EE559

Sarthak Maharana

Homework 6

1. For SVM, the optimisation problem is as follows :

$J(\underline{w}) = \frac{1}{2} \|w\|_2^2$

subject to : $z_i(\underline{w}^T \underline{u}_i + w_0) - 1 \geq 0 \quad \forall i$

(a) Yes, if the above set of constraints ($\forall i$) is satisfied, all the training data will be classified correctly. The SVM classifier finds the closest points of two classes and chooses the hyperplane such that the margin from both is maximum.

(b) The Lagrangian function $L(\underline{w}, w_0, \underline{\lambda})$ is as follows

$$L(\underline{w}, w_0, \underline{\lambda}) = \frac{1}{2} \|w\|_2^2 - \sum_{i=1}^{N} \lambda_i \left[z_i (\underline{w}^T \underline{u}_i + w_0) - 1\right]$$

The KKT conditions are as follows :

1. $\lambda_i \geq 0 \quad \forall i$
2. $\left[z_i (w^T u_i + w_0) - 1\right] \geq 0. \quad \forall i$
3. $\lambda_i \left[z_i (w^{*T} u_i + w_0^*) - 1\right] = 0 \quad \forall i$ //

In the Lagrangian equation, $N \rightarrow$ total size of data.

(c) To derive the dual representation L_D :

(i) $\nabla_{\underline{w}} J(\underline{w}) = \frac{1}{2} \cdot 2\underline{w}^* - \sum_{i=1}^{N} \lambda_i z_i \underline{u}_i = 0$

or $\underline{w}^* = \sum_{i=1}^{N} \lambda_i z_i \underline{u}_i$ //

$J(\underline{w})$ can be written as $\frac{1}{2}\left(\sum_{i=1}^{} w_i^2\right) - \sum_{i=1}^{} \lambda_i\left[z_i\left(\underline{w}^T\underline{u}_i + w_0\right) - 1\right]$

where $\underline{w} = (w_1, w_2, \cdots w_d)$

$d \rightarrow$ dimensions

Now, $\frac{\partial J(\underline{w})}{\partial w_0} \Rightarrow \sum_{i=1}^{N} \lambda_i z_i = 0$ //

(ii) $L = \frac{1}{2}\|\underline{w}\|_2^2 - \sum_{i=1}^{N} \lambda_i\left[z_i\left(\underline{w}^T\underline{u}_i + w_0\right) - 1\right]$

By substituting $w^* = \sum_{i=1}^{N} \lambda_i z_i u_i$ and introducing L_2 norm to a summation, we get a double summation, as:

$$L_D(\underline{\lambda}) = \frac{1}{2}\|w\|_2^2 - \sum_{i=1}^{N} \lambda_i z_i w^{*T}\underline{u}_i - \sum_{i=1}^{N} \lambda_i z_i w_0 + \sum_{i=1}^{N} \lambda_i$$

$$= \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \lambda_i \lambda_j z_i z_j \underline{u}_i^T \underline{u}_j - \sum_{i=1}^{N} \lambda_i z_i\left(\sum_{j=1}^{N} \lambda_j z_j \underline{u}_j^T\right)\underline{u}_i - w_0 \sum_{i=1}^{N} \lambda_i z_i + \sum_{i=1}^{N} \lambda_i$$

$$= -\frac{1}{2}\left[\sum_{i=1}^{N}\sum_{j=1}^{N} \lambda_i \lambda_j z_i z_j \underline{u}_i^T \underline{u}_j\right] + \sum_{i=1}^{N} \lambda_i - w_0 \underbrace{\sum_{i=1}^{N} \lambda_i z_i}_{0}$$

$\rightarrow$ from $\frac{\partial L}{\partial w_0}$.

$$= -\frac{1}{2}\left[\sum_{i=1}^{N}\sum_{j=1}^{N} \lambda_i \lambda_j z_i z_j \underline{u}_i^T \underline{u}_j\right] + \sum_{i=1}^{N} \lambda_i = L_D(\lambda) \quad //$$

The KKT conditions are as follows :

1. $\lambda_i \geq 0 \quad \forall i$

2. $\sum_{i=1}^{N} \lambda_i z_i = 0$

3. $\lambda_i \left[z_i \left(w^{*T} \underline{u_i} + w_0 \right) - 1 \right] = 0$

or $\lambda_i \left[z_i \left(\sum_{j=1}^{N} \lambda_j z_j u_j^T u_i + w_0 \right) - 1 \right] = 0 \quad \forall i$

4. $z_i \left(w^{*T} u_i + w_0^* \right) - 1 \geq 0 \;\; \forall i$

In total, there are 4 KKT conditions.

2. Given, $N = 3$

(a) From 1(c), $L_D(\lambda) = -\frac{1}{2} \left[\sum_{i=1}^{N} \sum_{j=1}^{N} \lambda_i \lambda_j z_i z_j \underline{u_i}^T \underline{u_j} \right] + \sum_{i=1}^{N} \lambda_i$

Converting to Lagrangian optimisation problem by introducing the multiplier μ, we get :

$$L_D'(\underline{\lambda}, \mu) = -\frac{1}{2} \left[\sum_{i=1}^{N} \sum_{j=1}^{N} \lambda_i \lambda_j z_i z_j \underline{u_i}^T u_j \right] + \sum_{i=1}^{N} \lambda_i + \mu \left(\sum_{i=1}^{N} \lambda_i z_i \right)$$

(since we're maximising wrt $\underline{\lambda}$).

$$= -\frac{1}{2} \lambda^T \begin{bmatrix} z_1 u_1^T \\ z_2 u_2^T \\ \vdots \\ z_N u_N^T \end{bmatrix} \begin{bmatrix} z_1 u_1 & \cdots & z_N u_N \end{bmatrix} \underline{\lambda} + \underline{\lambda}^T \underline{1} + \mu \underline{\lambda}^T \begin{bmatrix} z_1 \\ \vdots \\ z_N \end{bmatrix}$$

$\underline{1} \rightarrow$ one's vector

$$\nabla_{\underline{\lambda}} L_D'(\underline{\lambda}, \mu) = -\begin{bmatrix} z_1 u_1^T \\ \vdots \\ z_N u_N^T \end{bmatrix} \begin{bmatrix} z_1 u_1 & \cdots & z_N u_N \end{bmatrix} \underline{\lambda}^T + \underline{1} + \mu \underline{\lambda}^T \begin{bmatrix} z_1 \\ \vdots \\ z_N \end{bmatrix} \quad - (1) \quad = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$$

$$\frac{\partial}{\partial \mu} L_D'(\underline{\lambda}, \mu) = \lambda^T \begin{bmatrix} z_1 \\ \vdots \\ z_N \end{bmatrix} = 0 \quad - (2)$$

(1) and (2) can be written as, (after expansion) : $(N = 3)$

$$= \begin{bmatrix} z_1^2 u_1^T u_1 & z_1 z_2 u_1^T u_2 & z_1 z_3 u_1^T u_3 & -z_1 \\ z_1 z_2 u_1^T u_2 & z_2^2 u_2^T u_2 & z_2 z_3 u_2^T u_3 & -z_2 \\ z_1 z_3 u_1^T u_3 & z_2 z_3 u_2^T u_3 & z_3^2 u_3^T u_3 & -z_3 \\ z_1 & z_2 & z_3 & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \mu \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} //$$

$$= \underbrace{\begin{bmatrix} \begin{bmatrix} z_1 u_1^T \\ z_2 u_2^T \\ z_3 u_3^T \end{bmatrix} \begin{bmatrix} z_1 u_1 & z_2 u_2 & z_3 u_3 \end{bmatrix} & -\begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} \\ \begin{bmatrix} z_1 & z_2 & z_3 \end{bmatrix} & 0 \end{bmatrix}}_{4\times 4} \underbrace{\begin{bmatrix} \lambda^T \\ \mu \end{bmatrix}}_{4\times 1} = \underbrace{\begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}}_{4\times 1} //$$

So, $\underline{\underline{A}} = \begin{bmatrix} z_1^2 u_1^T u_1 & z_1 z_2 u_1^T u_2 & z_1 z_3 u_1^T u_3 & -z_1 \\ z_1 z_2 u_1^T u_2 & z_2^2 u_2^T u_2 & z_2 z_3 u_2^T u_3 & -z_2 \\ z_1 z_3 u_1^T u_3 & z_2 z_3 u_2^T u_3 & z_3^2 u_3^T u_3 & -z_3 \\ z_1 & z_2 & z_3 & 0 \end{bmatrix} //$

$\underline{P} = \begin{bmatrix} \lambda^T \\ \mu \end{bmatrix} //$

$b = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} //$

This clearly satisfies $u_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $u_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \in S_1$

$u_3 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \in S_2$.

2. THIS HAS BEEN IMPLEMENTED AS CODE
(g)

(i) For the dataset $u_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$, $u_2 = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \in S_1$,

$u_3 = \begin{bmatrix} 0 \\ 1.5 \end{bmatrix} \in S_2$

we obtain a negative value of λ at ~~index 2~~ index 2.
i.e $\lambda_2 = -ve$. Hence, set $\lambda_2 = 0$ and we begin reoptimising $L_D'(\underline{\lambda}, \mu)$ (defined in 2(a)) by setting $\lambda_2 = 0$. The final expression of $L_D'(\underline{\lambda}, \mu)$

$$\therefore L_D'(\underline{\lambda}, \mu) = \lambda_1 + \lambda_3 + \mu(\lambda_1 z_1 + \lambda_3 z_3) - \frac{1}{2}\left\{ \lambda_1^2 z_1^2 u_1^T u_1 + \lambda_1 \lambda_3 z_1 z_3 u_1^T u_3 + \lambda_3 \lambda_1 z_3 z_1 u_3^T u_1 + \lambda_3^2 z_3^2 u_3^T u_3 \right\}$$

$$\therefore \frac{\partial L_D'}{\partial \lambda_1} = 1 + \mu(z_1) - \frac{1}{2}\left\{ 2\lambda_1 z_1^2 u_1^T u_1 + \lambda_3 z_1 z_3 u_1^T u_3 + \lambda_3 z_3 z_1 u_3^T u_1 \right\} \quad (1)$$

$$\frac{\partial L_D'}{\partial \lambda_3} = 1 + \mu(z_3) - \frac{1}{2}\left\{ \lambda_1 z_1 z_3 u_1^T u_3 + \lambda_1 z_3 z_1 u_3^T u_1 + 2\lambda_3 z_3^2 u_3^T u_3 \right\} \quad (2)$$

$$\frac{\partial L_D'}{\partial \mu} = \lambda_1 z_1 + \lambda_3 z_3 = 0 . \quad (3)$$

(1) can be written as, $\frac{\partial L_D'}{\partial \lambda_1} = 1 + \mu(z_1) - \frac{1}{\cancel{2}}\left\{ \cancel{2}\lambda_1 z_1^2 u_1^T u_1 + \cancel{2}\lambda_3 z_1 z_3 u_1^T u_3 \right\}$

Same applies for (2), $\frac{\partial L_D'}{\partial \lambda_3} = 1 + \mu(z_3) - \frac{1}{\cancel{2}}\left\{ \cancel{2}\lambda_3 z_3^2 u_3^T u_3 + \cancel{2}\lambda_1 z_1 z_3 u_1^T u_3 \right\}$

Recombining the equations, we get

$$\begin{bmatrix} z_1^2 u_1^T u_1 & z_1 z_3 u_1^T u_3 & -z_1 \\ z_1 z_3 u_1^T u_3 & z_3^2 u_3^T u_3 & -z_3 \\ z_1 & z_3 & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_3 \\ \mu' \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} \begin{bmatrix} z_1 u_1^T \\ z_3 u_3^T \end{bmatrix} \begin{bmatrix} z_1 u_1 & z_3 u_3 \end{bmatrix} & \begin{bmatrix} z_1 \\ z_3 \end{bmatrix} \\ \begin{bmatrix} z_1 & z_3 \end{bmatrix} & 0 \end{bmatrix} \begin{bmatrix} \lambda'^T \\ \mu' \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$$

And so, $A = \begin{bmatrix} z_1^2 u_1^T u_1 & z_1 z_3 u_1^T u_3 & -z_1 \\ z_1 z_3 u_1^T u_3 & z_3^2 u_3^T u_3 & -z_3 \\ z_1 & z_3 & 0 \end{bmatrix}$, $p = \begin{bmatrix} \lambda_1 \\ \lambda_3 \\ \mu' \end{bmatrix}$, $b = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$ //

PROBLEM 2

(b) The code submitted performs the following:

(i). Inverts the A matrix (obtained in 2(a)), to obtain values of λ and μ.

(ii). Checks the resulting KKT conditions on the obtained λ and μ values.

(iii). Calculates the optimal weight vector w^* and bias w_0 and checks if they satisfy the KKT conditions.

The entire problem is done with a non-augmented notation.

(c) For the dataset,

$u_1 = [1, 2]$, $u_2 = [2, 1] \in S_1$ and $u_3 = [0, 0] \in S_2$.

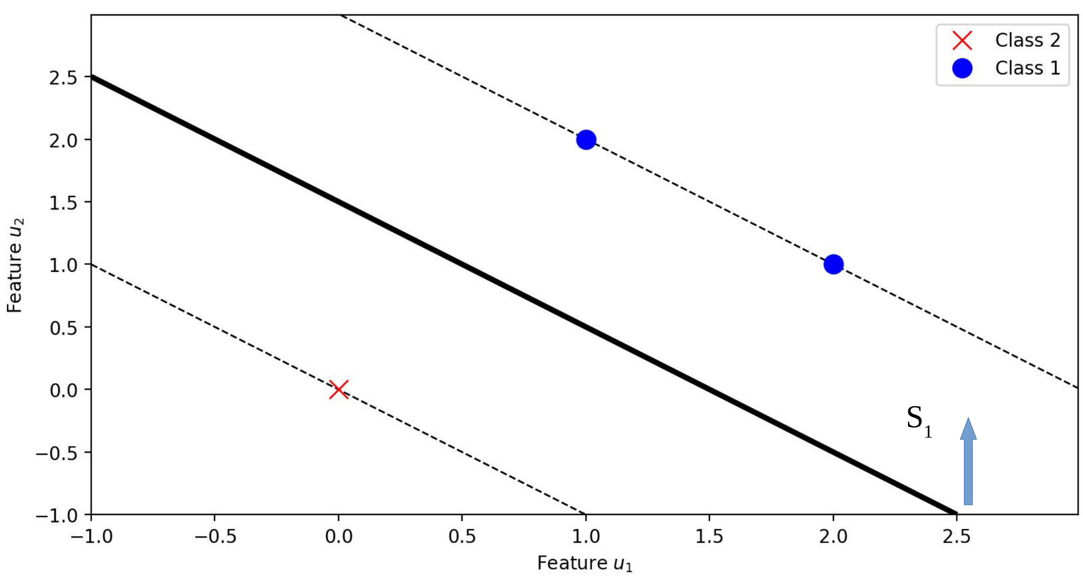
λ = [0.222, 0.222, 0.444] and μ = 1.0

Since, $\sum_{i=1}^{N} \lambda_i z_i$ = 0, is one of the KKT conditions along with λ_i (for all i), to be greater than 0, this is satisfied for all the obtained values of λ (0.222 * 1 + 0.222 * 1 + 0.444 * -1 = 0).

The optimal weight vector w* = [0.666, 0.666] and bias w_0 = 1.0.
The obtained weight vector and the bias satisfy the KKT conditions mentioned in 1(c) (KKT condition 3)

(d) The hyperplane equation is given by w* *u + w_0 = 0. The equation of the support vectors is w* *u + w_0 = b, -b. Here, b = 1, as we make the objective of minimizing the L2 norm of w. **The objective is to have the maximum distance between the hyperplane and the support vectors**. The 2D nonaugmented feature (u) space along with the SVM decision boundaries and the support vectors is illustrated below.

SVM decision boundary for dataset 1.



(e) **Yes**, on clear observation, the decision boundary (**plotted in bold**) does indeed classify the points correctly.

Yes, the hyperplane **is a maximum-margin boundary** because it for sure is maximising the distance to the closest data points from both the classes. The support vectors (**plotted in dashed lines**) also lie on the respective data points and are supportive in defining the maximum margin of the hyperplane shown above. **Since, we obtained all positive λ values, all the points lie on the boundary of the constraint region (defined by the support vectors).**

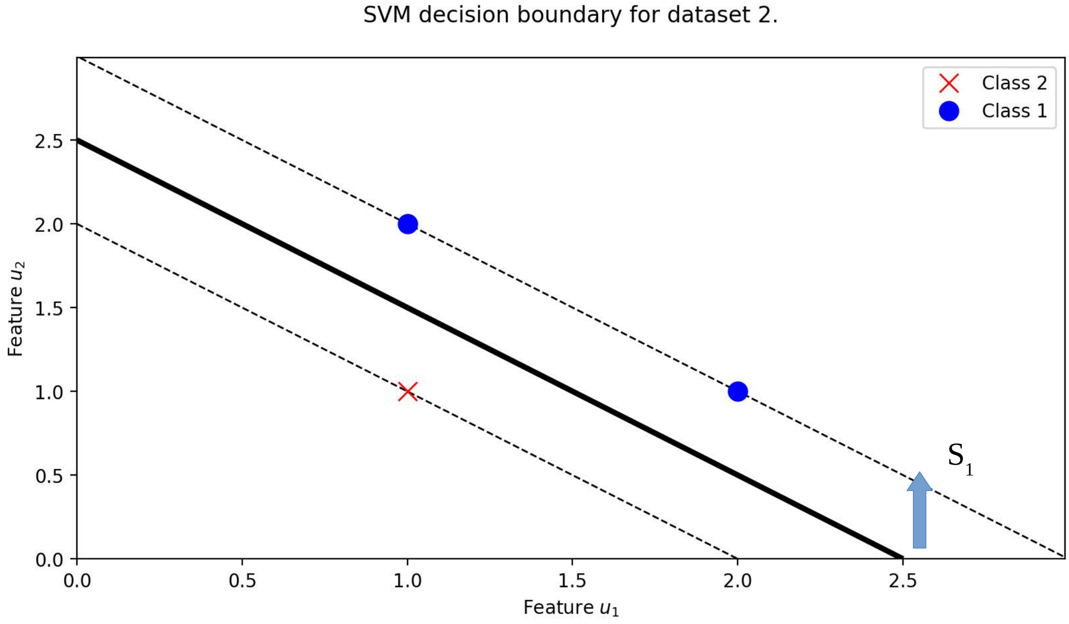
(f)
(i)-(iii) For the dataset,

$$u_1 = [1, 2],\ u_2 = [2, 1] \in S_1 \text{ and } u_3 = [1, 1] \in S_2.$$

λ = [2, 2, 4] and μ = 5.0

Since, $\sum_{i=1}^{N} \lambda_i z_i = 0$, is one of the KKT conditions along with λ_i (for all i) to be greater than 0, this is satisfied for all the obtained values of λ (2 * 1 + 2 * 1 + 4 * -1 = 0).
The obtained weight vector and the bias satisfy the KKT conditions mentioned in 1(c) (KKT condition 3).

(iv) The 2D non-augmented feature (u) space along with the SVM decision boundaries and the support vectors is illustrated below.



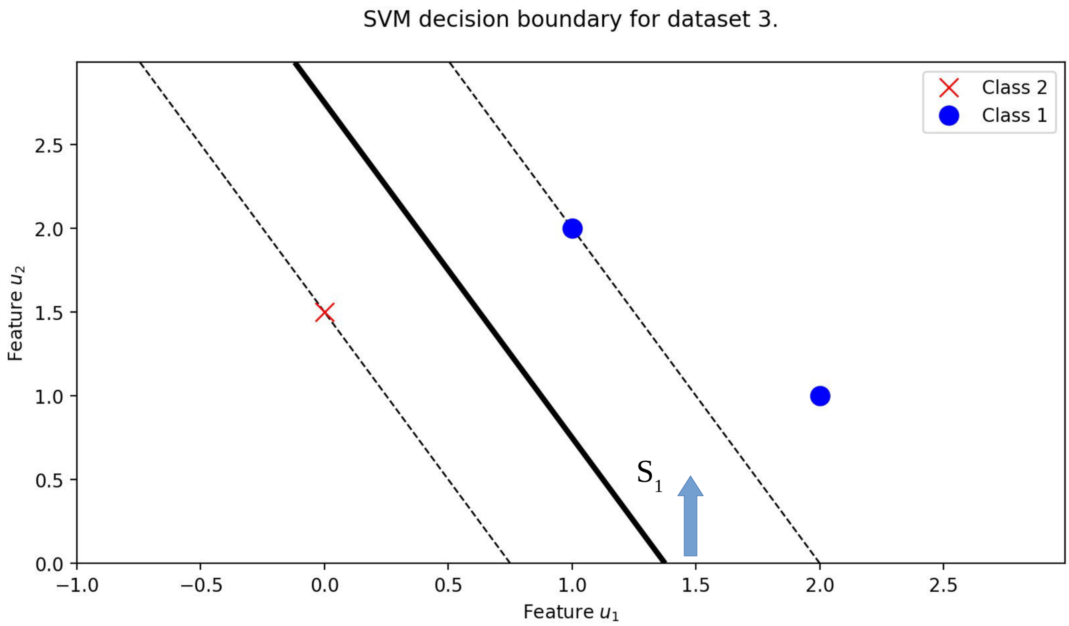
Yes, it's a maximum-margin hyperplane. Even though the optimal weight vector and the bias term are totally different from what we obtained in (d), the decision boundary from the SVM looks and **remains the same i.e there's no change. All the data points are on the boundaries of the constraint regions (defined by the support vectors).** However, the data point of class 2 (z = -1) has moved closer to the decision boundary.However, the distance between the two support vectors is maximum for this dataset, making sure that the distance between the closest data points is maximum.

(g) For the dataset,

$$u_1 = [1, 2],\ u_2 = [2, 1] \in S_1 \text{ and } u_3 = [0, 1.5] \in S_2.$$

(ii)-(iv) **λ = [1.6, 0, 1.6] and μ = 2.2.**
Initially, a negative lambda (λ2) was obtained for u2. However, it was set to 0 and the entire process was re-optimised by taking derivatives of the Lagrangian for λ1 , λ3, and μ. This resulted in a 3 x 3 matrix A, that when inverted resulted in the λ and μ values reported above. **The KKT conditions are satisifed.**
Optimal weight vector w* = [1.6, 0.8] and the bias w_0 = -2.199. These satisfy the KKT conditions defined in 1(c).

(v) The 2D non-augmented feature (u) space along with the SVM decision boundaries and the support vectors is illustrated below.



The above plot is pretty intuitive. **It's a maximum-margin classifier since none of the data points lie within the constrained region (that's defined by the support vectors). All points lie either on the support vectors or away from it. The area between the support vectors is the constrained region.**

The objective of finding all the data points on the boundary of the constraint region fails since one of the data points (u2) causes the Lagrange multiplier (λ2) to be negative initially, and so u2 is not on the support vector. This makes it different from (d) and (f).