Instituto Superior Técnico

Departamento de Engenharia Electrotécnica e de Computadores

Machine Learning

5^{th} Lab Assignment

Shift 5ª a tarde Group number 2

Number 81567 Name José Pedro Boavida Miragaia

Number 81138 Name João Gabriel Albuquerque Simões Furtado Ramiro

Support Vector Machines for Classification

1 Introduction

Simple linear classifiers, such as the one implemented by the Rosenblatt perceptron, are unable to correctly classify patterns, unless the classes under consideration are linearly separable. Neural networks that use hidden units with nonlinear activation functions are used in many classification problems, since they are able to perform nonlinear classification. However, several strong theoretical results, valid for the linearly separable case, are not applicable to nonlinear classifiers.

Support vector machines (SVMs) address the classification problem using linearly separable classes, not in the input space, but in the so-called *feature space*. Input patterns are mapped onto the higher-dimensional feature space, where the classification is performed using a hyperplane as classification border. Since the mapping from the input space to the feature space is usually nonlinear, these hyperplanes in feature space correspond to nonlinear borders in input space.

At first glance this might seem to be a double-edged sword, since it suggests that calculations have to be performed in the high-dimensional feature space. However, an interesting result proves that, since linear classification only requires inner product operations, all calculations can be performed in the lower-dimensional input space, if the nonlinear mapping is chosen in an appropriate way. This result is particularly strong when one takes into account that certain mappings yield infinite-dimensional feature spaces. This is the same as saying that linear classification in an infinite-dimensional feature space can be performed by means of operations in the lower-dimensional input space. Imagine all the power of infinite-dimensional hyperplanes, without the associated computational burden.

The purpose of this assignment is twofold: first, to work out, in detail, two simple classification problems in two-dimensional input space, one of them involving a mapping to a three-dimensional feature space; second, to provide some experience and some intuition on the capabilities of support vector machines.

2 Two simple examples

Consider the AND and XOR logic functions, defined in the following truth table:

x_1	x_2	d_{AND}	d_{XOR}
-1	-1	-1	-1
-1	1	-1	1
1	-1	-1	1
1	1	1	-1

Here, the input pattern is a vector $\mathbf{x} = (x_1, x_2)$, and d_{AND} and d_{XOR} are the desired values for the AND and XOR functions. Note that, in this assignment, we represent logical *true* by 1 and logical *false* by -1. Similarly, in binary classification problems, we assign the desired value of 1 to the patterns of one of the classes, and the desired value of -1 to those of the other class.

2.1(T) For the AND function, find (by inspection) the maximum-margin separating straight line, the support vectors and the margin boundaries. Then compute the vector $\mathbf{w}$ and the bias b that satisfy the equation

$$(\mathbf{w} \cdot \mathbf{x}^s + b)\, d^s = 1 \tag{1}$$

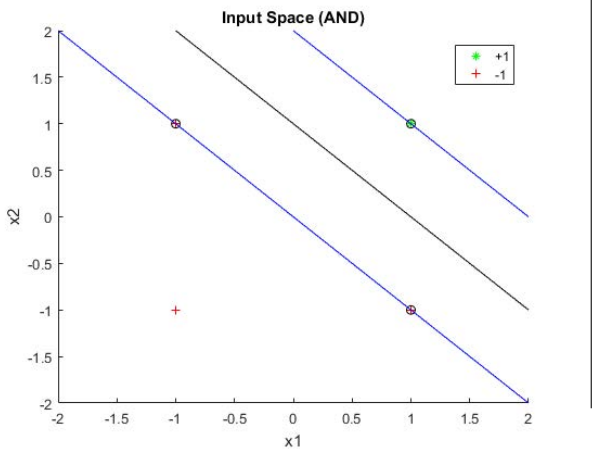
for all support vectors $\mathbf{x}^s$, where d^s is the desired value corresponding to $\mathbf{x}^s$.[1]

R.

Por inspeção compreende-se que a fronteira que origina uma margem máxima corresponde à equação $X_2 = 1 - X_1$

Desenvolvendo a fórmula da equação (1), obtém-se três equacções, uma para cada Support Vector:

$$\begin{cases} (W_1 + W_2 + b) \cdot 1 = 1 \\ (-W_1 + W_2 + b) \cdot (-1) = 1 \\ (W_1 - W_2 + b) \cdot (-1) = 1 \end{cases}$$



Resolvendo este sistema são obtidos:

$$\begin{cases} W_1 = 1 \\ W_2 = 1 \\ b = -1 \end{cases}$$

Apartir deste valores consegue-se as equacções da fronteira e margens:

$$fronteira: X_1 + X_2 = 1$$

$$margem_+: X_1 + X_2 = 2$$

$$margem\_: X_1 + X_2 = 0$$

2.2(T) Since, for the XOR function, a linear classification cannot be performed in the input space – explain why – we will consider here a simple nonlinear mapping to a three-dimensional feature space:

$$\tilde{\mathbf{x}} = \varphi(\mathbf{x}) = (x_1, x_2, x_1 x_2)^T . \tag{4}$$

Find the kernel function that corresponds to this mapping.

[1] It can be easily shown (but you're not asked to show) that, defining the border of the maximum-margin linear classifier by the equation

$$\widetilde{\mathbf{w}} \cdot \tilde{\mathbf{x}} + b = 0, \tag{2}$$

then $\widetilde{\mathbf{w}}$ and b obey the equation

$$(\widetilde{\mathbf{w}} \cdot \tilde{\mathbf{x}}^s + b)\, d^s = C \tag{3}$$

for all support vectors $\tilde{\mathbf{x}}^s$, where C is a constant. Normally, we choose $C = 1$.

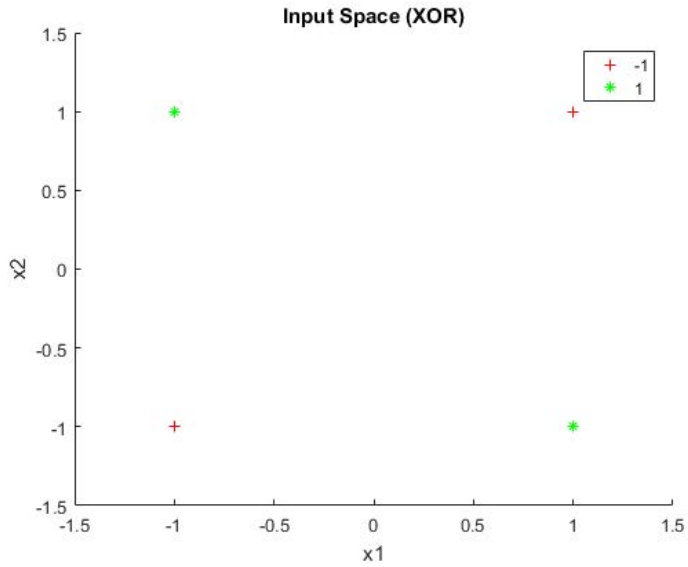
In our case of the AND function, vectors with and without tilde are equal, since the feature space (where the linear classification is performed) is the input space itself.

R. Ao observar a imagem ao lado compreende-se imediatamente que não é possível separar as duas classes, pois não há nenhuma reta que consiga fazer tal separação uma vez que as classes estam cruzadas entre si, no entanto se se transformar este espaço num de maior dimensão haverá um hiperplano capaz de separar as classes, que corresponderá a uma fronteira no input space, que não é uma reta em 2D.

$$\varphi(\boldsymbol{X}) = \begin{bmatrix} X_1 \\ X_2 \\ X_1X_2 \end{bmatrix}$$

$$K(\boldsymbol{X},\boldsymbol{Y}) = \varphi(\boldsymbol{X}) \cdot \varphi(\boldsymbol{Y})^T$$

$$K(\boldsymbol{X},\boldsymbol{Y}) = X_1Y_1 + X_2Y_2 + X_1X_2Y_1Y_2$$



2.3(T) Visualize the points in this 3D feature space. Find, by inspection, which are the support vectors. Compute $\widetilde{\mathbf{w}}$ and b in this feature space, so that equation (1) is satisfied for all support vectors.

R.

Observado o gráfico ao lado compreende-se que todos os pontos são Support Vectors pois apenas sendo todos vetores de suporte é que se obtém um plano de fronteira com margem máxima.

Tal como anteriormente para descobrir $\widetilde{\mathbf{W}}$ **e** b é necessária a equação:

$$\left(\widetilde{\mathbf{W}} \cdot \widetilde{\boldsymbol{X^s}} + b\right) d^s = 1$$

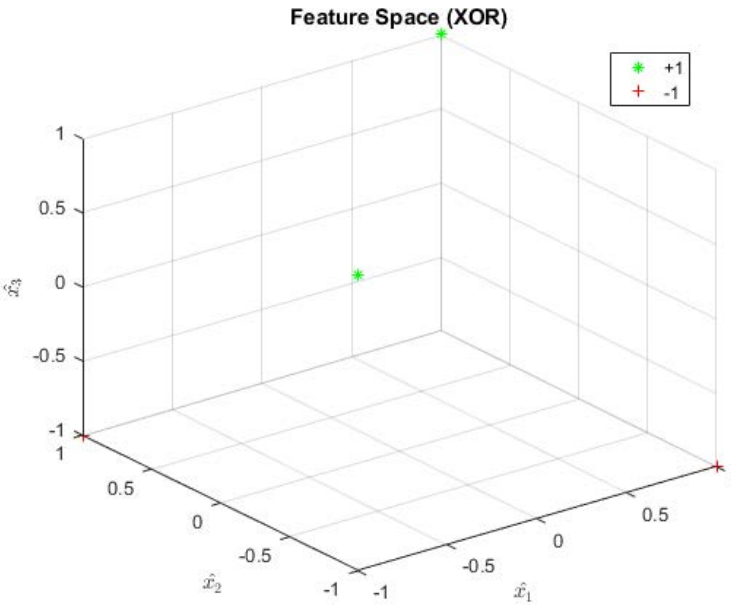
Onde $\widetilde{\boldsymbol{X^s}} = \varphi(\boldsymbol{X^s})$

Assim obtém-se os segunites quatro Support Vectors no feature space:

$$\widetilde{\boldsymbol{X_+^1}} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad \widetilde{\boldsymbol{X_+^2}} = \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix} \quad \widetilde{\boldsymbol{X_-^2}} = \begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix} \widetilde{\boldsymbol{X_-^2}} = \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix}$$

Substituindo na equação (1) obtém-se:

$$\begin{cases} \widetilde{W}_1+\widetilde{W}_2+\widetilde{W}_3+b=1 \\ -\widetilde{W}_1-\widetilde{W}_2+\widetilde{W}_3+b=1 \\ -\widetilde{W}_1 + \widetilde{W}_2 - \widetilde{W}_3 + b = -1 \\ \widetilde{W}_1 - \widetilde{W}_2 - \widetilde{W}_3 + b = -1 \end{cases} = \begin{cases} \widetilde{W}_1=0 \\ \widetilde{W}_2=0 \\ \widetilde{W}_3 = 1 \\ b = 0 \end{cases}$$



2.4(T) Algebraically express, in the two-dimensional input space, the classification border and the margin boundaries corresponding to the classifier found above for the XOR problem. Then sketch them in a graph, together with the input patterns.

R. Para o feature space, as equacões de margem e fronteiras serão as seguintes:

$$fronteira: \widetilde{X_3}\widetilde{W_3} = 0$$

$$margem_+: \widetilde{X_3}\widetilde{W_3} = 1$$

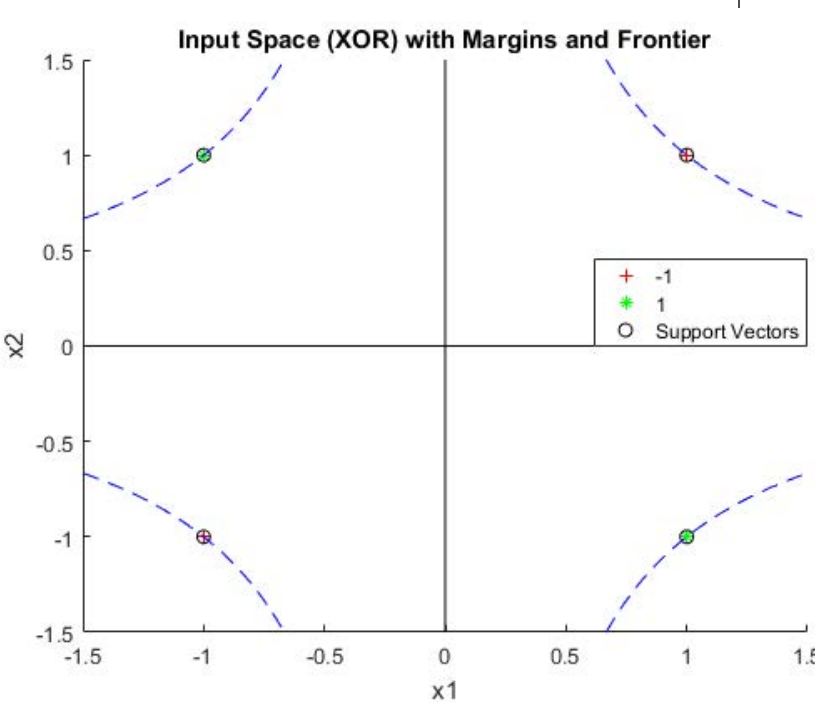
$$margem\_: \widetilde{X_3}\widetilde{W_3} = -1$$

Convertendo em input space obtém se

$$fronteira: X_1 = 0 \vee X_2 = 0$$

$$margem_+: X_2 = -\frac{1}{X_1}$$

$$margem\_: X_2 = \frac{1}{X_1}$$



2.5(T) Indicate the mathematical condition under which the classifier that you have just developed will produce an output of 1. The condition should be expressed in terms of the input space coordinates. It shouldn't use coordinates from the feature space.

R. Atavés da imagem da alínea anterior, compreende-se que os quadrantes ímpares correspondem à classe 1 e os pares à classe -1. Assim, a condição que que corresponde à região da classe 1 corresponde a:

$$(X_1 > 0 \wedge X_2 > 0) \vee (X_1 < 0 \wedge X_2 < 0)$$

3 Classification using SVMs

A kernel commonly employed in pattern recognition problems is the polynomial one, defined by

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + a)^p - a^p, \tag{5}$$

where $a \in \mathbb{R}^+$ and $p \in \mathbb{N}$.[†]

3.1(T) Consider $\mathbf{x}, \mathbf{y} \in \mathbb{R}^2$, $a = 1$ and $p = 2$. Indicate the mapping to feature space that this kernel corresponds to, and the dimensionality of the feature space.

R. Substituindo no Kernel a e p obtém-se

$$K(\boldsymbol{X}, \boldsymbol{Y}) = (\boldsymbol{X} \cdot \boldsymbol{Y} + 1)^2 + 1$$

Como o polinómio é de grau 2, o $\varphi(\boldsymbol{X})$ terá o seguinte formato:

$$\varphi(\boldsymbol{X}) = \begin{bmatrix} a \\ bX_1^2 \\ cX_2^2 \\ dX_1X_2 \\ eX_1 \\ fX_2 \end{bmatrix} = \begin{bmatrix} \tilde{X}_1 \\ \tilde{X}_2 \\ \tilde{X}_3 \\ \tilde{X}_4 \\ \tilde{X}_5 \\ \tilde{X}_6 \end{bmatrix}$$

Desenvolvendo a expressão do Kernel obtém-se a seguinte equação:

$$K(\boldsymbol{X}, \boldsymbol{Y}) = X_1^2Y_1^2 + 2X_1Y_1X_2Y_2 + 2X_1Y_1 + 2X_2^2Y_2^2 + 2X_2Y_2$$

Também é possível chegar a uma fórmula semelhante através do produto interno de $\varphi(\boldsymbol{X})$:

$$K(\boldsymbol{X}, \boldsymbol{Y}) = \varphi(\boldsymbol{X}) \cdot \varphi(\boldsymbol{Y})^T$$

$$K(\boldsymbol{X}, \boldsymbol{Y}) = a^2 + b^2 \cdot X_1^2Y_1^2 + c^2 \cdot X_2^2Y_2^2 + d^2 \cdot X_1Y_1X_2Y_2 + e^2 \cdot X_1Y_1 + f^2 \cdot X_2Y_2$$

Apartir da comparação das duas expressões é possível descobrir os valores das constantes

$$a = 0; \quad b = 1 \quad c = \sqrt{2} \quad d = \sqrt{2} \quad e = \sqrt{2} \quad f = \sqrt{2}$$

Em alternativa todos os $\sqrt{2}$ poderiam ter sido substituidos por $-\sqrt{2}$

Assim concluí-se que o feature space terá dimensão 5.

$$\varphi(\boldsymbol{X}) = \begin{bmatrix} X_1^2 \\ X_2^2 \\ X_1X_2 \\ X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} \tilde{X}_2 \\ \tilde{X}_3 \\ \tilde{X}_4 \\ \tilde{X}_5 \\ \tilde{X}_6 \end{bmatrix}$$

[†]This is one of the variants of the polynomial kernel. Another variant omits the term "$-a^p$" in the defining equation.

3.2(T) Assume again that $p = 2$ and $a = 1$. Find the vector $\widetilde{\mathbf{w}}$ that represents, in this new feature space, the same classification border and margins as in 2.2.

R.

Aplicando a transformação obtém-se no feature space os seguintes Support Vector:

$$\widetilde{X_+^1} = \begin{bmatrix} 1 \\ 1 \\ \sqrt{2} \\ \sqrt{2} \\ \sqrt{2} \end{bmatrix} \quad \widetilde{X_+^2} = \begin{bmatrix} 1 \\ 1 \\ \sqrt{2} \\ -\sqrt{2} \\ -\sqrt{2} \end{bmatrix} \quad \widetilde{X_-^1} = \begin{bmatrix} 1 \\ 1 \\ -\sqrt{2} \\ -\sqrt{2} \\ \sqrt{2} \end{bmatrix} \quad \widetilde{X^2} = \begin{bmatrix} 1 \\ 1 \\ -\sqrt{2} \\ \sqrt{2} \\ -\sqrt{2} \end{bmatrix}$$

Assim resolvendo o sistema de equações

$$(\widetilde{\mathbf{W}} \cdot \widetilde{X^s} + b)d^s = 1$$

Descobre-se os seguintes $\widetilde{\mathbf{W}}$ e b:

$$\widetilde{\mathbf{W}} = \begin{bmatrix} \widetilde{W_2} \\ \widetilde{W_3} \\ \widetilde{W_4} \\ \widetilde{W_5} \\ W_6 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -\frac{1}{\sqrt{2}} \\ 0 \\ 0 \end{bmatrix} \qquad b = 0$$

Este reultado faz sentido, pois apenas observando o sinal de $\tilde{X}_4$ obtém-se a classe do ponto em análise, sem usar as outras dimensões do espaço, daí a atribuição de 0 a estas. b corresponde a 0 pois o hiperplano que separa as classes passa pela origem.

Assim tal como para o feature space feito em 3D préviamente, as equacões de margem e fronteiras para este feature space são as seguintes:

$$fronteira: \widetilde{X_4}\widetilde{W_4} = 0$$
$$margem_+: \widetilde{X_4}\widetilde{W_4} = 1$$
$$margem_-: \widetilde{X_4}\widetilde{W_4} = -1$$

Que substituindo $\widetilde{X_3}$ por X_1X_2 obtém-se as mesmas margens e fronteira.

$$fronteira: X_1 = 0 \vee X_2 = 0$$
$$margem_+: X_2 = -\frac{1}{X_1}$$
$$margem_-: X_2 = \frac{1}{X_1}$$

4 Experiments

The experimental part of this assignment uses the SVM toolbox from MatLab. Use function `svmtrain` for training the SVM and function `svmclassify` for testing (type help for more information on these functions). You will need to specify the 'kernel_function'. Use 'linear' for a linear classifier (*i.e.,* the feature space is equal to the input space), 'polynomial' for the kernel (5), where 'polyorder' stands for parameter p, and 'rbf' (radial basis function) for the kernel

$$K(\mathbf{x}, \mathbf{y}) = e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}}, \tag{6}$$

where 'rbf_sigma' stands for σ. Among these kernels, Gaussian RBF is the one that is most frequently used, for several reasons (for instance, because it is shift-invariant and isotropic).

Set the `svmtrain` parameter 'Method' to 'QP' to choose Quadratic Programming as the optimization method, and the 'boxconstraint' parameter to 10^4. This parameter corresponds to the soft margin penalty 'C' which specifies the relative weight of the margin violations in the objective function that is optimized in the training of the classifier.

When training and testing your classifiers, set option 'Showplot' to true in order to obtain the plot of the classification.

4.1(E) Load the file `spiral.mat`. This file contains the classical spiral example, with 50 patterns per class. Determine experimentally, using the polynomial kernel, the value of p for which you get the best classifier. (start with $p = 1$). Write down all experiments performed, together with the classification error percentages and number of support vectors (the support vectors can be obtained from the SVMStruct returned by svmtrain). Comment on the results you obtained.

R.

4.2(E) Using the same data file (`spiral.mat`), try now the Gaussian RBF kernel. Find the approximate value of σ for which you can get the best classifier. Comment on the results you obtained.

R.

4.3(E) Load the file `chess33.mat` and set 'boxconstraint' parameter to Inf to enforce a hard margin SVM, for separable data. Using the Gaussian RBF kernel, find a value of σ that approximately minimizes the number of support vectors, while correctly classifying all patterns. Indicate the value of σ and the number of support vectors.

R.

4.4(E) Load the file `chess33n.mat` which is similar to the one used in the previous question, except for the presence of a couple of outlier patterns. Run the classification algorithm on these data with the same value of σ, and comment on how the results changed, including the shape of the classification border, the margin size and the number of support vectors.

R.

4.5(E) Now reduce the value of 'boxconstraint' parameter in order to obtain the so-called *soft margin* SVM. Try different values (suggestion: use powers of 10) and comment on the results.

R.