321123788 IML - תרגיל 3

אשכון רומן

1) נניח בה"כ ש- $\arg\max \mathbb{P}(x|y)\mathbb{P}(y)=1$, ההוכחה עבור -1 זהה. אם כן:

$$\mathbb{P}(x|y=1)\mathbb{P}(y=1) \geqslant \mathbb{P}(x|y=-1)\mathbb{P}(y=-1)$$

$\Updownarrow$ חוק בייס

$$\mathbb{P}(y=1|x)\mathbb{P}(x) \geqslant \mathbb{P}(y=-1|x)\mathbb{P}(y)$$

$\Updownarrow$ נצמצם ב- $\mathbb{P}(x)$

$$\mathbb{P}(y=1|x) \geqslant \mathbb{P}(y=-1|x)$$

$+\mathbb{P}(y=1|x)$ $\Updownarrow$

$$2\cdot\mathbb{P}(y=1|x) \geqslant \mathbb{P}(y=-1|x)+\mathbb{P}(y=1|x)=1$$

$\Updownarrow$

$$\mathbb{P}(y=1|x) \geqslant \frac{1}{2}$$

כלומר:

$$\arg\max \mathbb{P}(x|y)\mathbb{P}(y)=1 \iff \mathbb{P}(y=1|x)\geq\frac{1}{2} \iff h_D(x)=1$$

לכן נסיק ש:

$$h_D(x)=\underset{y\in\{\pm1\}}{\arg\max}\,\mathbb{P}(x|y)\mathbb{P}(y)$$

2) נציב לנוסחה מהסעיף הקודם את f ונקבל:

$$h_D(x)=\underset{y\in\{\pm1\}}{\arg\max}\, f(x|y)\cdot\mathbb{P}(y)$$

$$=\underset{y\in\{\pm1\}}{\arg\max}\left(\frac{1}{\sqrt{(2\pi)^d\det(\Sigma)}}\exp\left(-\frac{1}{2}(x-\mu_y)^T\Sigma^{-1}(x-\mu_y)\right)\right)\mathbb{P}(y)$$

כעת נשים לב ש- Σ זהה עבור $y=\pm1$, ולכן גם הביטוי $\left(\sqrt{(2\pi)^d\det(\Sigma)}\right)^{-1}$ זהה ולכן ניתן להתעלם ממנו:

$$= \underset{y\in\{\pm1\}}{\operatorname{argmax}} \left(\exp\left(-\frac{1}{2}(x-\mu_y)^T\Sigma^{-1}(x-\mu_y)\right)\right)\mathbb{P}(y)$$

כעת נשים לב שהפונ' ln() מונוטונית, לכן:

$$= \underset{y\in\{\pm1\}}{\operatorname{argmax}} \left(\ln\left(\exp\left(-\frac{1}{2}(x-\mu_y)^T\Sigma^{-1}(x-\mu_y)\right)\right)\mathbb{P}(y)\right)$$

$$= \underset{y\in\{\pm1\}}{\operatorname{argmax}} \left(\left(-\frac{1}{2}(x-\mu_y)^T\Sigma^{-1}(x-\mu_y)\right) + \ln \mathbb{P}(y)\right) =$$

$$= \underset{y\in\{\pm1\}}{\operatorname{argmax}} \left(-\frac{1}{2}x^T\Sigma^{-1}x + \frac{1}{2}x^T\Sigma^{-1}\mu_y + \frac{1}{2}\mu_y^T\Sigma^{-1}x - \frac{1}{2}\mu_y^T\Sigma^{-1}\mu_y + \ln \mathbb{P}(y)\right) =$$

כעת נשים לב שהביטוי $-\frac{1}{2}x^T\Sigma^{-1}x$ לא תלוי ב-y

לכן נתעלם גם ממנו, כמו כן מכך ש-Σ^{-1} סימטרית

מתקיים:

$$\mu_y^T\Sigma^{-1}x = \left(x^T\Sigma^{-1^T}\mu_y\right)^T = x^T\Sigma^{-1}\mu_y$$

לכן:

$$= \underset{y\in\{\pm1\}}{\operatorname{argmax}} \left(x^T\Sigma^{-1}\mu_y - \frac{1}{2}\mu_y^T\Sigma^{-1}\mu_y + \ln \mathbb{P}(y)\right) = \underset{y\in\{\pm1\}}{\operatorname{argmax}}\, \delta_y(x)$$

3) נגדיר:

$$S_1 = \{(x,y)\in S \mid y=1\},\quad S_2 = \{(x,y)\in S \mid y=-1\}$$

כעת החישוב של μ_y יהיה:

$$\mu_y = \frac{1}{|S_y|}\sum_{(x,y)\in S_y} x$$

והחישוב של Σ:

$$\Sigma = \frac{1}{|S|-1}(x-\mu_x)(x-\mu_x)^T$$

עבור μ_x ממוצע כל הנקודות
בהתפלגות של x

והחישוב של $\mathbb{P}(y)$:

$$\mathbb{P}(Y=1) = \frac{|S_1|}{|S|},\quad \mathbb{P}(Y=-1) = \frac{|S_2|}{|S|}$$

4) נגדיר positive להיות החלטה לסווג מייל כספאם,
ו- negative כלא ספאם.
הטעויות האפשריות הן:

* false-positive - הגדרנו מייל כספאם על אף שאינו באמת ספאם. זוהי הטעות החמורה יותר מכיוון שהיא עלולים להתעלמות ממיילים חשובים (פוטנציאלית).
* false-negative - הגדרנו מייל כלא ספאם על אף שהוא אכן ספאם. זוהי טעות חמורה פחות מכיוון שבמקרה זה בסך הכל "יחדרו" יותר מיילים של ספאם, אבל מיילים שאינם ספאם לא "יחסמו" לכן לא נפספס מיילים חשובים.

5) נסמן:

$$Q = 2I_n,\quad a = 0_{n\times 1},\quad A = \begin{bmatrix} - & -y_1x_1 & - \\ & \vdots & \\ - & -y_mx_m & - \end{bmatrix}$$

$$d = \begin{bmatrix} y_1b-1 \\ \vdots \\ y_mb-1 \end{bmatrix}$$

ואכן נקבל:

$$\underset{v\in\mathbb{R}^n}{\operatorname{argmin}}\left(\frac{1}{2}v^TQv + a^Tv\right) = \underset{v\in\mathbb{R}^n}{\operatorname{argmin}}\left(\frac{1}{2}v^T2I_nv + 0_{n\times 1}v\right) =$$

$$= \underset{v\in\mathbb{R}^n}{\operatorname{argmin}}\left(2\cdot\frac{1}{2}\cdot v^T\cdot v + 0\right) = \underset{v\in\mathbb{R}^n}{\operatorname{argmin}}\|v\|^2$$

וכמו כן:

$$\forall i,\ y_i(\langle w,x_i\rangle + b) \geq 1 \iff \begin{bmatrix} y_1\langle w,x_1\rangle \\ \vdots \\ y_m\langle w,x_m\rangle \end{bmatrix} \geq \begin{bmatrix} 1-y_1b \\ \vdots \\ 1-y_mb \end{bmatrix} \iff$$

$$\iff \begin{bmatrix} \langle w,-y_1x_1\rangle \\ \vdots \\ \langle w,-y_mx_m\rangle \end{bmatrix} \leq \begin{bmatrix} -1+y_1b \\ \vdots \\ -1+y_mb \end{bmatrix} \iff \begin{bmatrix} - & -y_1x_1 & - \\ & \vdots & \\ - & -y_mx_m & - \end{bmatrix}\cdot w \leq \begin{bmatrix} -1+y_1b \\ \vdots \\ -1+y_mb \end{bmatrix} \iff$$

$$\iff Aw \leq d$$

6) הביטוי $\frac{\lambda}{2}\|w\|^2$ נמצא בשתי הבעיות, לכן ההבדל
נובע מהסכומים $\frac{1}{m}\sum_{i=1}^{m}\xi_i$ ו- $\frac{1}{m}\sum_{i=1}^{m}\ell^{hinge}(y_i\langle w,x_i\rangle)$.

אם כן נראה שהם שקולים בתנאי הבעיות:

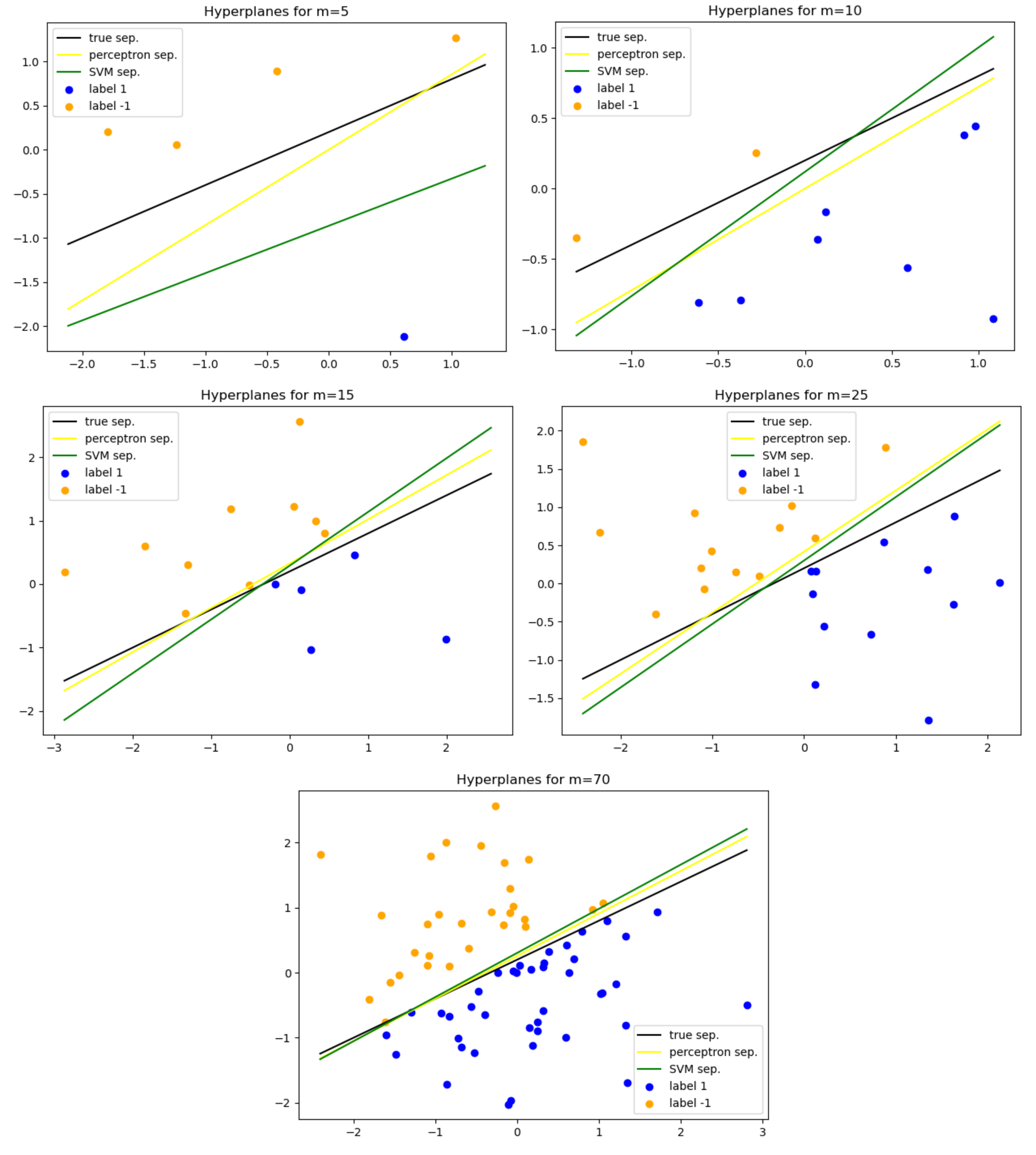
* $\xi_i \geq 0$ עבור כל $i\in[n]$ היא אחת הדרישות. נשים לב שמהגדרת ℓ^{hinge} מתקיים:

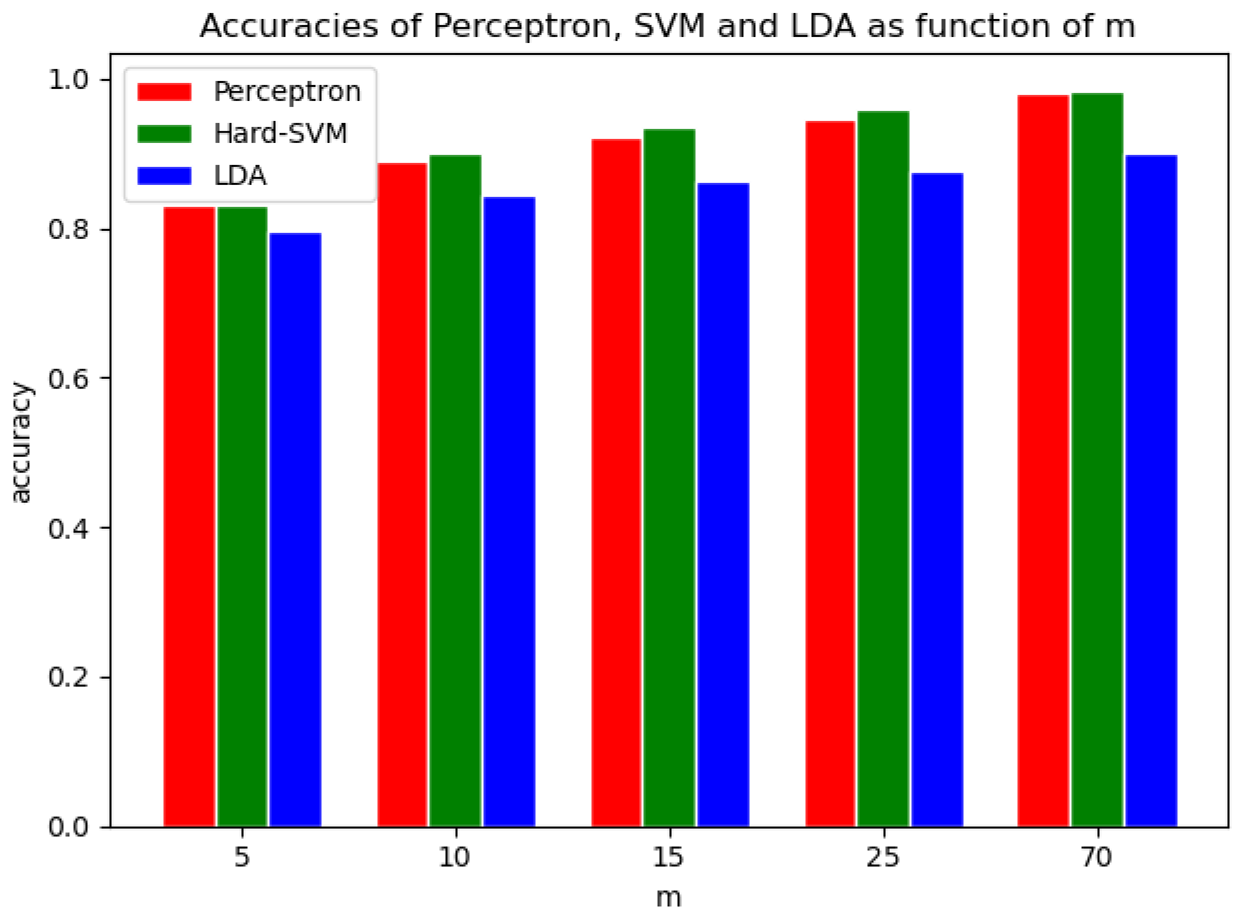
$$\ell^{hinge}(y_i\langle w,x_i\rangle)=\max\{0, 1-y_i\langle w,x_i\rangle\} \geq 0$$

* $y_i\langle w,x_i\rangle \geq 1-\xi_i \iff \xi_i \geq 1-y_i\langle w,x_i\rangle$. גם פה מהגדרת ℓ^{hinge} מתקיים:

$$\ell^{hinge}(y_i\langle w,x_i\rangle)=\max\{0, 1-y_i\langle w,x_i\rangle\} \geq 1-y_i\langle w,x_i\rangle$$

אם כך הביטוי $\frac{1}{m}\sum_{i=1}^{m}\ell^{hinge}(y_i\langle w,x_i\rangle)$ מקיים את התנאים של הבעיה המקורית ולכן כל פיתרון לבעיה יפתור גם את שתיהן כנדרש.

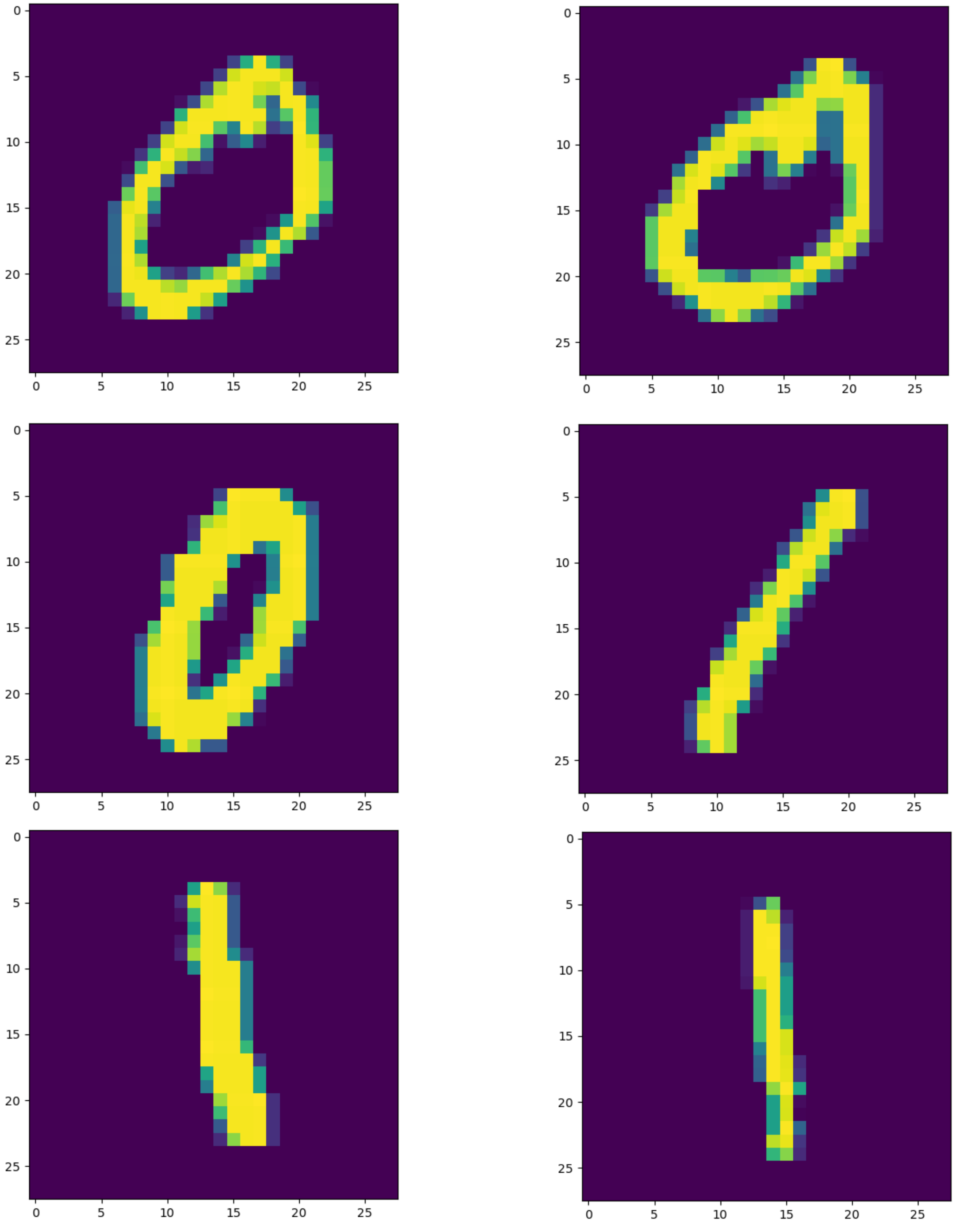

Hyperplanes for m=5
Hyperplanes for m=10
Hyperplanes for m=15
Hyperplanes for m=25
Hyperplanes for m=70
true sep.
perceptron sep.
SVM sep.
label 1
label -1


Accuracies of Perceptron, SVM and LDA as function of m
Perceptron
Hard-SVM
LDA
accuracy
0.0
0.2
0.4
0.6
0.8
1.0
5
10
15
25
70
m

11) המודלים הטובים ביותר בסדר יורד הם:

Hard-SVM , Perceptron , LDA

לדעתי, ה- Hard-SVM המודל הטוב ביותר מכיוון ש-SVM מוצא את ה- hyperplane שמפריד בצורה האופטימלית את הנקודות הקרובות ביותר לקו המפריד, בזמן ש-LDA מחשב את קו המפריד בעזרת התוחלת של כל הנקודות, ולכן יש לו יותר מקום לסטיית תקן.



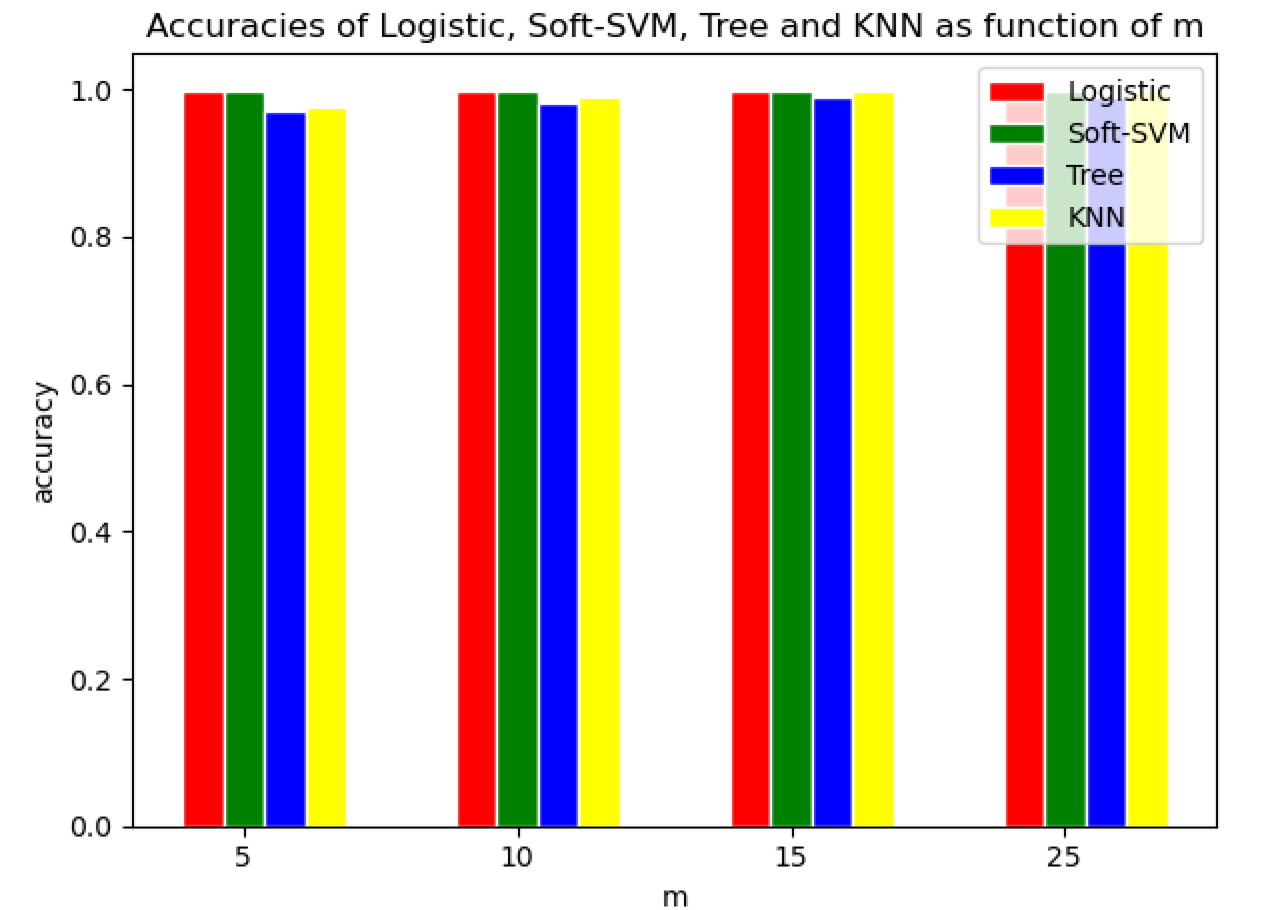
0
5
10
15
20
25
0 5 10 15 20 25

14) ניתן לראות כי זמני הריצה הם:

Logistic < Tree < Soft-SVM < K-Nearest Neighbors

לדעתי הסיבה לכך היא ש- KNN צריך לבצע הכי הרבה חישובים. גודל קלט בודד הוא 784xm, ועל מנת להגיע לתשובה KNN צריך לחשב את המרחק מכל נקודה בערכי האימון וזה לוקח הרבה זמן. ~~[illegible]~~

מנגד שאר האלגוריתמים לא מבצעים השוואה בין נקודות שונות ולכן זמן הריצה שלהם נמוך יותר.



Accuracies of Logistic, Soft-SVM, Tree and KNN as function of m
accuracy
1.0
0.8
0.6
0.4
0.2
0.0
5
10
15
25
m
Logistic
Soft-SVM
Tree
KNN

```
Logistic :
m =  50 :  0.024007797241210938
m =  100 :  0.016999244689941406
m =  300 :  0.022999048233032227
m =  500 :  0.031995534896850586

Soft-SVM :
m =  50 :  0.05699133872985 84
m =  100 :  0.06499934196472168
m =  300 :  0.11614871025085449
m =  500 :  0.12100052833557129

Tree :
m =  50 :  0.013003587722778 32
m =  100 :  0.012002229690551758
m =  300 :  0.016860008239746094
m =  500 :  0.031000137329101562

KNN :
m =  50 :  0.31099462509155273
m =  100 :  0.4159977436065674
m =  300 :  1.3459882736206055
m =  500 :  2.4542009830474854
```