

### 3 Linear - IML

373574006 שם: דניאל דניאל

$$\forall x \in X \quad h_g(x) = \begin{cases} +1 & P(y=1|x) \geq \frac{1}{2} \\ -1 & \text{else} \end{cases} \quad \text{דיון}$$

יש להיזהר כי  $x \in X$  חתך בין שני מרחבי

$$P(y=1|x) = \frac{P(x|y=1) \cdot P(y=1)}{P(x)}$$

$$P(y=-1|x) = \frac{P(x|y=-1) \cdot P(y=-1)}{P(x)}$$

לכן  $h_g$  כי המכונה  $h$  מרחבי

$$h_g(x) = 1 \Leftrightarrow P(x|y=1) \cdot P(y=1) > P(x|y=-1) \cdot P(y=-1)$$

$$h_g(x) = -1 \Leftrightarrow P(x|y=-1) \cdot P(y=-1) > P(x|y=1) \cdot P(y=1) \quad \text{דיון}$$

$\Rightarrow$  נגד לנסות למצוא  $h(x)$  גלוי מילה  $h$

מורה לקבל ישר, כאשר  $y=1$  או  $y=-1$

$$\Rightarrow h_g = \underset{y \in \{-1, 1\}}{\operatorname{argmax}} P(x|y) \cdot P(y) \quad \square$$

2. למשל (הקדמה)  $h$  לוג

$$\ln(h_g) = \underset{y \in \{-1, 1\}}{\operatorname{argmax}} (\ln(P(x|y) \cdot P(y))) = \underset{y \in \{-1, 1\}}{\operatorname{argmax}} (\ln(P(x|y)) + \ln(P(y)))$$

3. שר הוקצו הבעיה, ולכן מהגורמים שונים גלויים  $y$

$$\Rightarrow = \ln(\exp\{-\frac{1}{2}(x-\mu_y)^T \Sigma^{-1}(x-\mu_y)\}) + \ln(P(y)) =$$

$$= (-\frac{1}{2}x^T \Sigma^{-1}x + \frac{1}{2}\mu_y^T \Sigma^{-1}x - \frac{1}{2}\mu_y^T \Sigma^{-1}\mu_y) + \ln(P(y)) =$$

$$= -\frac{1}{2}x^T \Sigma^{-1}x + \frac{1}{2}x^T \Sigma^{-1}\mu_y + \frac{1}{2}\mu_y^T \Sigma^{-1}x - \frac{1}{2}\mu_y^T \Sigma^{-1}\mu_y + \ln(P(y))$$

שם, נניח למדוד שונים גלויים  $y$

דיון



$$\underset{y \in \{+1\}}{\operatorname{argmax}} \quad = \frac{1}{2} x^T \Sigma^{-1} m_y + \frac{1}{2} m_y^T \Sigma^{-1} x - \frac{1}{2} m_y^T \Sigma^{-1} m_y + \ln(P(y))$$

$$= x^T \Sigma^{-1} m_y - \frac{1}{2} m_y^T \Sigma^{-1} m_y + \ln(P(y)) = \mathcal{J}_y(x)$$

$$\Rightarrow \ln(h_\theta(1)) = \underset{y \in \{+1\}}{\operatorname{argmax}} \mathcal{J}_y(x) \quad \Leftrightarrow \quad h_\theta(x) = \underset{y \in \{+1\}}{\operatorname{argmax}} \mathcal{J}_y(x)$$





$$\forall y \in \{-1, 1\} \quad \mu_y = \frac{1}{M_y} \sum_{y_i = y} x_i$$

3

mean of  $x$  for  $y=1$  is  $\mu_1$  and for  $y=-1$  is  $\mu_{-1}$ .  
 $y \rightarrow$

$$\forall y \in \{-1, 1\}, \quad P(Y=y) = \frac{M_y}{M}$$

$$\Sigma = \frac{1}{M-2} \sum_{y \in \{-1, 1\}} \sum_{y_i = y} (x_i - \mu_y)(x_i - \mu_y)^T$$

not spam = negative

spam  $\Rightarrow$  positive

4

false positive  $\Leftarrow$  spam  $\rightarrow$  not spam  
 Type-I error

false negative  $\Leftarrow$  not spam  $\rightarrow$  spam  
 Type-II error



$$Q = 2I_d, \quad a = \vec{0}, \quad v = \vec{w}$$

$$d = \begin{bmatrix} \gamma_1 b - 1 \\ \vdots \\ \gamma_m b - 1 \end{bmatrix} \quad A = \begin{bmatrix} -\gamma_1 x_1^T \\ \vdots \\ -\gamma_m x_m^T \end{bmatrix}$$

$$\gamma_i (\langle w | x_i \rangle + b) \geq 1 \Leftrightarrow \gamma_i w^T x_i + \gamma_i b \geq 1$$

$$\Leftrightarrow -\gamma_i w^T x_i \leq \gamma_i b - 1 \Leftrightarrow A v \leq d$$

$$\frac{1}{2} w^T Q w + a^T w = \frac{1}{2} w^T 2I_d w + \vec{0}^T w$$

$$= w^T w = \|w\|^2$$

hyperplane

$(w, b)$

$\xi_1, \dots, \xi_m$

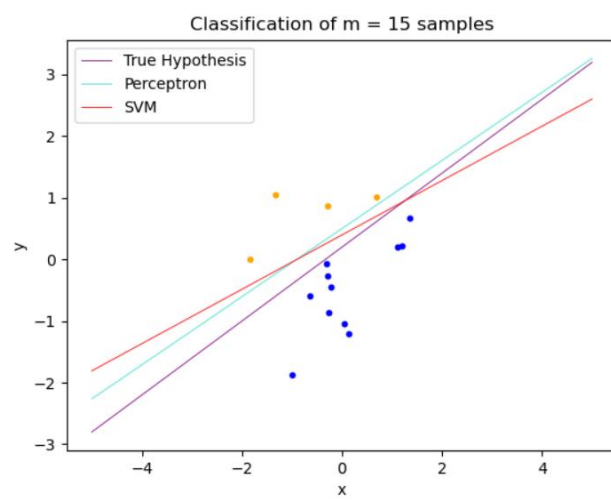
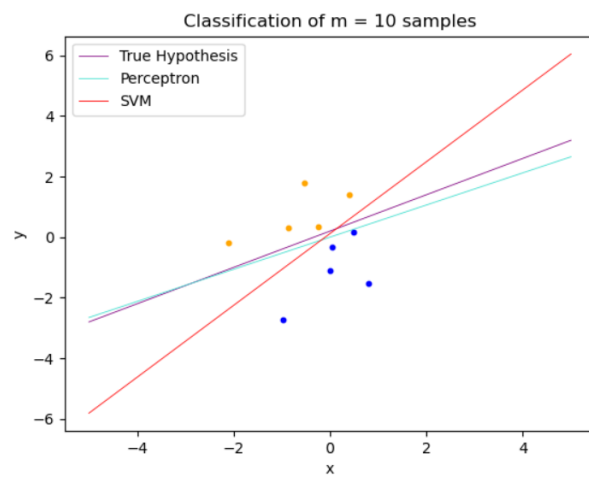
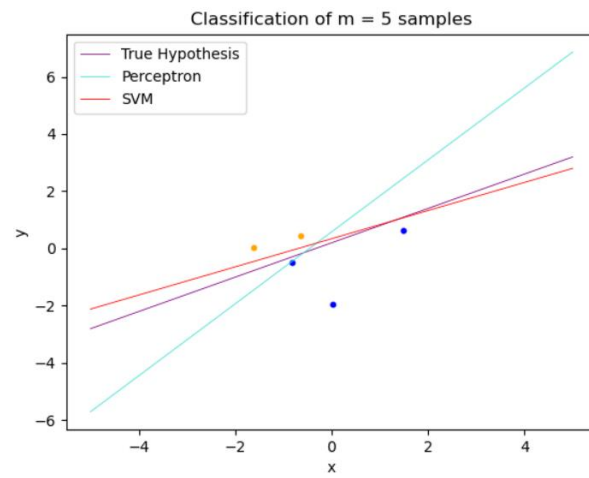
$\ell$

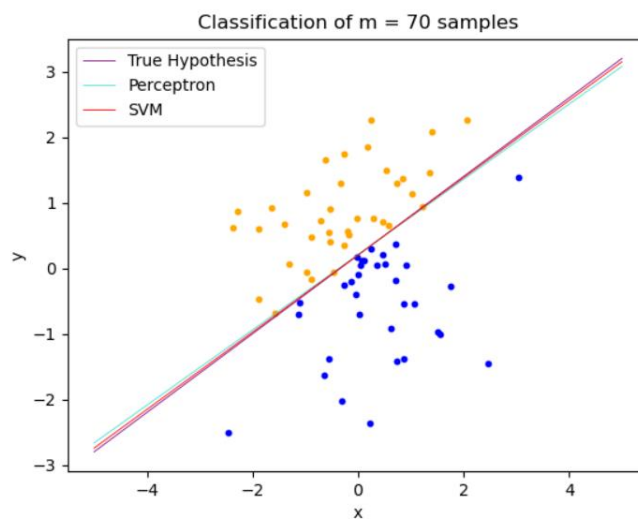
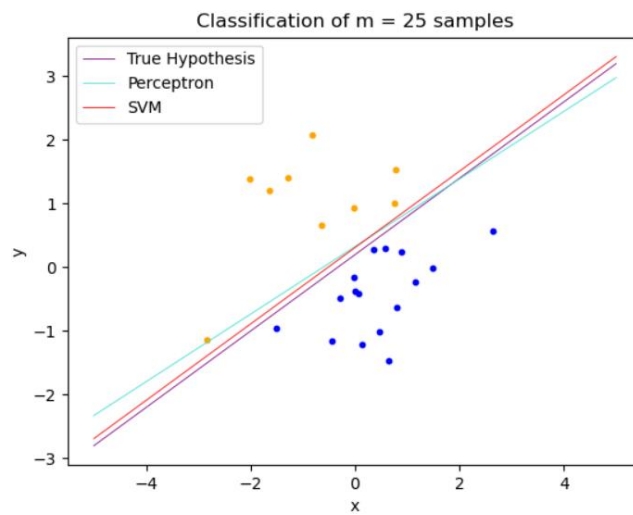
$\xi_i$  is the slack variable

$$\xi_i = \begin{cases} 0 & \gamma_i (\langle x_i, w \rangle + b) \geq 1 \\ 1 - \gamma_i (\langle x_i, w \rangle + b) & \text{else} \end{cases}$$

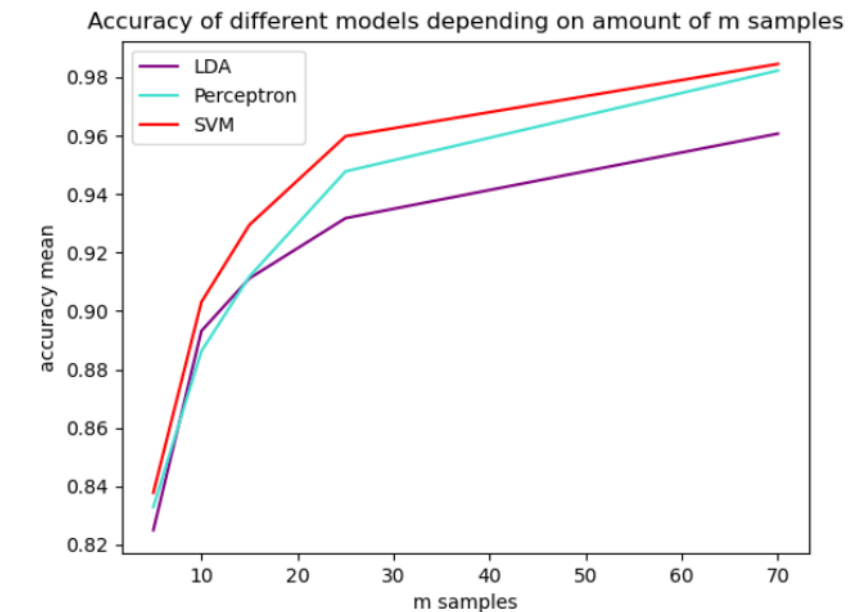
$$\xi_i = \ell^{\text{hinge}}(\gamma_i (\langle x_i, w \rangle + b))$$

$\ell^{\text{hinge}}(a) = \max\{0, 1-a\}$





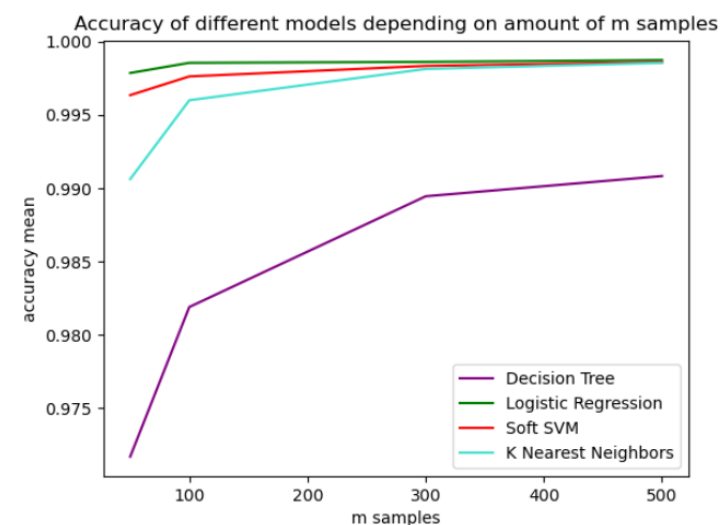
10.



11.

ניתן לראות שאלגוריתם SVM הוא המוצלח ביותר מבין השלושה. הסיבה לכך היא שSVM ממקסם את הmargin, ולעומת זאת perceptron לוקח את העל מישור הראשון שהוא מוצא. לעומת זאת, LDA עובד יותר טוב כאשר הדאטה שאנו מקבלים הם משני התפלגויות גאוסיות שונות, אך בעלות אותן שונות, ולכן ספציפית עבור הדאטה הנתון, LDA עובד פחות טוב משני האלגוריתמים האחרים.

14.



50 samples run time:

Logistic: 0.44689178466796875

Decision Tree: 0.4140474796295166

Soft SVM: 0.7531154155731201

K Nearest Neighbors: 3.6736912727355957

100 samples run time:

Logistic: 0.48216843605041504

Decision Tree: 0.46292781829833984

Soft SVM: 0.8555600643157959

K Nearest Neighbors: 3.807950973510742

300 samples run time:

Logistic: 0.7194459438323975

Decision Tree: 0.7294919490814209

Soft SVM: 1.1340610980987549

K Nearest Neighbors: 4.587292671203613

500 samples run time:

Logistic: 0.9694986343383789

Decision Tree: 1.169306993484497

Soft SVM: 1.4753365516662598

K Nearest Neighbors: 5.350306034088135

ניתן לראות הבדלי זמן ריצה משמעותיים בעיקר מ K Nearest Neighbors וזו כנראה מאופן פעולת האלגוריתם, שעושה קלאסיפיקציה לפי בדיקה של השכנים שלו.