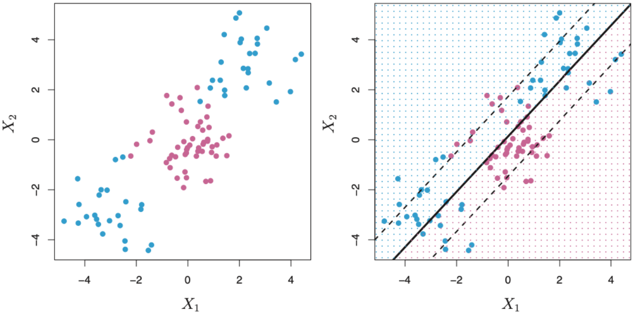


Figure 5: Left: The observations fall into two classes, with a non-linear boundary between them. Right: The support vector classifier seeks a linear bound- ary, and consequently performs very poorly[1].

SVC with quadratic boundaries. Enlarging the feature space

$$\begin{array}{ll} \text{maximize} & M = M(\beta_0, \beta_{11}, \beta_{12} \dots, \beta_{p1}, \beta_{p2}, \varepsilon_1, \dots, \varepsilon_n) \\ & \sum_{j=1}^{p} \sum_{k=1}^{2} \beta_{jk}^2 = 1 \\ \text{subject to} & y_i(\beta_0 + \sum_{j=1}^{p} \beta_{j1} x_{ij} + \sum_{j=1}^{p} \beta_{j2} x_{ij}^2) \geq M(1 - \varepsilon_i)\ \forall\ i = 1, \dots, n \\ & \varepsilon_i \geq 0,\ \sim_{i=1}^{n} \leq C \end{array}$$

Kernels

- We may want to enlarge our feature space support vector machine kernel in order to accommodate a non-linear boundary between the classes.
- The kernel approach is an efficient computational approach for enacting this idea.
- The function of kernel is to take data as input and transform it into the required form.
- Different SVM algorithms use different types of kernel functions. These functions can be linear, nonlinear, polynomial, radial basis function (RBF), and sigmoid.

SVC includes just inner products of the observations

It can be shown that[1]

- The linear support vector classifier, for n training observations, can be represented as

$$f(x) = \beta_0 + \sum_{i=1}^{n} \alpha_i < x, x_i >$$

- To estimate the parameters α_i we just need the inner products of the support vectors

$$f(x) = \beta_0 + \sum_{i \in S}^{n} \alpha_i < x, x_i >$$

What if we could substitute the collection of inner products by other functions that would help us expanding features in an efficient way? This is using kernels in SVM.

Mapping data to higher dimensions

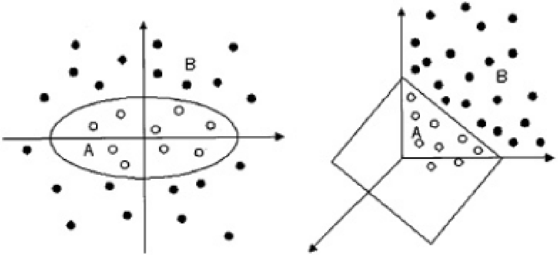


Figure 6: Mapping of non-linear separable training data from $\mathbb{R}^2$ into $\mathbb{R}^3$ [2].

- As feature transformation results in large number of features, it makes the modelling (i.e. the learning process) computationally expensive. The use of kernel resolves this issue. A kernel is a function that quantifies the similarity of two observations. In case of the linear kernel, it provides the SVC:

$$K(X_i, X_i') = \sum_{j=1}^{p} x_{ij} x_{i'j}$$

- To find a best fit model, the learning algorithm only needs the inner products of the observations.
- Kernel Function generally transforms the training set of data so that a non-linear decision surface is able to transformed to a linear equation in a higher number of dimension spaces.
- Basically, It returns the inner product between two points in a standard feature dimension.

Popular kernels

- The linear kernel.
- The polynomial kernel.
- The radial basis function (RBF) kernel. It is capable of creating elliptical (i.e. enclosed) decision boundaries.
- The sigmoid kernel.

$$\text{Linear}: \; K(w,b) = w^T x + b$$

$$\text{Polynomial}: \; K(w,x) = (\gamma w^T x + b)^N$$

$$\text{Gaussian RBF}: K(w,x) = \exp(-\gamma ||x_i - x_j||^n$$

$$\text{Sigmoid}: K(x_i, x_j) = \tanh(\alpha x_i^T x_j + b)$$

Fariha Sohil, Muhammad Umair Sohali, and Javid Shabbir.
An introduction to statistical learning with applications in R: by Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani, New York, Springer Science and Business Media, 2013, $41.98, eISBN: 978-1-4614-7137-7.
Statistical Theory and Related Fields, 6(1):87–87, January 2022.

Martin Hofmann.
Support Vector Machines — Kernels and the Kernel Trick An elaboration for the Hauptseminar " Reading Club : Support Vector Machines ".
2006.