Machine Learning

Support Vector Machines (SVM)

Edgar F. Roman-Rangel.
edgar.roman@itam.mx

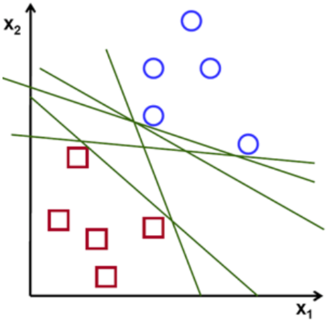
Digital Systems Department.
Instituto Tecnológico Autónomo de México, ITAM.

May 8th, 2021.

Outline

Intro

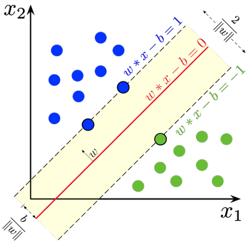
Support Vector Machine: binary classifier.
Separate two sets (classes, $y = \{-1, +1\}$), by using a hyperplane.



Challenge: find the optimal hyperplane.

Optimal hyperplane

Largest margin with respect to closest point from each class.



- Distance $\rightarrow$ confidence on the decision.
- $\mathbf{w}$: vector perpendicular to the hyperplane.
- Hyperplane of $(N-1)$-D, where N is the number of features.
- Number of hyperplanes: $|C|-1$; $|C|$ is the number of classes.

Margin

Distance from positive to negative boundaries,

$$\begin{array}{rrl} & wx^{+} + b & = \ \ 1 \\ - & wx^{-} + b & = -1 \\ \hline & w(x^{+} - x^{-}) & = \ \ 2 \end{array}$$

Normalize distance to unit length:

$$\frac{w(x^{+} - x^{-})}{\|w\|} = \frac{2}{\|w\|}.$$

Maximizing $\frac{2}{\|w\|}$ is equivalent to minimizing $\|w\|$. Let's use:,

$$\frac{1}{2}\|w\|^2.$$

Constraints

We must consider some constraints for the margin.

Hard margin classifier

$$\frac{1}{2}\|w\|^2 + \sum_{m=1}^{M} \left(1 - y^{(m)} \langle\, x^{(m)}, w \rangle\right).$$

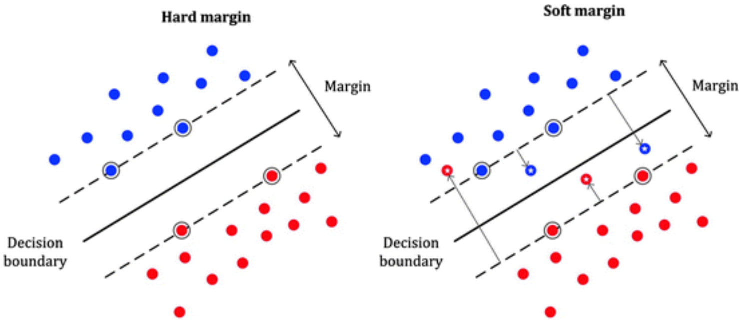
Let's become robust to outliers by allowing a few mistakes.

Soft margin classifier

$$\frac{1}{2}\|w\|^2 + C\left(\sum_{k=1}^{K} \xi^{(k)}\right),$$

where, ξ is a slack variable that penalizes misclassifications, and C is a hyperparameter.

Hard vs soft margins



Lagrangian dual

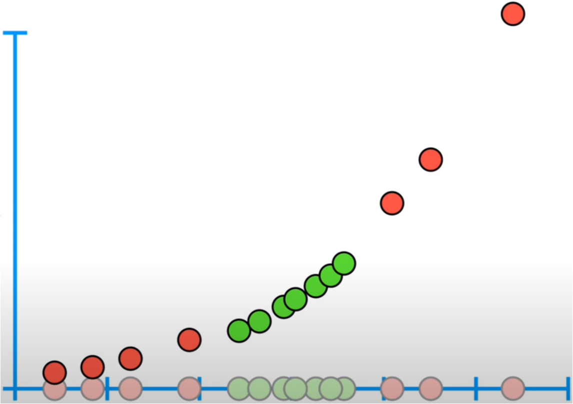
After some math, and optimization rules, we end up with:

$$\begin{aligned} \max_{\alpha} \quad & \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i^T x_j \\ \text{s.t.} \quad & \sum_i \alpha_i y_i = 0, \\ & C \geq \alpha_i \geq 0, \forall i. \end{aligned}$$

Outline

Non-linearly separable set

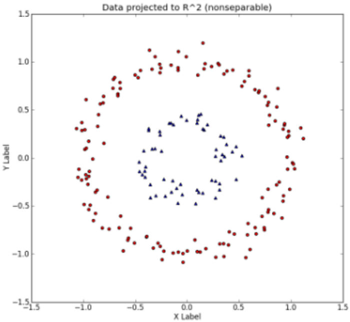
Often, real-world problems are not really linearly separable.

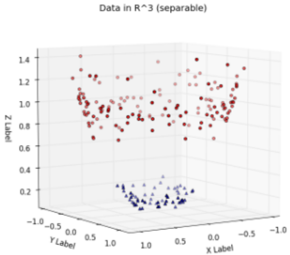


We must find a projection of data, where it becomes linearly separable.

Projection

2-D data projected onto a 3-D plane.





Kernel functions

Linear:

$$\kappa(\mathbf{a}, \mathbf{b}) = \mathbf{a}^T \mathbf{b}.$$

Polynomial:

$$\kappa(\mathbf{a}, \mathbf{b}) = \left(1 + \sum_{j=1} a_j b_j\right)^d.$$

Radial Basis Function:

$$\kappa(\mathbf{a}, \mathbf{b}) = \exp\left(\frac{-\gamma \|\mathbf{a} - \mathbf{b}\|^2}{2\sigma^2}\right).$$

Sigmoid:

$$\kappa(\mathbf{a}, \mathbf{b}) = \tanh\left(c\mathbf{a}^T \mathbf{b} + h\right).$$

Kernel trick

Compute the relationship between pairs of point, as if they were in the higher dimensional space (do not perform the actual mapping).

Somehow, it is equivalent to map each point to a 1-D space, where its position is the distance with respect to a reference point.

Q&A

Thank you!

edgar.roman@itam.mx