Homework 5: Support Vector Machines

TDA231 - Algorithms for Machine Learning & Inference

Theodor Åstrand, theast@student.chalmers.se, 931109-9114
Linnéa Otterlind, linott@student.chalmers.se, 921126-3620

Chalmers University of Technology

February, 2016

Problem 1.1

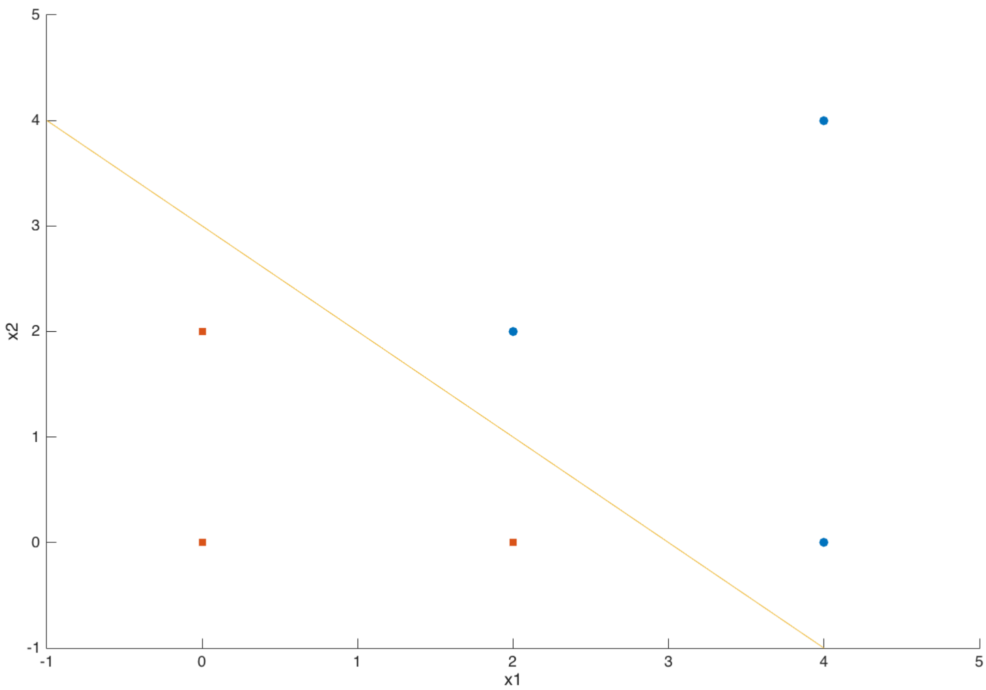

Figure 1: Plot with training points and hyperplane.

It is clear from the way the points are scattered in Figure 1 that the dividing line has to pass exactly in between the points $(0,2)$ and $(2,2)$ and the points $(2,0)$ and $(4,0)$ to maximize γ. Thus we can see that the resulting line $w_1x_1 + w_2x_2 + b = 0$ has to cross the x_1-axis in $(3,0)$ and the x_2-axis in $(0,3)$ so the line becomes $x_1 + x_2 - 3 = 0$. Which means $w_1 = 1$, $w_2 = 1$ and $b = -3$.

The distance between either of the points $(0,2)$, $(2,2)$, $(2,0)$ and $(4,0)$ and the line is equal. Choosing $(a_1, a_2) = (0,2)$ we get:

$\gamma = |w_1a_1 + w_2a_2 + b|/\sqrt{w_1^2 + w_2^2} = |2w_2 + b|/\sqrt{w_1^2 + w_2^2} = |2w_2 - 3|/\sqrt{w_1^2 + w_2^2} =$

$= |2-3|/\sqrt{1^2 + 1^2} = 1/\sqrt{2} \Leftrightarrow$

$2\gamma = \sqrt{2}$

Problem 1.2

(a)

$\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$

$\mathbf{x}_1 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$, $\mathbf{x}_2 = \begin{bmatrix} 4 \\ 4 \end{bmatrix}$, $\mathbf{x}_3 = \begin{bmatrix} 4 \\ 0 \end{bmatrix}$, $\mathbf{x}_4 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $\mathbf{x}_5 = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$, $\mathbf{x}_6 = \begin{bmatrix} 0 \\ 2 \end{bmatrix}$

$t_1 = 1,\ t_2 = 1,\ t_3 = 1,\ t_4 = -1,\ t_5 = -1,\ t_6 = -1$

$\underset{\mathbf{w}}{\operatorname{argmin}}\ \frac{1}{2}\mathbf{w}^T\mathbf{w}$

subject to

$t_1(\mathbf{w}^T\mathbf{x}_1 + b) \geq 1$

$t_2(\mathbf{w}^T\mathbf{x}_2 + b) \geq 1$

$t_3(\mathbf{w}^T\mathbf{x}_3 + b) \geq 1$

$t_4(\mathbf{w}^T\mathbf{x}_4 + b) \geq 1$

$t_5(\mathbf{w}^T\mathbf{x}_5 + b) \geq 1$

$t_6(\mathbf{w}^T\mathbf{x}_6 + b) \geq 1$

Inserting the values of all $\mathbf{x}_i$ and t_i we get:

$\underset{\mathbf{w}}{\operatorname{argmin}}\ \frac{w_1^2 + w_2^2}{2}$

subject to

$2w_1 + 2w_2 + b \geq 1$

$4w_1 + 4w_2 + b \geq 1$

$4w_1 + b \geq 1$

$-b \geq 1$

$-(2w_1 + b) \geq 1$

$-(2w_2 + b) \geq 1$

(b)

$\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, $b = -3$

(c)

$\mathbf{x}_1 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$, $\mathbf{x}_2 = \begin{bmatrix} 4 \\ 4 \end{bmatrix}$, $\mathbf{x}_3 = \begin{bmatrix} 4 \\ 0 \end{bmatrix}$, $\mathbf{x}_4 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $\mathbf{x}_5 = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$, $\mathbf{x}_6 = \begin{bmatrix} 0 \\ 2 \end{bmatrix}$

$t_1 = 1$, $t_2 = 1$, $t_3 = 1$, $t_4 = -1$, $t_5 = -1$, $t_6 = -1$

$\underset{\alpha}{\text{argmax}}\ \Sigma_{i=1}^{6}\alpha_i - \frac{1}{2}\Sigma_{i=1}^{6}\Sigma_{j=1}^{6}\alpha_i\alpha_j t_i t_j \mathbf{x}_i^T\mathbf{x}_j$

subject to

$\Sigma_{i=1}^{6}\alpha_i t_i = 0$

$\alpha \geq 0$

Since $\mathbf{x}_2$ and $\mathbf{x}_4$ are not support vectors $\alpha_2 = 0$ and $\alpha_4 = 0$. Given this and the values of all $\mathbf{x}_i$ and t_i we get:

$\underset{\alpha}{\text{argmax}}\ \alpha_1 + \alpha_3 + \alpha_5 + \alpha_6 - 4\alpha_1^2 - 8\alpha_3^2 - 2\alpha_5^2 - 2\alpha_6^2 - 8\alpha_1\alpha_3 + 4\alpha_1\alpha_5 + 4\alpha_1\alpha_6 + 8\alpha_3\alpha_5$

subject to

$\alpha_1 + \alpha_3 - \alpha_5 - \alpha_6 = 0$

$\alpha_1 \geq 0$

$\alpha_3 \geq 0$

$\alpha_5 \geq 0$

$\alpha_6 \geq 0$

(d)

Solving the above problem yields:

$\alpha_1 = 0.75$

$\alpha_3 = 0.25$

$\alpha_5 = 0.75$

$\alpha_6 = 0.25$

As we have already mentioned $\mathbf{x}_2$ and $\mathbf{x}_4$ are not support vectors, i.e. they do not lie on either of the margins. This is why we can set $\alpha_2 = 0$ and $\alpha_4 = 0$ and thus disregard any term in the original problem that includes them.

Problem 2.1

(a)

We chose the default value of the box constraint parameter i.e. 1.

(b)

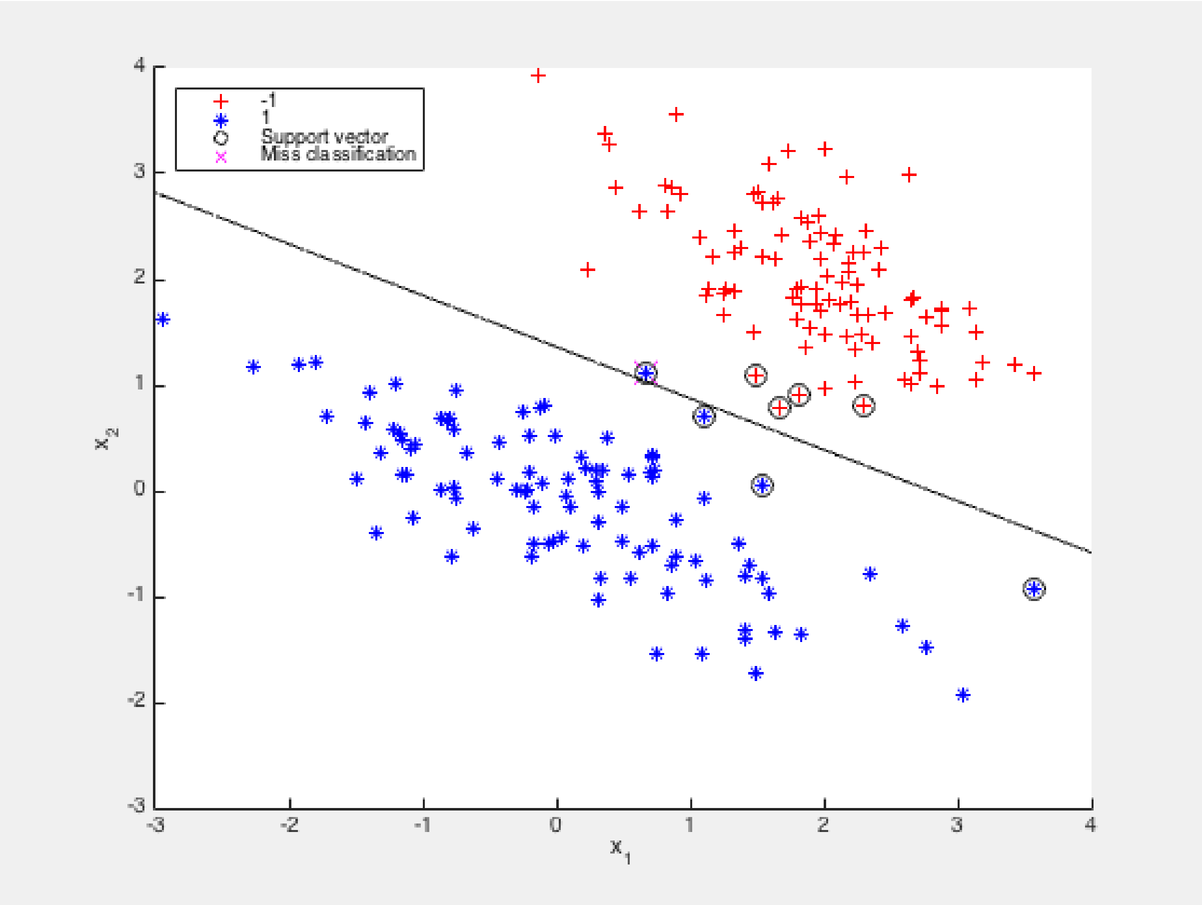

Figure 2: *svmtrain* on d1b.mat using linear kernel.

(c)

The classifier bias is -2.4528.

(d)

The soft margin is 0.9986.

Problem 2.2

(a)

The box constraint parameter C is set to 1 which yields the result shown in Figure 3.

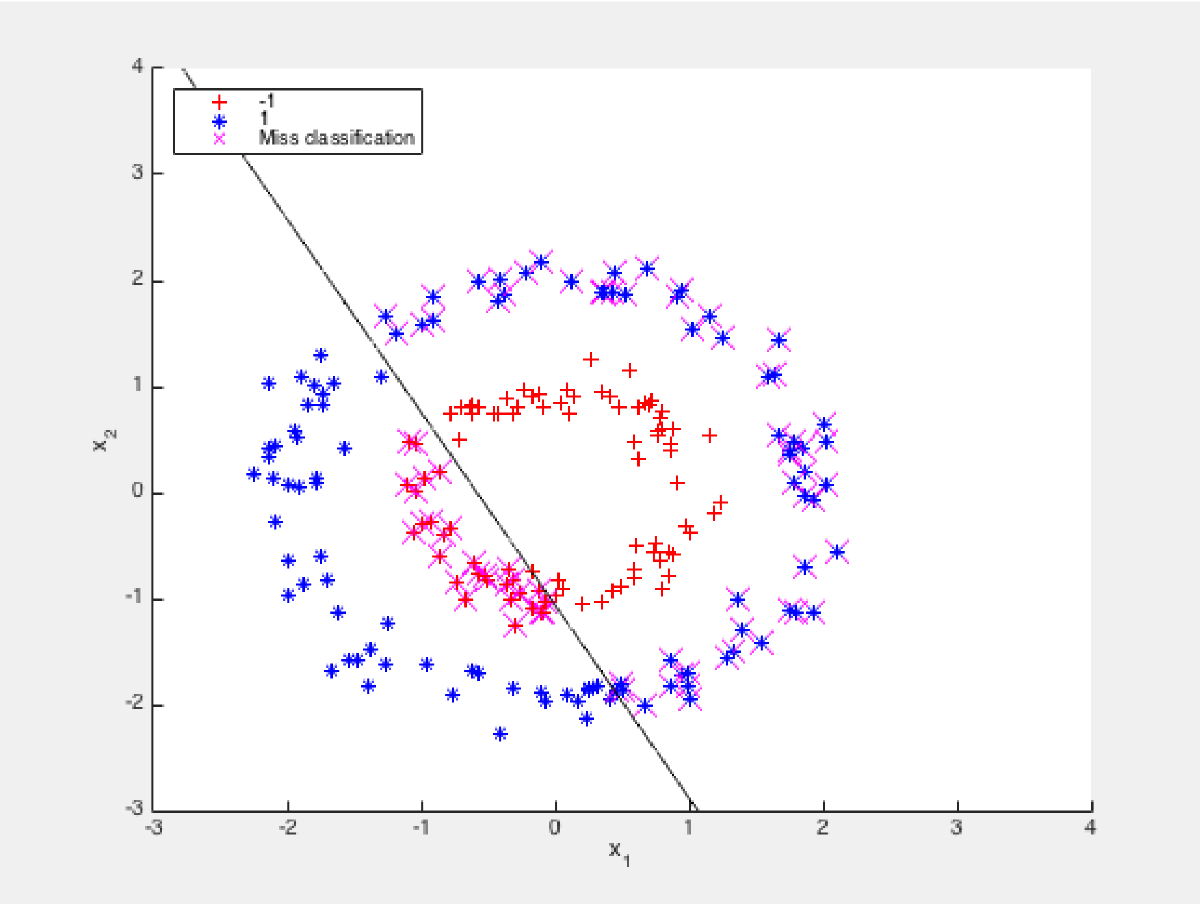

Figure 3: *svmtrain* on d2.mat using linear kernel

(b)

Kernel	Optimization	Time (sec)	Accuracy (%)
Linear	SMO	0.5926	58
Linear	QP	0.5285	61
Quadratic	SMO	0.0749	100
Quadratic	QP	0.2924	100
RBF	SMO	0.0690	100
RBF	QP	0.0670	100

We can clearly see that the linear kernel does not perform very well on the given data set. This is expected since it is quite hard to separate two circles with a linear hyperplane, thus its accuracy should be around 50%.

We can also see that quadratic and RBF gives 100% accuracy, this is also expected since these kernel methods should be able to easily separate the classes.

Finally we can see that RBF was by far the fastest kernel function.

(c)

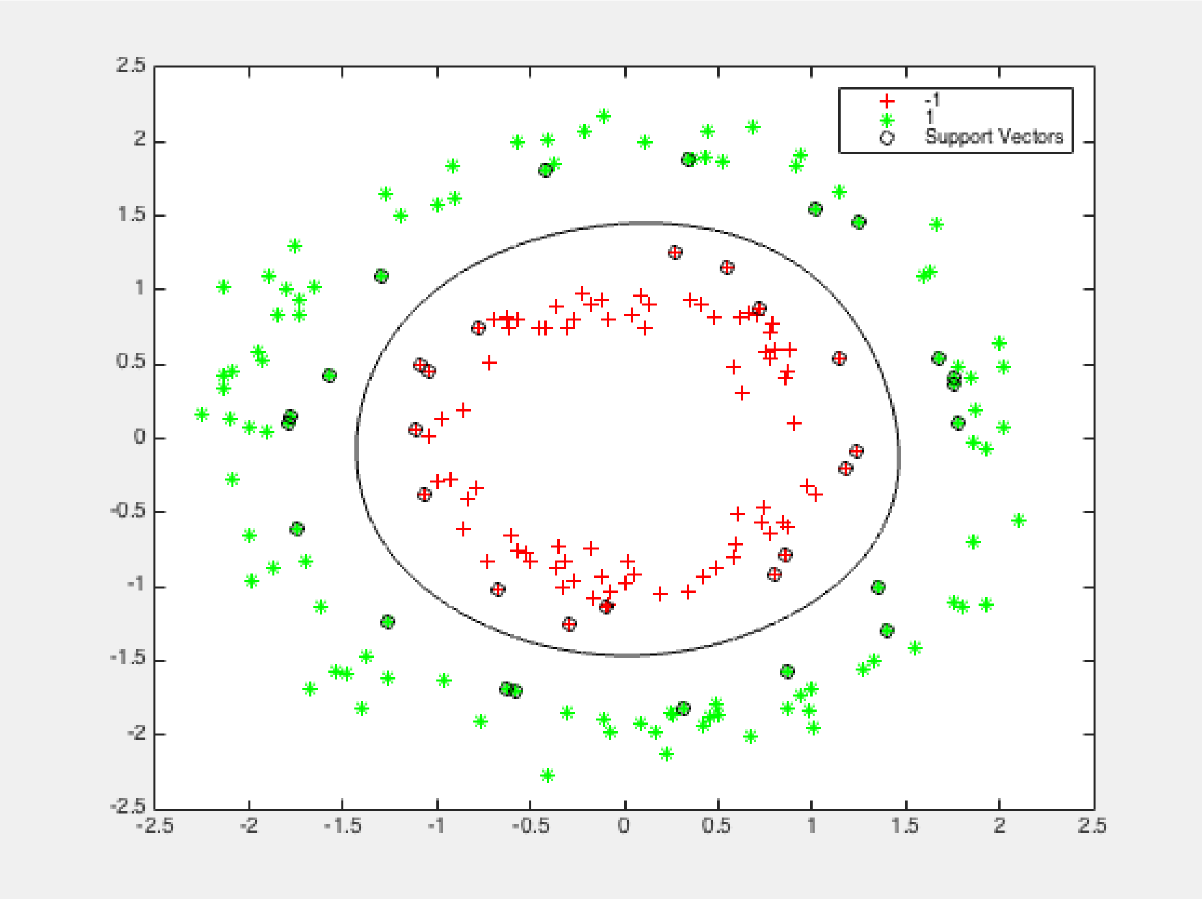

Figure 4: *svmtrain* on d2.mat using the RBF kernel