Aprendizaje Automático

Beltrán Domínguez, Victoria
Ge, Xiangyang

Memoria Práctica 2: Máquinas de vectores soporte

En esta práctica, se pretende que el alumno establezca una primera toma de contacto con máquinas de vectores soporte(SVM) y la librería de octave LibSVM.
Así, la primera parte de la práctica explica cómo representar gráficamente ciertos conjuntos en octave y también expone el funcionamiento general de la librería.

El primer ejercicio para entregar que se nos plantea, se realiza sobre dos conjuntos diferentes, uno separable y el otro no separable.
Para cada uno de estos conjuntos debemos:

1. **Obtener una SVM sin kernel (es decir, kernel tipo lineal). Para simular la optimización estándar del caso separable basta usar un valor grande de C.**
   Para realizar esta actividad basta con escribir el siguiente comando para cada uno de los diferentes conjuntos con sus respectivos trlabels y tr: res=svmtrain(trlabels,tr,'-t 0 -c 9999');
   Así, el modelo quedará guardado en la variable res.

2. **Determinar para cada conjunto:**
   a. **Los multiplicadores de Lagrange, α, asociados a cada dato de entrenamiento,**

      Para el caso separable, después de ejecutar la instrucción del apartado anterior y teniendo el modelo en la variable res, los multiplicadores de Lagrange se obtienen ejecutando la instrucción res.sv_coef. Así obtenemos los siguientes resultados:

res.sv_coef
ans =

```
  0.87472
  0.74989
 -1.62461
```

Para el caso no separable, obtenemos los siguientes resultados:

res.sv_coef
ans =

```
   250.87
   500.75
  1000.00
  -751.62
 -1000.00
```

b. Los vectores soporte

Los vectores soporte para el conjunto linealmente separable se obtienen ejecutando la siguiente instrucción:
tr(res.sv_indices,:)
ans =

```
  1   4
  4   2
  3   4
```

Para el caso no separable, obtenemos los siguientes resultados:

tr(res.sv_indices,:)
ans =

1 4
4 2
4 4
3 4
4 3

c. **El vector de pesos y umbral de la función discriminante lineal**
Para obtener el valor del umbral, basta con mirar el valor de rho multiplicado por -1 para obtenerlo. Así, para el caso linealmente separable el umbral es 7.9881 y para el caso no linealmente separable, el umbral es 7.9980.

Para calcular el vector de pesos, debemos multiplicar los vectores soporte por los multiplicadores de Lagrange. Así, ejecutando la siguiente instrucción (pesos = tr(res.sv_indices,:)' * res.sv_coef;), obtenemos para el caso linealmente separable:
pesos =

-0.99955
-1.49978

Y para el caso no linealmente separable, obtenemos:

pesos =

-0.99955
-1.49977

d. **El margen correspondiente**

El margen correspondiente se calcula haciendo la norma de los vectores de pesos y dividiendo 2 entre ese número (margen = 2/norm(pesos);). Así, tanto para el caso linealmente separable como para el caso no linealmente separable obtenemos un margen de 1.1097.

3. **Calcular los parámetros de la frontera lineal (recta) de separación**

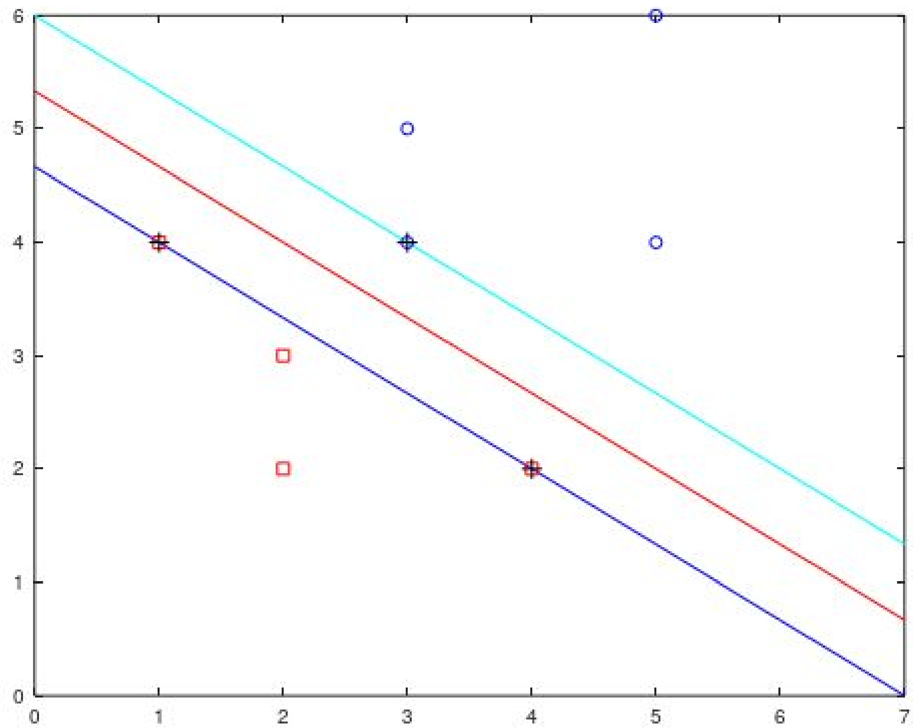
Para calcular los parámetros de la frontera lineal nos basta con calcular la siguiente fórmula:

$$\phi(\mathbf{x};\boldsymbol{\theta},\theta_0)=0 \quad \Rightarrow \quad \theta_1 x_1+\theta_2 x_2+\theta_0=0 \quad \Rightarrow \quad x_2=-\frac{\theta_1}{\theta_2}x_1-\frac{\theta_0}{\theta_2},$$

Así, el primer parámetro se obtendría con la siguiente instrucción -(pesos(1)/pesos(2)) mientras que el segundo parámetro se obtendría con la siguiente instrucción - (umbral/pesos(2)).
Tanto para el caso linealmente separable como para el caso no linealmente separable, el valor de los parámetros es -0.66647 e 5.3332 respectivamente.

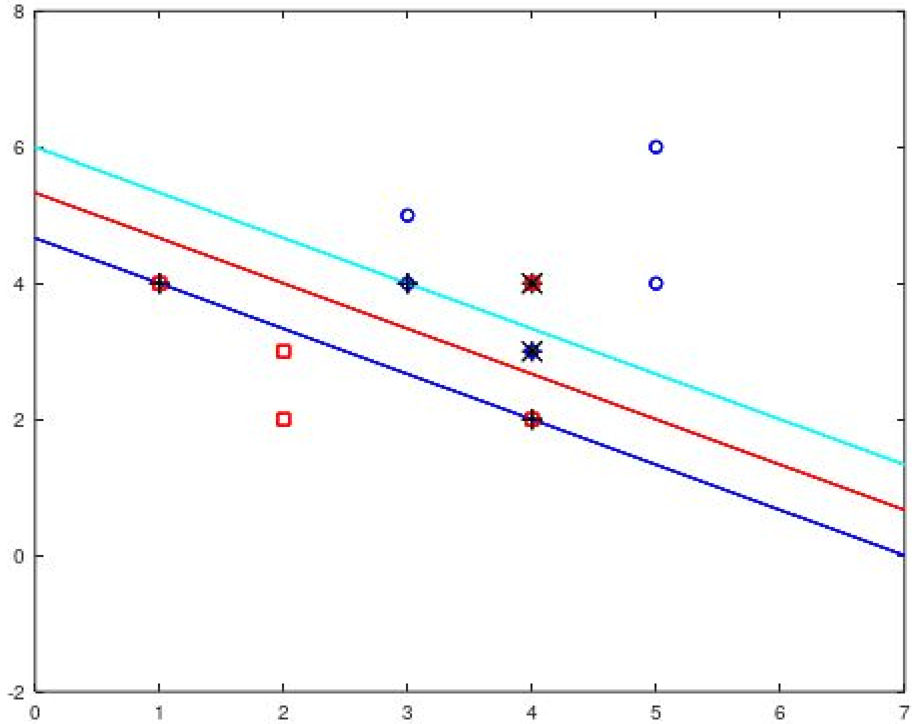
4. **Representar gráficamente los vectores de entrenamiento, marcando los que son vectores soporte, y la recta separadora correspondiente.**
Como se puede observar en ambas gráficas, los vectores soporte están marcados con una cruz +.

LINEALMENTE SEPARABLE



NO LINEALMENTE SEPARABLE

Como nos piden más adelante, hemos marcado directamente en la gráfica los vectores soporte erróneos con una x (C=1000).



Además, para el caso no linealmente separable, se nos pide:

5. **Determinar los valores de tolerancia de margen, ζ, asociados a cada dato de entrenamiento.**

La tolerancia de margen ζ_m para un vector soporte m ∈ V se calcula como:

$$\theta_0 = c_m - \boldsymbol{\theta}^t \mathbf{x}_m$$

Esta fórmula se traduce en octave como:

```
for j=1:k
     tolerancia=[tolerancia 1-(signos(j)*aux(j))];
   endfor
```

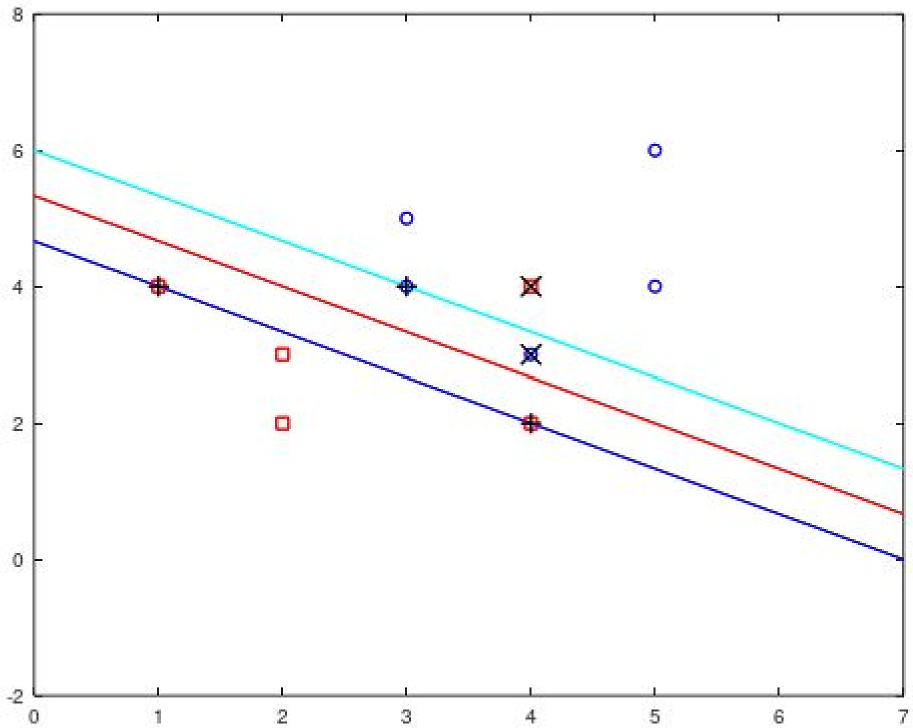
Obtenemos los siguientes valores para la tolerancia:

tolerancia =

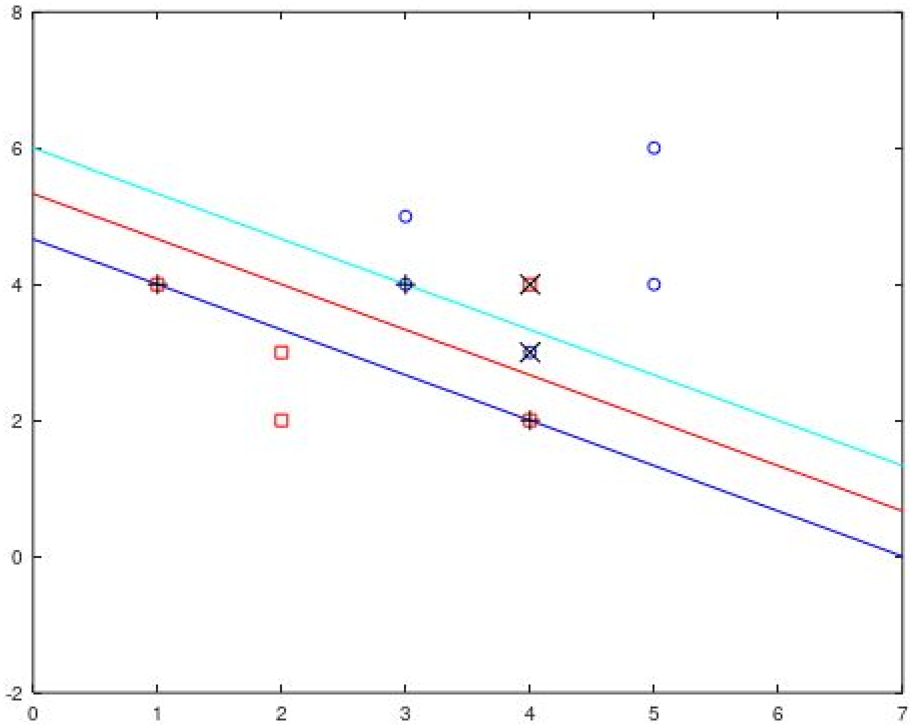
0.00000 -0.00090   2.99865   0.00090   0.50113

6. **Obtener los resultados (graficas) indicados anteriormente para diversos valores relevantes de C**

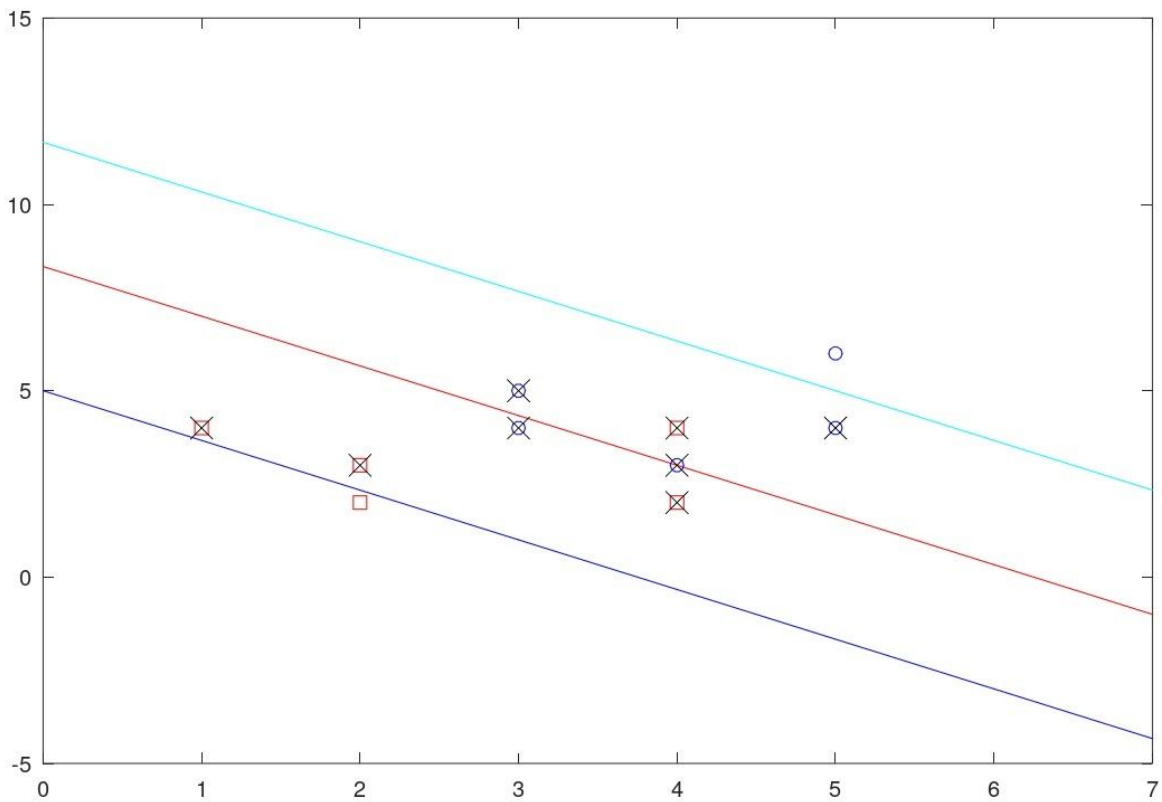
   Para C=1000 la gráfica es la que hemos obtenido anteriormente.
   Para C=9999 la gráfica es la siguiente:



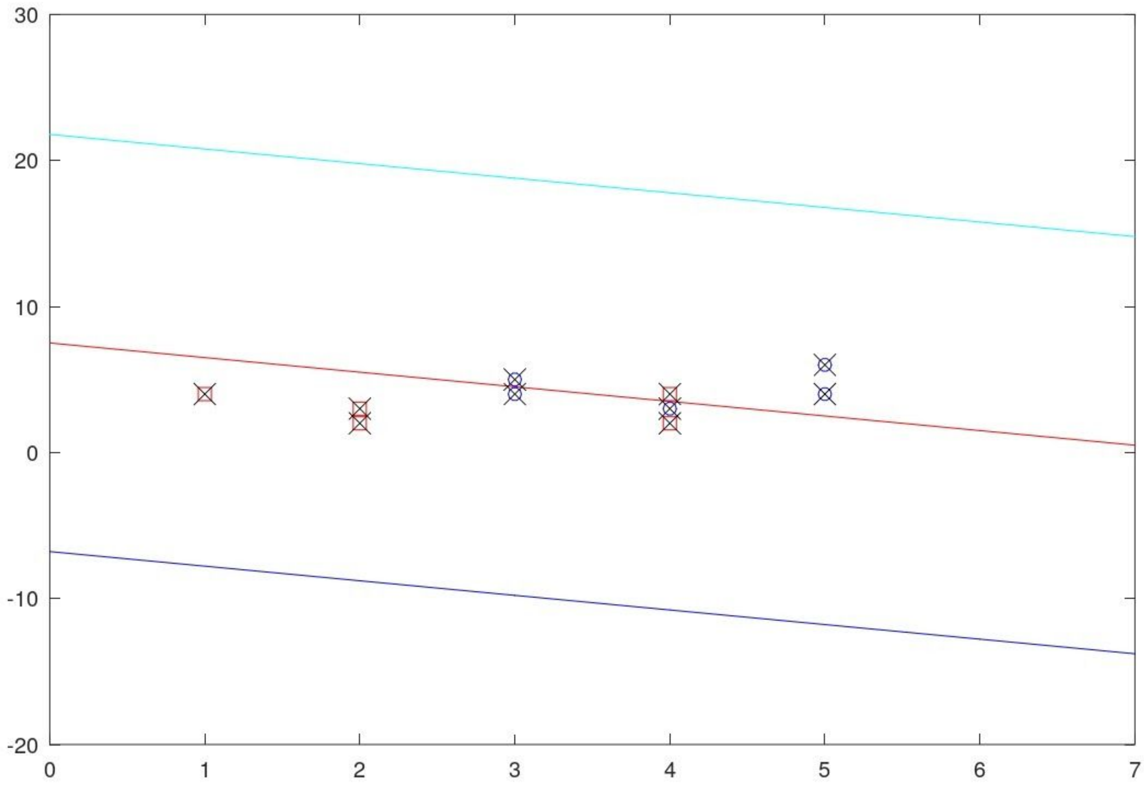
   Para C=50 la gráfica es la siguiente:



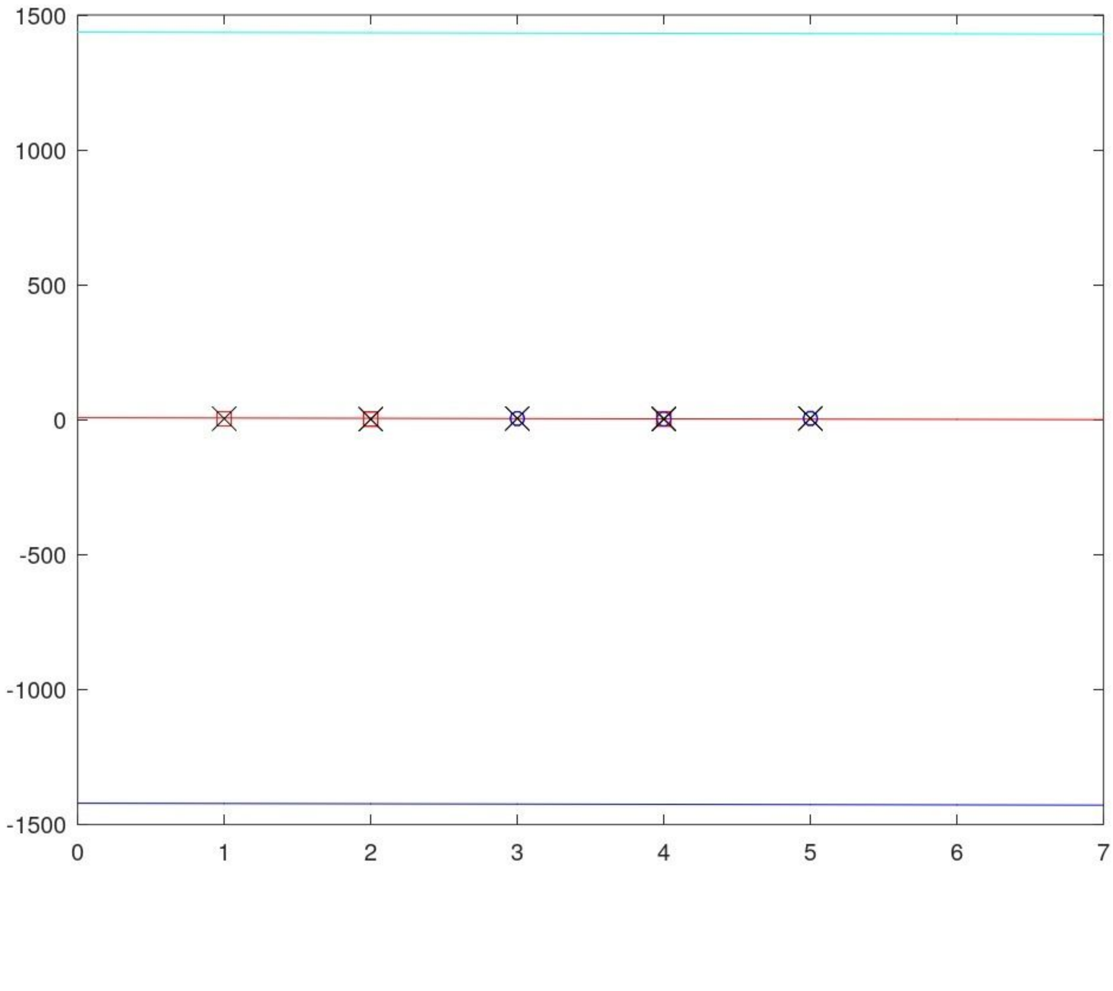
Para C=0.1, la gráfica es la siguiente:



Para C=0.01, la gráfica es la siguiente:



Para c=0.0001, la gráfica es la siguiente:



El segundo ejercicio para entregar trata sobre realizar diferentes pruebas de SVM sobre MNIST para obtener el kernel y el valor de C que obtienen mejores resultados. Para ello, debemos crear una tabla donde se muestran diversos resultados del error con un intervalo de confianza del 95% que permiten visualizar la tendencia del modelo mejorar o empeorar según varían los parámetros considerados.

Vamos a realizar pruebas sólo con el kernel polinomial, puesto que el resto de kernels, tras varios intentos de ejecución, no nos terminan en un tiempo aceptable.

ERROR(%)	C=50	C=9999
d=2	1.95±0.27%	1.95±0.27%
d=3	2.09±0.28%	2.09±0.28%
d=4	2.63±0.31%	2.63±0.31%
d=5	3.35%±0.35%	3.35%±0.35%
d=6	4.12%±0.39%	4.12%±0.39%
d=7	5.27%±0.44%	5.27%±0.44%
d=8	6.36%±0.48%	6.36%±0.48%
d=9	7.42%±0.51%	7.42%±0.51%

Como podemos observar, los resultados no varían con respecto al parámetro C. Sin embargo, los resultados tienden a empeorar cuando el grado del polinomio aumenta.