APRENDIZAJE AUTOMÁTICO
Memoria de SVM

- Fabián Scherle Carboneres
- Lishuang Sun (María)

1. En el subdirectorio data/mini se encuentran dos pequeños conjuntos de datos de entrenamiento en dos dimensiones: (trSep.dat, trSeplabels.dat) y (tr.dat, trlabels.dat). El primero es linealmente separable (no es necesario kernel) y el segundo no. Para cada uno de estos conjuntos4:

- **Obtén los SVM sin kernel (es decir, kernel tipo lineal). Para simular la optimización estándar del caso separable basta usar un valor grande de C (C = 1000).**
- **Determina:**
  - **a) los multiplicadores de Lagrange, asociados a cada dato de entrenamiento.**
  - **b) los vectores soporte.**
  - **c) el vector de pesos y umbral de la función discriminante lineal.**
  - **d) el margen correspondiente.**
- **Calcula los parámetros de la frontera lineal (recta) de separación;**
- **Representa gráficamente los vectores de entrenamiento, marcando los que son vectores soporte, y la recta separadora correspondiente.**

Ademas, para el conjunto no-separable utilizando diversos valores relevantes de C:

- **Determina los valores de tolerancia de margen, , asociados a cada dato de entrenamiento.**
- **Marca los vectores soporte “erróneos" en la representación grafica.**

Datos linealmente separables:

```
load data/mini/trSep.dat
load data/mini/trSeplabels.dat
res = svmtrain(xl, X, “-t 0 -c 1000”);
```

a)

abs(res.sv_coef) = **[**0.87472, 0.74989, 1.62461**]**

Utilizamos *abs* debido a que la clase 2 se interpreta como -1 y en *sv_coef* se tiene la multiplicación de la α por la etiqueta de clase.

b)

full(res.SVs) = [1 4; 4 2; 3 4]

c)

theta = res.sv_coef’ * res.SVs = [-0.99955 -1.49978]

Escogemos cualquier vector soporte tal que 0 < α < C. Para la muestra 1:

theta0 = sign(res.sv_coef(1)) - theta*res.SVs(1,:)’ = 7.9987

d)

Margen = 2 / norm(theta) = 1.1097

Frontera lineal de separación: 7.9987 - 0.99955 * x1 - 1.49978 * x2 = 0

x1 = [0:7];

x2 = -theta(1)/theta(2)*x1 - theta0/theta(2);

Ecuaciones Margen:

7.9987 - 0.99955 * x1 - 1.49978 * x2 = +1

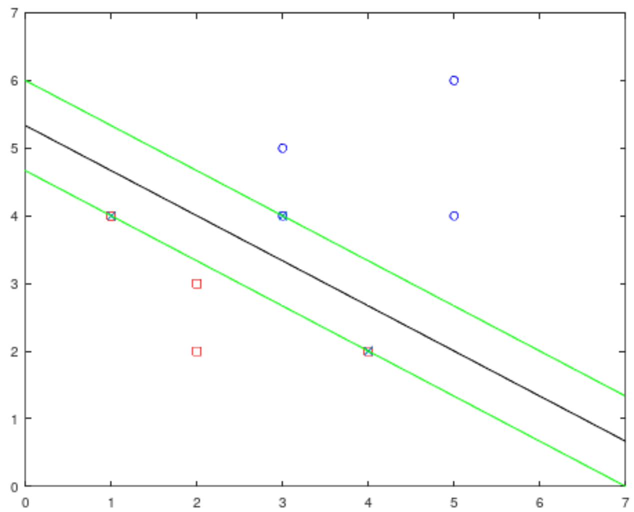
7.9987 - 0.99955 * x1 - 1.49978 * x2 = -1

x3 = -theta(1)/theta(2)*x1 - theta0/theta(2) - 1/theta(2);

x4 = -theta(1)/theta(2)*x1 - theta0/theta(2) + 1/theta(2);

Representación gráfica:

plot(X(xl==1),X(xl==1,2),"sr",X(xl==2,1),X(xl==2,2),"ob", X(res.sv_indices,1), X(res.sv_indices,2),"x",x1,x2,"-k",x1,x3,"-g",x1,x4,"-g"); axis([0 7 0 7])



Datos linealmente no separables:

(falta dibujar las rectas separadoras y marcar los SV erróneos)

load data/mini/tr.dat

load data/mini/trlabels.dat

C = 1

res = svmtrain(xl, X, “-t 0 -c 1”)

a)

abs(res.sv_coef) = [0.65306, 0.73472, 1, 1, 0.38778, 1]

b)

full(res.SVs) = [1 4; 4 2; 4 4; 3 4; 3 5; 4 3]

c)

theta = res.sv_coef’ * res.SVs = [-0.57139 -0.85722]

theta0 = sign(res.sv_coef(1)) - theta*res.SVs(1,:)’ = 5.0003

d)

Margen = 2/norm(theta) = 1.9414

Frontera lineal de separación: 5.0003 - 0.57139 * x1 - 0.85722 * x2 = 0

x1 = [0:7];
x2 = -theta(1)/theta(2)*x1 - theta0/theta(2);

Ecuaciones Margen:

5.0003 - 0.57139 * x1 - 0.85722 * x2 = +1
5.0003 - 0.57139 * x1 - 0.85722 * x2 = -1

x3 = -theta(1)/theta(2)*x1 - theta0/theta(2) - 1/theta(2);
x4 = -theta(1)/theta(2)*x1 - theta0/theta(2) + 1/theta(2);

Tolerancia de margen:

```
T = [];
for n=1:rows(res.sv_coef)
if abs(res.sv_coef(n)) == 1
T(n) = 1 - sign(res.sv_coef(n))*(theta*X(res.sv_indices(n),:)'+theta0');
else  T(n) = 0;
endif
endfor
```

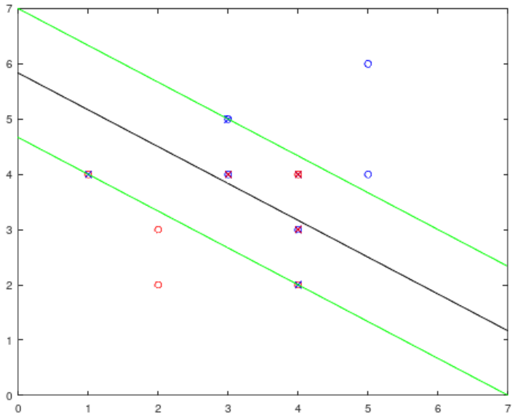
n=3 → $\zeta 9 = 1.171418$
n=4 → $\zeta 5 = 0.85722$
n=6 → $\zeta 10 = 1.14304$

Las demás muestras tienen $\zeta n = 0$.

Aquellos vectores cuyas tolerancias de margen ζ son diferente de 0, tienen error de clasificación si la tolerancia de margen $\zeta > 1$ y dentro del margen si $\zeta <= 1$. Por lo tanto los vectores soporte erróneos son las muestras 5, 9 y 10.

Representación gráfica:

```
plot(X(xl==1),X(xl==1,2),"or",X(xl==2,1),X(xl==2,2),"ob", X(res.sv_indices,1),
X(res.sv_indices,2),"xb",x1,x2,"-k",x1,x3,"-g",x1,x4,"-g",X([5,9,10],1),
X([5,9,10],2),"xr"); axis([0 7 0 7])
```



C = 10

res = svmtrain(xl, X, "-t 0 -c 10")

a)

abs(res.sv_coef) = [3.3749, 5.7498, 10, 9.1247, 10]

b)

full(res.SVs) = [1 4; 4 2; 4 4; 3 4; 4 3]

c)

theta = res.sv_coef' * res.SVs = [-1 -1.4996]
theta0 = sign(res.sv_coef(1)) - theta*res.SVs(1,:)' = 7.9986

d)

Margen = 2/norm(theta) = 1.1096

Frontera lineal de separación: 7.9986 - 1 * x1 - 1.4996 * x2 = 0

x1 = [0:7];
x2 = -theta(1)/theta(2)*x1 - theta0/theta(2);

Ecuaciones Margen:

7.9986 - 1 * x1 - 1.4996 * x2 = +1
7.9986 - 1 * x1 - 1.4996 * x2 = -1
x3 = -theta(1)/theta(2)*x1 - theta0/theta(2) - 1/theta(2);
x4 = -theta(1)/theta(2)*x1 - theta0/theta(2) + 1/theta(2);

Tolerancia de margen:

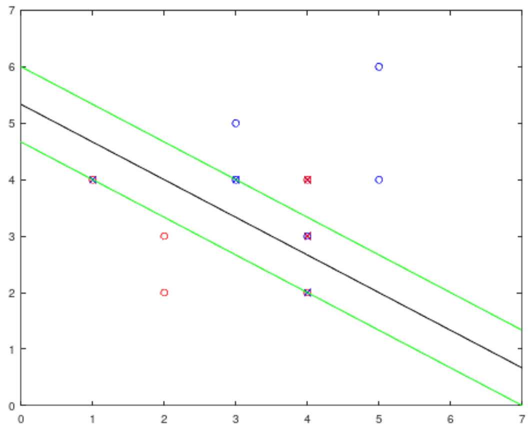
```
T = [];
for n=1:rows(res.sv_coef)
if abs(res.sv_coef(n)) == 10
T(n) = 1 - sign(res.sv_coef(n))*(theta*X(res.sv_indices(n),:)'+theta0');
else  T(n) = 0;
endif
endfor
```

n=3 → $\zeta 9 = 3$
n=5 → $\zeta 10 = 0.49965$
Las demás muestras tienen $\zeta n = 0$.

Aquellos vectores cuyas tolerancias de margen ζ son diferente de 0, tienen error de clasificación si la tolerancia de margen $\zeta > 1$ y dentro del margen si $\zeta <= 1$. Por lo tanto los vectores soporte erróneos son las muestras 9 y 10.

Representación gráfica:

```
plot(X(xl==1),X(xl==1,2),"or",X(xl==2,1),X(xl==2,2),"ob", X(res.sv_indices,1),
X(res.sv_indices,2),"xb",x1,x2,"-k",x1,x3,"-g",x1,x4,"-g",X([9,10],1), X([9,10],2),"xr");
axis([0 7 0 7])
```



C = 100

res = svmtrain(xl, X, "-t 0 -c 100")

a)

abs(res.sv_coef) = [25.875, 50.750, 100, 76.625, 100]

b)

full(res.SVs) = [1 4; 4 2; 4 4; 3 4; 4 3];

c)

theta = res.sv_coef' * res.SVs = [-0.99955 -1.49978]
theta0 = sign(res.sv_coef(1)) - theta*res.SVs(1,:)' = 7.9987

d)

Margen = 2/norm(theta) = 1.1097

Frontera lineal de separación: 7.9987 -0.99955*x1 - 1.49978* x2 = 0

x1 = [0:7];
x2 = -theta(1)/theta(2)*x1 - theta0/theta(2);

Ecuaciones Margen:

7.9987 -0.99955*x1 - 1.49978* x2 = +1
7.9987 -0.99955*x1 - 1.49978* x2 = -1

x3 = -theta(1)/theta(2)*x1 - theta0/theta(2) - 1/theta(2);
x4 = -theta(1)/theta(2)*x1 - theta0/theta(2) + 1/theta(2);

Tolerancia de margen:

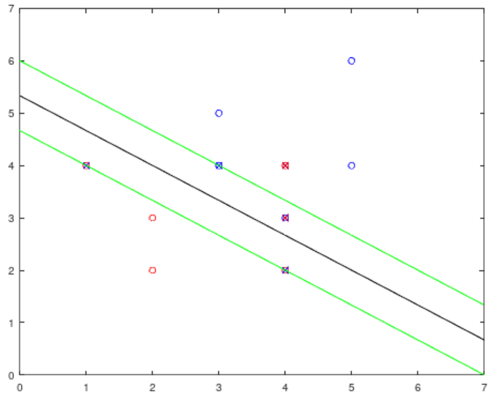
```
T = [];
for n=1:rows(res.sv_coef)
if abs(res.sv_coef(n)) == 100
T(n) = 1 - sign(res.sv_coef(n))*(theta*X(res.sv_indices(n),:)'+theta0');
else  T(n) = 0;
endif
endfor
```

n=3 → ζ_9 = 2.99866
n=5 → ζ_{10} = 0.50112
Las demás muestras tienen ζ_n = 0.

Aquellos vectores cuyas tolerancias de margen ζ son diferente de 0, tienen error de clasificación si la tolerancia de margen ζ > 1 y dentro del margen si ζ <= 1. Por lo tanto los vectores soporte erróneos son las muestras 9 y 10.

Representación gráfica:

```
plot(X(xl==1),X(xl==1,2),"or",X(xl==2,1),X(xl==2,2),"ob", X(res.sv_indices,1),
X(res.sv_indices,2),"xb",x1,x2,"-k",x1,x3,"-g",x1,x4,"-g",X([9,10],1), X([9,10],2),"xr");
axis([0 7 0 7])
```



<u>C = 1000</u>

res = svmtrain(xl, X, "-t 0 -c 1000")

a)

abs(res.sv_coef) = [250.87, 500.75, 1000, 751.62, 1000]

b)

full(res.SVs) = [1 4; 4 2; 4 4; 3 4; 4 3]

c)

theta = res.sv_coef' * res.SVs = [-0.99955 -1.49977]
theta0 = sign(res.sv_coef(1)) - theta*res.SVs(1,:)' = 7.9986

d)

Margen = 2/norm(theta) = 1.1097

Frontera lineal de separación: 7.9986 - 0.99955 * x1 - 1.49977 * x2 = 0

x1 = [0:7];
x2 = -theta(1)/theta(2)*x1 - theta0/theta(2);

Ecuaciones Margen:

7.9986 - 0.99955 * x1 - 1.49977 * x2 = +1

7.9986 - 0.99955 * x1 - 1.49977 * x2 = -1

```
x3 = -theta(1)/theta(2)*x1 - theta0/theta(2) - 1/theta(2);
x4 = -theta(1)/theta(2)*x1 - theta0/theta(2) + 1/theta(2);
```

Tolerancia de margen:

```
T = [];
for n=1:rows(res.sv_coef)
if abs(res.sv_coef(n)) == 1000
T(n) = 1 - sign(res.sv_coef(n))*(theta*X(res.sv_indices(n),:)'+theta0');
else  T(n) = 0;
endif
endfor
```

n=3 → $\zeta 9 = 2.9986$

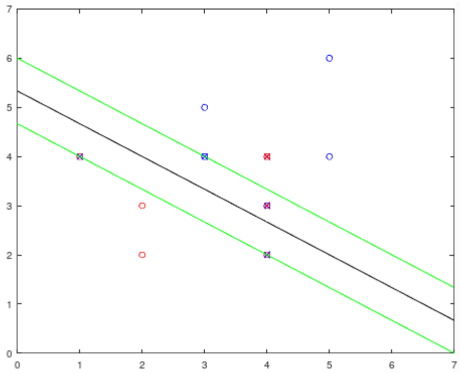
n=5 → $\zeta 10 = 0.50113$

Las demás muestras tienen $\zeta n = 0$.

Aquellos vectores cuyas tolerancias de margen ζ son diferente de 0, tienen error de clasificación si la tolerancia de margen $\zeta > 1$ y dentro del margen si $\zeta <= 1$. Por lo tanto los vectores soporte erróneos son las muestras 9 y 10.

Representación gráfica:

```
plot(X(xl==1),X(xl==1,2),"or",X(xl==2,1),X(xl==2,2),"ob", X(res.sv_indices,1),
X(res.sv_indices,2),"xb",x1,x2,"-k",x1,x3,"-g",x1,x4,"-g",X([9,10],1), X([9,10],2),"xr");
axis([0 7 0 7])
```



2. Realiza un experimento donde se evalúe el error de clasificación en función de los parámetros del clasificador basado en SVM. Mas concretamente, explora los valores del parámetro C (-c 1, 10, 100...) y el tipo de de kernel (-t 0, 1, 2, 3). Para aquellos tipos de kernel que lo permitan, explora sus parámetros específicos. Realiza la exploración de parámetros descrita utilizando un conjunto limitado valores de PCA (D = 50,100,200).

Para realizar dicho experimento creamos un pca+script svm-exp.m al cual le pasamos distintos valores de de D (dimensiones), C, K (kernel), para el caso del kernel polinomial, le pasamos diferentes valores de G (grado), para el caso del kernel radial y sigmoid diferentes valores de gamma y usamos un 9% de las muestras para entrenar el clasificador y un 1% para test. Obteniendo los siguientes resultados:

Dimension	Kernel	C	Grado	Gamma	Error
50	Lineal	1	-	-	8,333
50	Lineal	10	-	-	8,5
50	Lineal	100	-	-	9,5

50	Polinomial	1	1	-	9,5
50	Polinomial	1	2	-	32,167
50	Polinomial	1	3	-	83,667
50	Polinomial	1	4	-	87
50	Polinomial	1	5	-	87,5
50	Polinomial	10	1	-	8,333
50	Polinomial	10	2	-	7,833
50	Polinomial	10	3	-	50,833
50	Polinomial	10	4	-	81
50	Polinomial	10	5	-	86,167
50	Polinomial	100	1	-	8,667
50	Polinomial	100	2	-	6
50	Polinomial	100	3	-	9
50	Polinomial	100	4	-	59,667
50	Polinomial	100	5	-	78,167
50	Radial	1	-	0,1	3,833
50	Radial	1	-	0,01	8,833
50	Radial	1	-	0,001	18
50	Radial	1	-	0,0001	86,5
50	Radial	1	-	1E-05	87,667
50	Radial	10	-	0,1	4
50	Radial	10	-	0,01	6,5
50	Radial	10	-	0,001	9,5
50	Radial	10	-	0,0001	18
50	Radial	10	-	1E-05	86,5
50	Radial	100	-	0,1	4
50	Radial	100	-	0,01	5,667
50	Radial	100	-	0,001	7,833
50	Radial	100	-	0,0001	9,5
50	Radial	100	-	1E-05	18
50	Sigmoid	1	-	0,1	14,333
50	Sigmoid	1	-	0,01	10,833
50	Sigmoid	1	-	0,001	27,833
50	Sigmoid	1	-	0,0001	87,667
50	Sigmoid	1	-	1E-05	87,667
50	Sigmoid	10	-	0,1	17,333
50	Sigmoid	10	-	0,01	8,833
50	Sigmoid	10	-	0,001	10,833
50	Sigmoid	10	-	0,0001	27,833
50	Sigmoid	10	-	1E-05	87,667
50	Sigmoid	100	-	0,1	18,833
50	Sigmoid	100	-	0,01	8,333
50	Sigmoid	100	-	0,001	8,667
50	Sigmoid	100	-	0,0001	10,833
50	Sigmoid	100	-	1E-05	27,833
100	Lineal	1	-	-	9,167
100	Lineal	10	-	-	9,667
100	Lineal	100	-	-	11,5
100	Polinomial	1	1	-	10,333
100	Polinomial	1	2	-	77,833
100	Polinomial	1	3	-	87,667
100	Polinomial	1	4	-	87,667

100	Polinomial	1	5	-	87,667
100	Polinomial	10	1	-	8,833
100	Polinomial	10	2	-	13,333
100	Polinomial	10	3	-	81
100	Polinomial	10	4	-	87,333
100	Polinomial	10	5	-	87,667
100	Polinomial	100	1	-	9,167
100	Polinomial	100	2	-	6,333
100	Polinomial	100	3	-	44,167
100	Polinomial	100	4	-	83,5
100	Polinomial	100	5	-	86,833
100	Radial	1	-	0,1	4
100	Radial	1	-	0,01	8,667
100	Radial	1	-	0,001	17,167
100	Radial	1	-	0,0001	86,5
100	Radial	1	-	1E-05	87,667
100	Radial	10	-	0,1	4
100	Radial	10	-	0,01	7
100	Radial	10	-	0,001	9,167
100	Radial	10	-	0,0001	17,5
100	Radial	10	-	1E-05	86,5
100	Radial	100	-	0,1	4
100	Radial	100	-	0,01	6,667
100	Radial	100	-	0,001	8,333
100	Radial	100	-	0,0001	9,167
100	Radial	100	-	1E-05	17,5
100	Sigmoid	1	-	0,1	14,167
100	Sigmoid	1	-	0,01	10,333
100	Sigmoid	1	-	0,001	27,333
100	Sigmoid	1	-	0,0001	87,667
100	Sigmoid	1	-	1E-05	87,667
100	Sigmoid	10	-	0,1	17,5
100	Sigmoid	10	-	0,01	8,833
100	Sigmoid	10	-	0,001	10,333
100	Sigmoid	10	-	0,0001	27,333
100	Sigmoid	10	-	1E-05	87,667
100	Sigmoid	100	-	0,1	18,833
100	Sigmoid	100	-	0,01	8,5
100	Sigmoid	100	-	0,001	8,833
100	Sigmoid	100	-	0,0001	10,333
100	Sigmoid	100	-	1E-05	27,333
200	Lineal	1	-	-	8,667
200	Lineal	10	-	-	10
200	Lineal	100	-	-	10,5
200	Polinomial	1	1	-	12,833
200	Polinomial	1	2	-	87,5
200	Polinomial	1	3	-	87,667
200	Polinomial	1	4	-	87,667
200	Polinomial	1	5	-	87,667
200	Polinomial	10	1	-	8,833
200	Polinomial	10	2	-	49,667
200	Polinomial	10	3	-	87,5

200	Polinomial	10	4	-	87,667
200	Polinomial	10	5	-	87,667
200	Polinomial	100	1	-	8,5
200	Polinomial	100	2	-	8
200	Polinomial	100	3	-	79,5
200	Polinomial	100	4	-	87,5
200	Polinomial	100	5	-	87,667
200	Radial	1	-	0,1	4,167
200	Radial	1	-	0,01	8,667
200	Radial	1	-	0,001	17,333
200	Radial	1	-	0,0001	86,5
200	Radial	1	-	1E-05	87,667
200	Radial	10	-	0,1	3,667
200	Radial	10	-	0,01	7,333
200	Radial	10	-	0,001	9,5
200	Radial	10	-	0,0001	17,5
200	Radial	10	-	1E-05	86,5
200	Radial	100	-	0,1	3,667
200	Radial	100	-	0,01	6,5
200	Radial	100	-	0,001	8,667
200	Radial	100	-	0,0001	9,5
200	Radial	100	-	1E-05	17,5
200	Sigmoid	1	-	0,1	13,5
200	Sigmoid	1	-	0,01	10,5
200	Sigmoid	1	-	0,001	27,167
200	Sigmoid	1	-	0,0001	87,667
200	Sigmoid	1	-	1E-05	87,667
200	Sigmoid	10	-	0,1	17,5
200	Sigmoid	10	-	0,01	8,333
200	Sigmoid	10	-	0,001	10,333
200	Sigmoid	10	-	0,0001	27,167
200	Sigmoid	10	-	1E-05	87,667
200	Sigmoid	100	-	0,1	19,333
200	Sigmoid	100	-	0,01	8,833
200	Sigmoid	100	-	0,001	8,5
200	Sigmoid	100	-	0,0001	10,333
200	Sigmoid	100	-	1E-05	27,167

Se puede apreciar que el mínimo error lo obtiene la dimensión 50 con un kernel radial y gamma de 0,1.

3. Una vez determinado los valores óptimos de los parámetros del clasificador basado en SVM, entrena y evalúa un clasificador final en los conjuntos oficiales MNIST de entrenamiento y test, respectivamente. Discute los resultados obtenidos comparándolos con los obtenidos con el clasificador de mixtura de gaussianas y con los reportados en la tarea MNIST, especialmente los basados en SVM..

Tomando como base los conjuntos de parámetros óptimos obtenidos en el apartado anterior evaluamos el conjunto de muestra de test:

Dimension	Kernel	C	Grado	Gamma	Error	Intervalos
50	2	1	-	0,1	2,02	[1,744 - 2,296]

Podemos concluir que el porcentaje de error no ha variado respecto al obtenido en el apartado anterior y que es peor comparándolo con el clasificador de mixturas gaussianas (1.62%).

En relación a las tasas de error de la MNIST web (http://yann.lecun.com/exdb/mnist/) comprobamos que se acerca a la tasa de error del clasificador de SVM con kernel gaussiano.