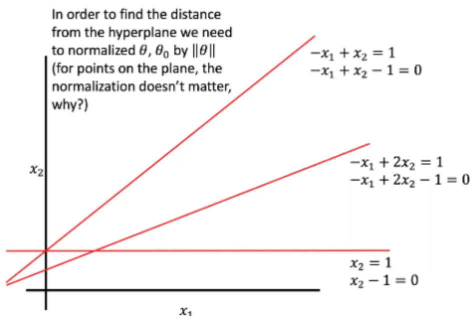
למידה חישובית ממידע: תרגול 5 – Linear Classifiers

מפרידים לינאריים בקלסיפיקציה –

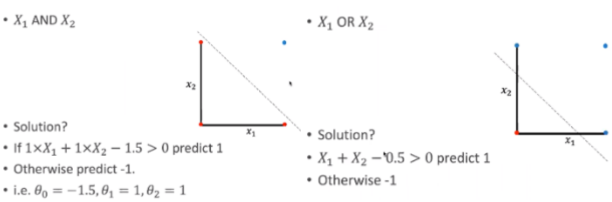
- <u>היפר-מישור במרחב</u>: במרחב חד ממדי היפר מישור הוא נקודה, במרחב דו ממדי היפר מישור הוא קו ובמרחב תלת ממדי היפר מישור הוא משטח דו-ממדי. נשים לב שכביכול מפריד לינארי הוא ממרחב בממד אחד פחות (בייצוג הגיאומטרי) ממרחב הדאטה בו אנו מתעסקים.
- <u>הגדרת היפר-מישור במרחב</u>: מרחב ההיפר-מישור הוא n-1 (אם n הוא המרחב בו אנו עובדים). כל הנקודות על ההיפר-מישור פותרות את המשוואה הבאה: $\theta_1 x_1 + \cdots + \theta_n x_n = b \quad (= \theta_0)$ כאשר x הוא וקטור הקואורדינטות של הדגימה. היפר-מישור מפריד את המרחב לשני מרחבים, כאשר כל נקודה שמניבה תוצאה במשוואה שגדולה מ-b (ה-bias, טטה0) נמצאת מעל המישור, וכל נקודה שמניבה תוצאה במשווה שקטנה מ-b, נמצאת מתחת למישור.



- ה-bias, b, טטה0 – מייצג את **התזוזה של המישור מראשית הצירים**, והטטה (הכללית, המקדמים של x) בכלל מייצגת את **הזווית של המישור במרחב**.
- נשים לב שהדוט-פרודקט = המכפלה הפנימית, בין x לטטה, משמעותה **המרחק של הנקודה מהמישור**. על המישור יש נורמל שמאונך למישור, הדוט פרודקט הוא הטלה של האיקס על הנורמל. **מרחק חיובי משמעותו = מעל המישור, ומרחק שלילי משמעותו = מתחת למישור**. נשים לב שכדי לקבל את המרחק האמיתי על הוקטור טטה להיות מנורמל (לחלק אותו בנורמל של עצמו).
- אם הטטה לא מנורמלת = המרחק לא מייצג את המרחק האוקלידי מהמישור וכך יש משמעות רק לסימן של המרחק כפי שצוין לעיל.

אנחנו מחפשים מפריד לינארי כך שכל הנקודות המניבות תוצאה גדולה מ-0 יסווגו בקלאס +1 (או -1). וכל הנקודות עם תוצאה קטנה מ-0 יסווגו לקלאס -1 (או +1). כלומר אנחנו מחפשים טטה $\theta \in R^{n+1}$ (n hyperplane weights & the bias θ_0) , שהיא וקטור משקולות או מקדמים ל-x, שמקיימת את המכפלה הפנימית המתוארת לעיל ומסווגת באופן הבא: 1 if $\sum_{i=1}^{n} \theta_i x_i + \theta_0 > 0$ and -1 otherwise.

להלן כמה דוגמאות:



אלגוריתם הפרספטרון

אלגוריתם שמחפש מפריד לינארי במרחב, לכן מחפשים טטה באופן הבא: נתחיל עם טטה רנדומית (מישור) ובכל שלב נשפר אם יש צורך בשיפור (כלומר יש יש שגיאה = אאוטפוט פחות טרגט). השינוי בטטה הרלוונטית היא learning rate כפול גרדיאנט.

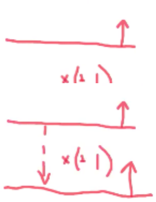
<u>כלל העדכון של הפרספטרון</u>

$$\Delta\theta_i = -\eta \sum_{d \in D} \left(o^{(d)} - t^{(d)}\right) x_i^{(d)}$$

זהו הכלל לעדכון כך שאם Od – Td = 0, אין טעות ולכן אין צורך בעדכון.

0	t	o-t	x_i	$\Delta\theta_i$	$x_i \cdot \theta_i$
-1	+1	<0	>0	>0	increased
-1	+1	<0	<0	<0	increased
+1	-1	>0	>0	<0	decreased
+1	-1	>0	<0	>0	decreased

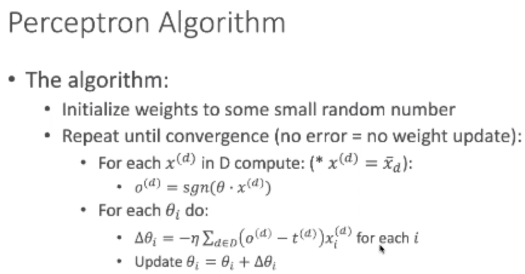
עבור השורה הראשונה: נשים לב שיש לנו נקודה שהיא מהקלאס החיובי t=+1 אבל הפרדיקציה עבורה שלילית o=-1,על כן מבחינה גאומטרית כך יראה המישור(איור). לכן ההפרש בין o ל-t הוא שלילי, ונניח שפיציר ה-xi הוא חיובי, אזי הדוט-פרודקט שלהם שלילי והדלתא טטה חיובית (מפני שאנחנו מכפילים במינוס אטה את הדוט-פרודקט). לכן, המישור "יורד" לכיוון הנקודה, לכיוון השלילי, ונקווה שהעדכון גרם לכך שהנקודה תהיה כעת מעל המישור – המרחק בין xi למישור גדל מפני שהדוט פרודקט בינו לבין טטה גדל.

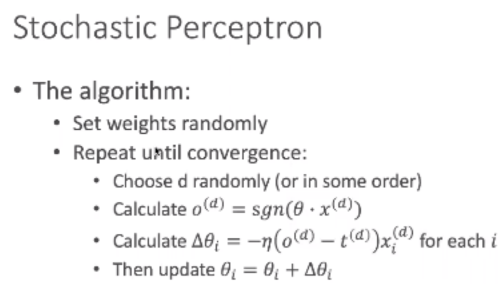


עבור השורה השניה: קורה דבר דומה, שוב יש להוריד את המישור לכיוון נקודה ולכן המרחק, המכפלה הפנימית בין xi לטטה, גדל.

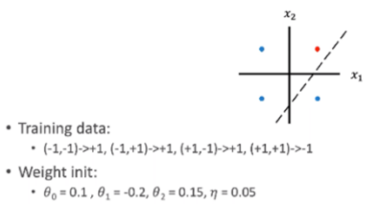
לכן באופן כללי אם המכפלה הפנימית בין xi לטטהi גדלה, המישור "יורד" ביחס לנקודה שאנחנו מדברים עליה. ואם המכפלה קטנה, המישור "עולה" ביחס לנקודה שאנחנו מדברים עליה.

להלן **אלגוריתם הפרספטרון** אשר מתייחס לכל השגיאות (ועל כל הסכום לפני האטה) אבל נוכל להשתמש **בפרספטרון טוכסטי**, שהוא משתמש כל פעם בשגיאה של נקודה אחת.



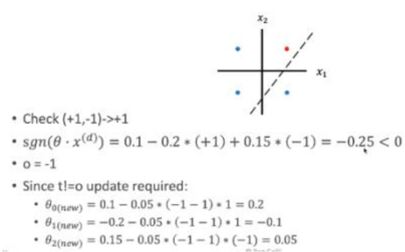


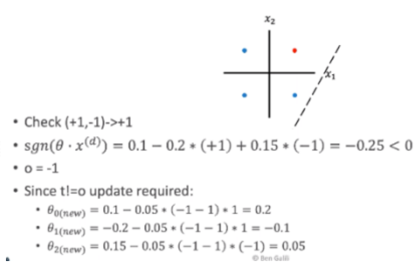
דוגמה עבור פרספטרון סטוכסטי:



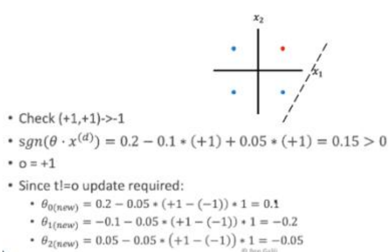
שלב ראשון : אתחול הטטה ולהלן המישור ההתחלתי, כאשר ״מעל״ המישור מוגדר להיות שמאלה כלפי מעלה, לכן יש שתי טעויות האדומה שמעל המישור והכחולה שמתחת למישור.

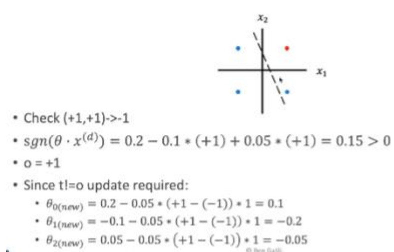
שלב שני: בדיקה עבור כל אחת מהנקודות, עבור 2 נקודות לא יתבצע עדכון ויתבצעו שני עדכונים עבור הנקודה הכחולה שמתחת למישור ועבור הנקודה האדומה שמתחת למישור.





<u>נבצע בדיקה עבור הנקודה הכחולה</u>, ומכיוון שהפרדיקציה לא נכונה יתבצע עדכון – לפני העדכון מופיע משמאל ולאחריו מופיע מימין. נשים לב כי הטטה0 גדל ולכן המרחק מראשי הצירים גדל, שזה מצוין עבורנו כי כך הנקודה הכחולה אכן תתמקם מעל המישור. ונשים לב שטטה1 וטטה2 שומרות על הסימנים שלהן ועל כן ניתן להבין שהזווית לא משתנה מאוד.

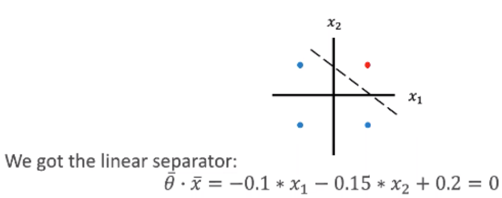




<u>נבצע בדיקה עבור הנקודה האדומה</u>, וגם כאן יתבצע עדכון. לפני העדכון משמאל ולאחריו מימין. נשים לב כי טטה0 קטנה ולכן התקרבנו לראשית, וכן השתנה הסימן של טטה2 ולכן נוצר שינוי בזווית של המישור. כעת הנקודה האדומה אכן מתחת למישור.

אבל נוצרה שגיאה עבור הנקודה הכחולה (+1,-1) שאמורה להיות מעל המישור אך מופיעה מתחת למישור.

לאחר עדכון נוסף עבור טעות זו נקבל את המישור הסופי שלנו.



איך יודעים מה זה מעל ומה זה מתחת למישור? מציבים את ראשית הצירים וכך אם מתקבל ערך חיובי, אנחנו מעל המישור אחרת מתחת. אם נציב את ראשית הצירים כאן נקבל (חיובי)0.2 ולכן לכיוון הכחולים נחשב מעל המישור.

<u>מה הבעיה באלגוריתם הפרספטרון ואיך נוכל לפתור אותה?</u>

לאלגוריתם אין תנאי עצירה למעט התכנסות ל-perfect classification, מה שלא תמיד יתאפשר. נפתור אותה על ידי הגבלת כמות הריצות (הבעיה כאן היא שאנחנו לא נדע אם יכולנו לשפר את מה שהגענו אליו עם ״עוד קצת״ ריצות). או על ידי הגבלת מספר הטעויות אבל גם כאן נוצרת אותה הבעיה. ולבסוף הפתרון האלגנטי ביותר היא להפוך את בעיה זו לבעיית אופטימיזציה – להשתמש בגרדיאנט-דיסנט ולמצוא מינימום טעות. אלגוריתם זה נקרא LMS.

אלגוריתם LMS = Least Mean Squares

אלגוריתם זה פועל בדיוק באותו האופן מלבד העובדה שהוא משתמש בפונקציית טעות שונה = אין שימוש בפונקציית הסימן אלא ממש משתמשים בדוט-פרודקט ומטרתו היא לא למצוא קלסיפיקציה מושלמת – מפריד מושלם, אלא למזער את הטעות, את המרחקים.

LMS – Least Mean Squares

- The algorithm:
  - Initialize weights to some small random number
  - Repeat until convergence (no error = no weight update):
    - For each $x^{(d)}$ in D compute: (* $x^{(d)} = \vec{x}_d$):
      - $o^{(d)} = (\theta \cdot x^{(d)})$
    - For each θ_i do:
      - $\Delta\theta_i = -\eta \sum_{d\in D}(o^{(d)} - t^{(d)})x_i^{(d)}$
      - Update $\theta_i = \theta_i + \Delta\theta_i$

$$E[\vec{\theta}] = \frac{1}{2}\sum_{d\in D}\left(o^{(d)} - t^{(d)}\right)^2 = \frac{1}{2}\left[\sum_{d\in D^+}\left(o^{(d)} - 1\right)^2 + \sum_{d\in D^-}\left(o^{(d)} + 1\right)^2\right]$$

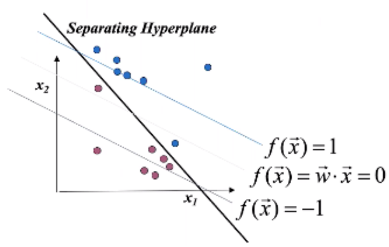
Minimize the distance between the positive instances and the +1 iso-line of the function

Minimize the distance between the negative instances and the -1 iso-line of the function

ההבדלים בין פרספטרון ל-LMS הם:

- ה-target / label הוא נומרי (לא קלאס) – במקרה זה +1,-1.
- חישוב האאוטפוט – התוצאה היא מספר ולא קלאס (הערך האמיתי של theta * xd ולא הסימן של מכפלה פנימית זו)
- פונקציית האופטימיזציה – ב-LMS המרחק המינימלי מ-+1,-1 יוצר את המישור. ובפרספטרון המישור יוצר 0 שגיאות קלסיפיקציה.
- LMS יתכנס, פרספטרון לא בהכרח מתכנס אלא אם הדאטה ניתנת להפרדה לינארית (אין שגיאה על ה-training data).

נשליך רגרסיה לינארית על קלסיפיקציה:
במקום לחזות ערכים רציפים ננסה לחזות קלאס.

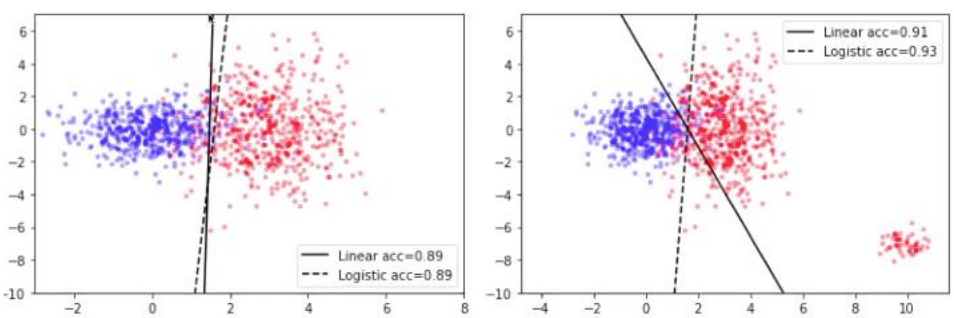


- What if instead of predicting a continues value we try to predict a class?
- Hypothesis function : $h_\theta(x) = \theta^T x$
- Cost function (MSE): $J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$
- Goal : $\min_\theta J(\theta)$

x_1	y
	1
	1
	0
	1
	0
	1

Predict

$if\ h_\theta(x) > 0.5\ then\ 1$
$else\ 0$



נשים לב שלסיכום,

- **פרספטרון** לא תמיד טוב מפני שלא תמיד יתכנס
- **וקלסיפיקציה על פי רגרסיה לינארית** = LMS לא תמיד תניב מפריד אופטימום (כפי שניתן לראות מימין ובדוגמה לעיל).

ולכן נשאלת השאלה איך מוצאים מישור באלגוריתם שמתכנס גם אם הדאטה לא מופרד לינארית? <<

Logistic Regression

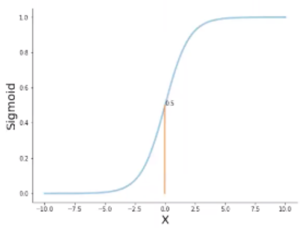
ננסה לחזות את ההסתברויות של דגימה להשתייך לקלאס 1. **פונקציית ההיפתזה היא הפוסטריור** : $h_\theta(x) = P(1|x)$
איך נעבור מרחק ממישור להסתברות? ככל שנקודה נמאצת בצד הנכון ורחוקה יותר מהמישור, ההסתברות שלה גבוהה יותר. כלומר עלינו לקחת למרחקים ולהפוך אותם להסתברויות ביחס למישור. מתמטית נעביר סקאלה של מרחקים לסקאלה של 0-1. אחת הפונקציות שיודעות לעשות זאת **נקראית**

x_1	y
	1
	1
	0
	1
	0
	1

סיגמויד sigmoid : $S(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}} = \frac{e^{\theta^T x}}{1 + e^{\theta^T x}}$ ונשתמש בה ב-LoR
כאשר מכפלה פנימית של טטה עם איקס היא בדיוק המרחק של הנקודה מהמישור ולכן אנחנו לוקחים את המרחק הזה מכניסים לסיגמויד ומקבלים תוצאה בין 0-1.

$$S(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{1+e^x}$$

לכן, נציב את הסיגמויד (מסומן ב-S) **בפונקצית ההיפותזה שלנו**: $h_\theta(x) = S(\theta^T x)$
והפרדיקציה שלנו תהיה 1 אם h0 (x) > 0.5 ואחרת, 0.



נשתמש ב-maximum likelihood כדי להגדיר את פונקציית העלות. נרצה לחזות את הלייבל y בהינתן הדאטה והמישור, ולקבל עבורו את ההסתברות הגבוהה ביותר. ובאופן מתמטי נרצה למקסם את ההסתברות הבאה (הלייקליהוד):

$$P(y|x,\theta) \doteq \left(h_\theta(x)\right)^y \cdot \left(1-h_\theta(x)\right)^{1-y}$$

נזכור כי h0 (x) היא ההסתברות של האינסטנס x להשתייך לקלאס 1.

ולכן נקבל:

- $P(0|x,\theta) = 1 - h_\theta(x)$
- $P(1|x,\theta) = h_\theta(x)$

Score	-∞	-2	0	+2	+∞
Sigmoid (Score)	$\frac{1}{1+e^{\infty}} = 0$		$\frac{1}{1+e^{0}} = 0.5$		$\frac{1}{1+e^{-\infty}} = 1$

<u>פיתוח מתמטי</u>: (לאחר שנניח אי תלות על האינסטנסים נרצה למצוא את המקסימום של מכפלת ההסתברויות המוקפת באדום, לכן נפעיל לן ונגזור ונקבל את הגרדיאנט – פיתוח מלא ומפורט יותר מופיע בהרצאה)

$$P(y|x,\theta) = \left(h_\theta(x)\right)^y \cdot \left(1-h_\theta(x)\right)^{1-y}$$

Assuming independent instances

$$P(D|\theta) = \prod_{d=1}^{m} P(y^{(d)} \mid x^{(d)}, \theta) =$$

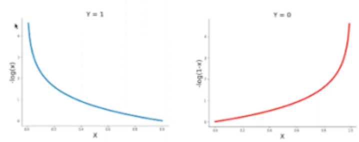
$$\prod_{d=1}^{m} \left(h_\theta(x^{(d)})\right)^{y^{(d)}} \cdot \left(1-h_\theta(x^{(d)})\right)^{1-y^{(d)}}$$

$$\underset{\theta}{argmin} \sum_{d=1}^{m} -y^{(d)} \cdot \ln\left(h_\theta(x^{(d)})\right) - \left(1-y^{(d)}\right) \cdot \ln\left(1-h_\theta(x^{(d)})\right)$$

$$-\sum_{d=1}^{m} y^{(d)}\theta^T x^{(d)} - \ln\left(1+e^{\theta^T x^{(d)}}\right)$$

Cost Function Intuition

- $Cost(x,\theta) = \begin{cases} -\log(h_\theta(x)) & y=1 \\ -\log(1-h_\theta(x)) & y=0 \end{cases}$



<u>**לאחר הצבת הסיגמויד זו הפונקציה שנרצה למזער:**</u>

נמזער את הפונקציה הזו על ידי **גרדיאנט דיסנט** <u>כאשר זו פונקציית העלות</u>:

$$cost(\vec{\theta}) = -\sum_{d=1}^{m} y^{(d)}\theta^T x^{(d)} - \ln\left(1+e^{\theta^T x^{(d)}}\right)$$

הנגזרת עבור כל אינסטנס כאשר i הוא הפיצ'ר ה-i תניב:

$$\frac{\partial}{\partial \theta_i} cost(\vec{x},\vec{\theta}) = -\left(y - S(\vec{\theta},\vec{x})\right) x_i$$

ולכל m נקודות הדאטה מתקיים:

$$\frac{\partial}{\partial \theta_i} cost(\vec{\theta}) = \sum_{d=1}^{m} \left(S(\vec{\theta},\vec{x}^{(d)}) - y^{(d)}\right) x_i^{(d)}$$

וכעת נפעיל גרדיאנט דיסנט.

- Hypothesis function :
$$h_\theta(x) = S(\theta^T x)$$
- Cost function:
$$J(\theta) = \frac{1}{m} \sum_{d=1}^{m} -y^{(d)} \cdot \ln\left(h_\theta(x^{(d)})\right) - \left(1-y^{(d)}\right) \cdot \ln\left(1-h_\theta(x^{(d)})\right)$$
- Goal :
$$\min_\theta J(\theta)$$

ולסיכום, להלן ההיפותזה, העלות ונרצה למצוא טטה אשר ממזערת את העלות.

(כאן אין פתרון על ידי pinv)

<u>**איך נוכל לפתור בעיה שיש בה יותר משני קלאסים עם מפריד ליניארי? גישת one vs all.**</u>
נניח כי יש לנו 3 קלאסים, נמצא לכל אחד מהם מפריד ליניארי עבור כל אחד מהקלאסים, סך הכל 3 מפרידים ליניארים. כאשר תגיע נקודה חדשה נסווג אל הקלאס שביחס לאחרים ההסתברות של נקודה חדשה להשתייך אליו תסווג אליו – ההיפותזה של כל קלאס אומרת מה ההסתברות שאנחנו משתייכים לקלאס זה.
עבור הנקודה העליונה ברור שההיפותזה של קלאס האיקסים תהיה גבוהה מההיפותזה של הריבועים והעיגולים ולכן היא תסווג כאיקס.