Machine Learning (CS 6140)
Homework 2

Instructor: Ehsan Elhamifar
Due Date: November 10, 2016, 11:45am

1 Support Vector Machine

1.1 Feature Map: Consider a binary classification problem in one-dimensional space where the sample contains four data points $S = \{(1, -1), (-1, -1), (2, 1), (-2, 1)\}$ as shown in Fig. 1.

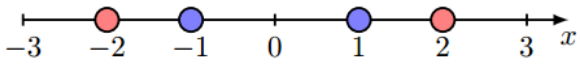


Figure 1: Red points represent instances from class +1 and blue points represent instances from class -1

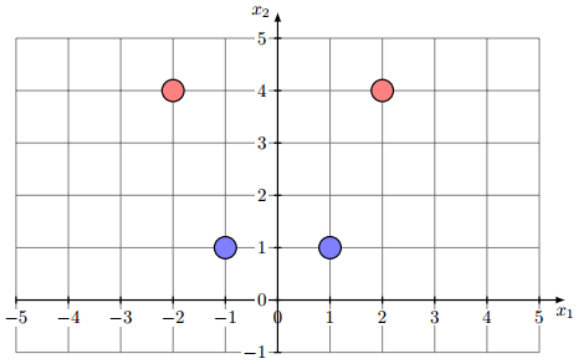


Figure 2: Data instances after the feature map transformation

1. Define $H_t = [t, \infty)$. Consider a class of linear separateors $\mathcal{H} = \{H_t : t \in \mathbb{R}\}$, i.e., for $\forall H_t \in \mathcal{H}$, $H_t(x) = 1$ if $x \geq t$ otherwise -1. Is there any linear separator $H_t \in \mathcal{H}$ that achieves 0 classification error on this sample? If yes, show one of the linear separators that achieves 0 classification error on this example. If not, briefly explain why there cannot be such linear separator.

2. Now consider a feature map $\phi : \mathbb{R} \to \mathbb{R}^2$ where $\phi(x) = (x, x^2)$. . Apply the feature map to all the instances in sample S to generate a transformed sample $S' = \{(\phi(x), y) : (x, y) \in S\}$,

shown in Fig. 2. Let $\mathcal{H}' = \{ax_1 + bx_2 + c \geq 0 : a^2 + b^2 \neq 0\}$ be a collection of half-spaces in R^2. More specifically, $H_{a,b,c}((x1, x2)) = 1$ if $ax_1 + bx_2 + c \geq 0$ otherwise -1. Is there any half-space $H' \in \mathcal{H}'$ that achieves 0 classification error on the transformed sample S'? If yes, give the equation of the max-margin linear separator and compute the corresponding margin. For this question, you can give the equation directly by inspection of Fig. 2.

3. What is the kernel corresponding to the feature map $\phi(\cdot)$ in the last question, i.e., give the kernel function $K(x, z) : \mathbb{R} \times \mathbb{R} \to R$.

1.2 Constructing Kernels

In this question you will be asked to construct new kernels from existing kernels. Suppose $K_1(x, z) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ and $K_2(x, z) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ are both kernels, show the following functions are also kernels:

1. $K(x, z) = c_1 K_1(x, z) + c_2 K_2(x, z)$ with $c_1, c_2 \geq 0$.

2. $K(x, z) = K_1(x, z) \cdot K_2(x, z)$.
3. Let $q(t) = \sum_{i=0}^{p} c_i t^i$ be a polynomial function with nonnegative coefficients, i.e., $c_i \geq 0, \forall i$. Show that $K(x, z) = q(K_1(x, z))$ is a kernel. (Hint: You can use the conclusions from last two questions toprove this one.)

4. $K(x, z) = \exp(K_1(x, z))$. (Hint: You can use the conclusion from the last question to prove this one.)

5. Let A be a positive semidefinite matrix and define $K(x, z) = x^T Az$.

6. $K(x, z) = \exp(-\|x - z\|_2^2)$.

1.3 Support Vectors

In question 1 we explicitly construct the feature map and find the corresponding kernel to help classify the instances using linear separator in the feature space. However in most cases it is hard to manually construct the desired feature map, and the dimensionality of the feature space can be very high, even infinity, which makes explicit computation in the feature space infeasible in practice. In this question we will develop the dual of the primal optimization problem to avoid working in the feature space explicitly. Suppose we have a sample set S = (x1, y1), . . . ,(xn, yn) of labeled examples in $\mathbb{R}^d$ with label set $\{+1, -1\}$. Let $\phi : \mathbb{R}^d \to R^D$ be a feature map that transform each input example to a feature vector in $\mathbb{R}^D$. Recall from the lecture notes that the primal optimization of SVM is given by

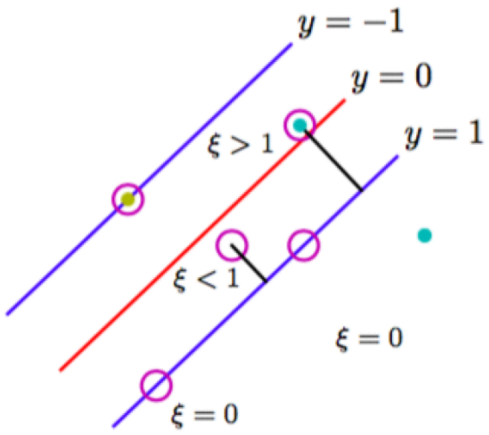
$$
\begin{aligned}
\underset{\mathbf{w},\xi_i}{\text{minimize}} \quad & \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{n}\xi_i & \\
\text{subject to} \quad & y_i(\mathbf{w}^T\phi(x_i)) \geq 1-\xi_i & \forall i = 1, \cdots, n \\
& \xi_i \geq 0 & \forall i = 1, \cdot, n
\end{aligned}
$$

which is equivalent to the following dual optimization

$$
\begin{aligned}
\underset{\alpha_i}{\text{minimize}} \quad & \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j) & \\
\text{subject to} \quad & 0 \leq \alpha_i \leq C & \forall i = 1, \cdots, n \\
& \sum_{i=1}^{n}\alpha_i y_i = 0 & \forall i = 1, \cdot, n
\end{aligned}
$$

Recall from the lecture notes $\xi_1, ..., \xi_n$ are called slack variables. The optimal slack variables have intuitive geometric interpretation as shown in Fig. 3. Basically, when $\xi_i = 0$, the corresponding feature vector $\phi(x_i)$ is correctly classified and it will either lie on the margin of the separator or on the correct side of the margin. Feature vector with $0 < \xi_i \leq 1$ lies within the margin but is still be correctly classified. When $\xi_i > 1$, the corresponding feature vector is misclassified. Support vectors correspond to the instances with $\xi_i > 0$ or instances that lie on the margin. The optimal vector $\mathbf{w}$ can be represented in terms of $\alpha_i, i = 1, \cdots, n$ as $\mathbf{w} = \sum_{i=1}^{n}\alpha_i y_i \phi(\mathbf{x}_i)$.

1. Suppose the optimal $\xi_1, ..., \xi_n$ have been computed. Use the ξ_i to obtain an upper bound



on the number of misclassified instances.

2. In the primal optimization of SVM, what?s the role of the coefficient C? Briefly explain your answer by considering two extreme cases, i.e., $C \to 0$ and $C \to \infty$.

3. Explain how to use the kernel trick to avoid the explicit computation of the feature vector $\phi(\mathbf{x}_i)$? Also, given a new instance $\mathbf{x}$, , how to make prediction on the instance without explicitly computing the feature vector $\phi(\mathbf{x})$?

2. SVM Implementation

Implement SVM with the SMO algorithm and train it on the provided dataset. For you implementation, you only have to use the linear kernel. Also run SVM from a package, try different kernels and compare the results. You can implement the simplified SMO, as described in http://cs229.stanford.edu/materials/smo.pdf

Apply the SVM on the data provided and report the classification result as a function of the regularization parameter C.

3. Multi-Layer Perceptron Implementation

Implement a Feed Forward Neural Network, with an input layer with S1 units, one hidden layer with S2 units, and an output layer with S3 units using the backpropagation algorithm and the sigmoid activation function. Run it on the dataset provided. Try different number of hidden nodes in the hidden layer, S2 = 10, 20, 30, 50, 100, and report the classification results as a function of S2.