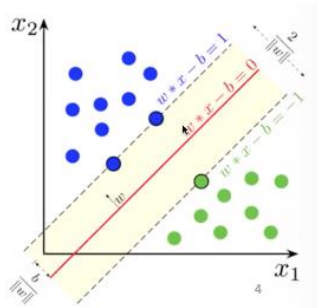
למידה חישובית ממידע: תרגול 7 – SVM

המטרה שלנו: למצוא מפריד לינארי שיכול להפריד את הדאטה (היום נתמקד בהפרדה ל-2 קלאסים).

אלגוריתם SVM מבוסס על 3 רעיונות:

- <u>הקרנל טריק / The Kernel Trick</u> – ממפה את הדאטה למרחב בממד גבוה שבו יותר קל לסווג עם משטחים מחליטים שהם ליניארים.
- <u>שוליים מקסימליים / Max Margins</u> – עבור בעיית ההפרדה הליניארית, ההיפר-מישור בעל השוליים המקסימליים הוא המסווג הליניארי האופטימלי.
- רגולריזציה ו-Soft Margins – מרחיב את ההגדרה לעיל עבור בעיות הפרדה שאינן לינאריות, לאפשר טעויות.



הגדרות:

- מסווג לינארי – פונקציה לינארית (היפר-מישור במרחב הפיצ׳רים) שיכול להפריד דאטה-סט d-ממדי.

$$f(\vec{x}, \vec{w}, b) = sign(\vec{w} \cdot \vec{x} + b)$$

- שוליים Margin(xi) = המרחק בין גבול ההחלטה ו-xi.
- $\text{Margin}(\vec{w}, b) = \min \text{Margin}(x_i)$
- Maximal margin classifier – $\vec{w}, b = argmax\, \text{Margin}(\vec{w}, b)$

<u>הרעיון הוא</u>: לקחת את כל הדאטה שלנו ולעשות עליו איזשהו **מיפוי למרחב אחר** שנקווה שבו יהיה מפריד לינארי טוב יותר.
פונקציית פי היא פונקציית המיפוי שלנו.

<u>הבעיה במיפוי הדאטה היא</u>: המיפוי עצמו היא פעולה יקרה (מבחינת יעילות) וכן העבודה במרחה גבוה היא מאוד יקרה (מבחינת סיבוכיות זמן).
לכן עלינו למצוא דרך לעבוד במרחב גבוה מבלי למפות לממד הזה. עלינו למצוא את הפונקציה שמדמה עבודה במרחב הגבוה = קרנל.

<u>הקרנל טריק / The Kernel Trick –</u>

- נניח כי אנחנו צריכים רק את המכפלה הפנימית במרחב המיפוי (נראה בהמשך למה הנחה זו היא נכונה)
- כלומר, נרצה לקחת שני אינסטנסים x ו-y, נמפה את שניהם על ידי פי למרחב הגבוה ולאחר מכן נבצע עליהם במרחב הגבוה מכפלה פנימית: $\varphi(x) \cdot \varphi(y)$
- אם נוכל למצוא פונקציה שמניבה את אותה התוצאה ״בלי״ למפות, נוכל לצמצם את סיבוכיות זמן הריצה
- פונקציה זו נקראת Kernel, והקרנל-טריק נועד למנוע את המיפוי

<u>דוגמה לקרנל</u>: ניקח את הוקטור הדו-ממדי הבא $x = (x_1, x_2)$ ונבצע עליו מיפוי למרחב תלת ממדי: $\varphi(x) = (x_1^2, \sqrt{2} \cdot x_1 x_2, x_2^2)$

ונחשב את המכפלה הפנימית של פי(x) ופי(y): $x_1^2 y_1^2 + 2x_1 x_2 y_1 y_2 + x_2^2 y_2^2$

כאשר y הוא גם וקטור דו ממדי. התקבל (כפל מקוצר):
$$x_1^2 y_1^2 + 2x_1 x_2 y_1 y_2 + x_2^2 y_2^2 = (x_1 y_1 + x_2 y_2)^2$$
(מהגדרת מכפלה פנימית) <=
$$x_1^2 y_1^2 + 2x_1 x_2 y_1 y_2 + x_2^2 y_2^2 = (x_1 y_1 + x_2 y_2)^2 = (x \cdot y)^2 = K(x, y)$$

<u>דוגמה נוספת</u>: עבור 4-ממדים $x = (x_1, x_2, x_3, x_4)$ והמיפוי הבא הממפה ל-15-ממדים
$$\varphi(x) = (1, \sqrt{2} \cdot x_1, \sqrt{2} \cdot x_2, \sqrt{2} \cdot x_3, \sqrt{2} \cdot x_4, x_1^2, x_2^2, x_3^2, x_4^2, \sqrt{2} \cdot x_1 x_2, \sqrt{2} \cdot x_1 x_3, \sqrt{2} \cdot x_1 x_4, \sqrt{2} \cdot x_2 x_3, \sqrt{2} \cdot x_2 x_4, \sqrt{2} \cdot x_3 x_4)$$

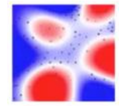
נרצה לחשב את המכפלה הפנימית בין שני וקטורים 4-ממדיים:
$$\varphi(x) \cdot \varphi(y) = 1 + \sum_{i=1}^{4} 2x_i y_i + \sum_{i=1}^{4} x_i^2 y_i^2 + \sum_{i=1}^{3} \sum_{j=i+1}^{4} 2x_i x_j y_i y_j$$

נתבונן בפונקציה הבאה: $(x \cdot y + 1)^2 = (x \cdot y)^2 + 2x \cdot y + 1$ ומתקיים עבורה <=
$$= 1 + \sum_{i=1}^{4} 2x_i y_i + \left(\sum_{i=1}^{4} x_i y_i\right)^2 = 1 + \sum_{i=1}^{4} 2x_i y_i + \sum_{i=1}^{4} \sum_{j=1}^{4} x_i y_i x_j y_j = 1 + \sum_{i=1}^{4} 2x_i y_i + \sum_{i=1}^{4} x_i^2 y_i^2 + \sum_{i=1}^{3} \sum_{j=i+1}^{4} 2x_i x_j y_i y_j$$

לכן $(x*y + 1)^2$ היא פונקציית הקרנל.

פונקציות קרנל

- יש מספר פונקציות קרנל ידועות
- **אנחנו לא צריכים לדעת את המרחב שהקרנל ממפה אליו**
- קרנל פולינומיאלי מדרגה d: $K(x,y) = (\alpha x^T y + \beta)^d$ = בעל 3 פרמטרים
- Radial Basis Function (RBF): $K(x,y) = \exp(\frac{-\|x-y\|^2}{2\sigma^2})$



- Where σ is a parameter
- We can replace $\frac{1}{2\sigma^2}$ with γ -> $\exp(-\gamma\|x-y\|^2)$
- The radius of the "balls" is determined by the parameter $\gamma = \frac{1}{2\sigma^2}$
  - A smaller γ means a larger radius, a lower "model complexity"
  - A larger γ means a smaller radius, a finer grain coverage but may lead to an overfitt

מעניין לדעת כי לאחר פתיחה של RBF קרנל עם טורי טיילור וסיגמה שהיא (1 חלקי שורש 2) נקבל **ממד אינסופי**.
ככל שהסיגמה קטנה יותר, הקרנל מתנהג יותר כמו instance based (כמו בkNN עבור k=1).

קרנל הוא היפר-פרמטר – כך שהקרנלים המצויינים לעיל הם מייצגים משפחות של קרנלים, וכאשר אנחנו מפעילים קרנל אנחנו חייבים להגדיר את הפרמטרים שלו (אלפא, בטא, או ב-RBF סיגמה) אחרת למעשה לא בחרנו קרנל ספציפי.

קרנל פרספטרון Kernel Perceptron

- נשים לב (כפי שצויין בהרצאה) כי בכל שלב של הפרספטרון, אם יש צורך בעדכון (כלומר הדגימה d סווגה באופן לא נכון = שגיאה) אנחנו מוסיפים חלק/שבר קטן של Xd למשל wi. המשמעות היא שיש רק אינסטנסים מסויימים שמשנים את w, רק האינסטנסים שאנחנו "טועים" בהם. כלומר תמיד יהיו לנו אינסטנסים מעניינים יותר ומעניינים פחות, כאשר המעניינים יותר הם אלה שאנחנו עושים עליהם טעות ואלו שיעזרו לנו לשפר את המודל ולבנות את המפריד.
- כלומר, אם בסופו של דבר w מקבל כל פעם חלק אחר מ-TdXd (אם דגימה d היא קיבלה קלסיפיקציה שגויה), נקבל שהמשקלים הם קומבינציה לינארית של חלק מדגימות הדאטה: $w = \sum_d \alpha_d t_d x_d$
- כאשר ad אי שלילי (גדול שווה מ-0).
- אינסטנסים שלא משפיעים על תהליך הלמידה, הם אינסטנסים שעבורם ad = 0.
- מכאן שנשנה את פונקציית ההחלטה באופן הבא: $f(x) = \vec{w}\cdot\vec{x} = \left(\sum_d \alpha_d t_d x_d\right)\cdot\vec{x} = \sum_d \alpha_d t_d(\vec{x}_d\cdot\vec{x})$
- האינסטנסים Xd שעבורם ad שונה מ-0 הם האינסטנסים שאנחנו צריכים והם נקראים **"support vectors"**
- כדי להשתמש בצורה הזו של פונקציית ההחלטה עלינו לעדכן את שלב העדכון של הפרספטרון:
$$\text{if } \left(t_i \sum_d \alpha_d t_d(\vec{x}_d\cdot\vec{x}_i)\right) < 0: \quad \alpha_i = \alpha_i + \eta$$
- כך שבמקום לעדכן את w, נעדכן את אלפא ad.
- כאשר אלפא היא המשקל של האינסטנס.
- **בפרספטרון הדואלי**, שהוא כמו הפרספטרון מלבד ההחלפה בין המשקלים w, למשקלים אלפא, יש לנו את **המכפלה הפנימית** של דגימה d (xd) עם הדגימה החדשה x מה שיסייע לנו לעבור על ידי פי לממד גבוה יותר, ומעודד אותנו למצוא פונקציית **קרנל** כדי למנוע מיפוי.
- **מה שמוביל אותנו לקרנל פרספטרון – השלב הראשון לכיוון אלגוריתם SVM.**

The Dual Perceptron algorithm:
- Initialize each α_i to zero
- Repeat until convergence (no error):
  - For each x_i in D compute:
    - $o_i = \sum_{d\in D}\alpha_d t_d(\vec{x}_d\cdot\vec{x}_i)$
    - If $t_i o_i < 0$
      - $\alpha_i = \alpha_i + \eta$

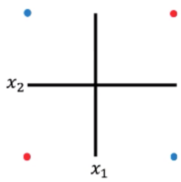
The Kernel Perceptron algorithm:
- Initialize each α_i to zero
- Repeat until convergence (no error):
  - For each x_i in D compute:
    - $o_i = \sum_{d\in D}\alpha_d t_d\left(\varphi(\vec{x}_d)\cdot\varphi(\vec{x}_i)\right) = \sum_{d\in D}\alpha_d t_d K(\vec{x}_d,\vec{x}_i)$
    - If $t_i o_i < 0$
      - $\alpha_i = \alpha_i + \eta$

דוגמה הרצה של קרנל פרספטרון:

יש לנו 4 אינסטנסים, ו-2 קלאסים, לכן המרחב הוא דו-ממדי, t הוא הלייבל. זוהי בעיית ה-XOR.

נבחר את הקרנל הבא : $K(x_i,x_j)=\varphi(x_i)\cdot\varphi(x_j)=(x_i\cdot x_j)^2$

להלן התוצאות המוצגות בטבלה = הדגימות החדשות בממד 4 .



x_1	x_2	t
1	1	1
-1	1	-1
-1	-1	1
1	-1	-1

- **נאתחל (אטה = 1):**
  - $\alpha=[\alpha^1,\alpha^2,\alpha^3,\alpha^4]=[0,0,0,0]$ (אלפא יהיה וקטור ה-0 בגודל 4)

K	x^1	x^2	x^3	x^4
x^1	4	0	4	0
x^2	0	4	0	4
x^3	4	0	4	0
x^4	0	4	0	4

- **נבצע בדיקה עבור האינסטנס הראשון i=1 [4,0,4,0]:**

$$\sum_{d\in D}\alpha_d t_d K(\vec{x}_d,\vec{x}_i)=$$
$$0*4-0*0+0*4-0*0=0$$
$$sgn(0)=-1\rightarrow\alpha^1\mathrel{+}=1$$

וכאן אלפא הופכת להיות [1,0,0,0] מפני שיש לנו טעות, קיבלנו 0 (שמסמל כאן את הקלאס 1-) בעוד שה-target value של האינסטנס הראשון הוא 1!

- **נבצע בדיקה עבור האינסטנס השני i=2 [0,4,0,4]:**

$$\sum_{d\in D}\alpha_d t_d K(\vec{x}_d,\vec{x}_i)=$$
$$1*0-0*4+0*0-0*4=0$$
$$sgn(0)=-1$$

באיטרציה הזו אנחנו לא צריכים לעדכן את האלפות כי צדקנו. (נשים לב ש-0 שקול ללייבל 1-)

- **נבצע בדיקה עבור האינסטנס השלישי i=3 [4,0,4,0]:**

$$\sum_{d\in D}\alpha_d t_d K(\vec{x}_d,\vec{x}_i)=$$
$$1*4-0*0+0*4-0*0=4$$
$$sgn(4)=1$$

התקבלה תוצאה חיובית שהדבר מסמל קלסיפיקציה לקלאס 1, ואכן צדקנו מהיות שהלייבל של האינסטנס השלישי הוא 1.

- **נבצע בדיקה עבור האינסטנס הרביעי i=4 [0,4,0,4]:**

$$\sum_{d\in D}\alpha_d t_d K(\vec{x}_d,\vec{x}_i)=$$
$$1*0-0*4+0*0-0*4=0$$
$$sgn(0)=-1$$

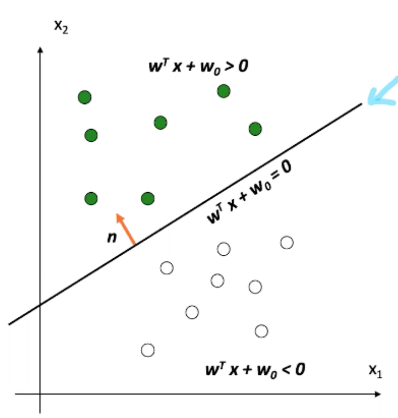
התקבל 0 שמשמעותו קלסיפיקציה לקלאס 1-, וזה סיווג נכון כי האינסטנס הרביעי והאחרון אכן שייך לקלאס 1-.

- **הדבר האחרון שעלינו לעשות הוא לבצע בדיקה על האינסטנס הראשון שוב, שבאיטרציה הקודמת האלפא הניבה עבורו שגיאה.**

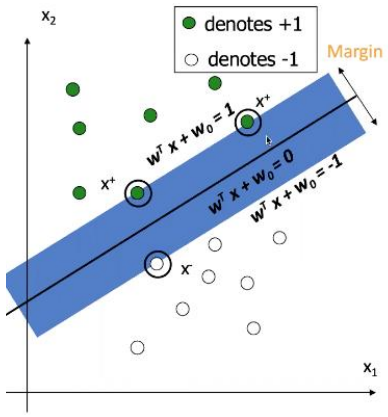
$$\sum_{d\in D}\alpha_d t_d K(\vec{x}_d,\vec{x}_i)=$$
$$1*4-0*0+0*4-0*0=4$$
$$sgn(4)=1$$

ואכן מתקבלת תוצאה חיובית כנדרש **ובכך מסתיים האלגוריתם עם אלפא [1,0,0,0] שמניבה פתרון עבור בעיית ה-XOR!**

מיקסום השוליים – Max Margin



- בהינתן פונקציה לינארית f(x) : $f(x)=w^Tx+w_0$
- ובהינתן ההיפר-מישור של מרחב הפיצ'רים
- הנורמל (וקטור יחידה) של הווקטור המייצג את ההיפר מישור הינו (הווקטור לחלק לנורמה שלו) : $n=\frac{w}{\|w\|}$
- השאלה שאנו שואלים – איזה מישור מפריד בין נקודות לבנות לירוקות הוא המישור בו הכי טוב לבחור, שימזער את הטעות? היוריסטיקה (ההנחה הלא מוכחת) אומרת ש**ככל ששולי המישור רחבים יותר, כך המישור יותר טוב בהכללה.**



כדי להבטיח מישור בעל שוליים מקסימליים נדרוש :

- <u>שכל הנקודות הירוקות יקיימו את המשוואה הזו :</u>

$$t_d = +1,\ w^T x_d + w_0 \geq 1$$

כלומר יהיו רחוקות לפחות ב-1 מהמישור.

- <u>וכל הלבנות יקיימו את המשוואה הזו :</u>

$$\text{For } t_d = -1,\ w^T x_d + w_0 \leq -1$$

כלומר יהיו רחוקות מהמישור לפחות במינוס 1 (ב-1 מתחת למישור).

ידוע לנו כי : $w^T x^+ + w_0 = +1$, $w^T x^- + w_0 = -1$ רוחב השוליים הוא :

$$M = (x^+ - x^-) \cdot n$$
$$= (x^+ - x^-) \cdot \frac{w}{\|w\|}$$
$$= \frac{2}{\|w\|}$$

כעת נרצה למקסם את הביטוי שקיבלנו עבור רוחב השוליים, ולמכן מטרתנו היא : $\text{Maximize } \frac{2}{\|w\|}$

אבל, המיקסום הוא תחת האילוצים הבאים :

$$\text{For } t_d = +1, w^T x_d + w_0 \geq 1$$
$$\text{For } t_d = -1, w^T x_d + w_0 \leq -1$$

שמשמעותם – לא יהיו נקודות בתוך השטח של השוליים (safe zone) של המישור וכן, המישור יפריד בין הקלאסים.

ניתן לפתור בעיית אופטימיזציה זו ע"י quadratic programming, שבשל הקורונה לא נתעמק בפתרון עבורה.

המטרה שלנו שקולה להביא למינימום את הביטוי : $\frac{1}{2} \cdot \|w\|^2$ **תחת האילוץ הבא:** $t_d(w^T x_d + w_0) \geq 1$ **שזהו למעשה אילוץ עבור כל נקודת דאטה d, ועל כן יש לנו אילוצים כמספר נקודות הדאטה.**

כעת, נוכל להחליף את w באלפות שלנו – ע"י כופלי לגרנז'.
כך נוכל למצוא את הגרדיאנט עבור w ו-w0 (לא נכנס לכל התהליך בדרך)... ובסופו של דבר נגיע ל :

- Minimize

$$\min_{w,w_0} \max_{\alpha_d} L(w, w_0, \alpha_d) = \min_{w,w_0} \max_{\alpha_d} \frac{1}{2}\|w\|^2 - \sum_d \alpha_d (t_d(w^T x_d + w_0) - 1)$$

- Subject to:

$$\alpha_d \geq 0$$

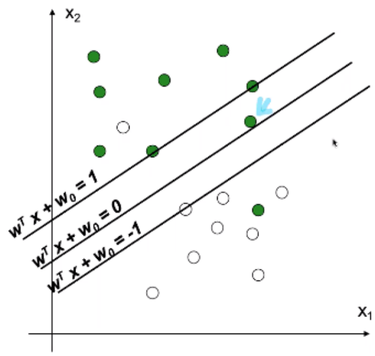
<u>המשוואה הדואלית של ה-SVM בה נרצה למקסם את השוליים בהינתן האילוצים</u>

$$\sum_d \alpha_d - 1/2 \sum_d \sum_e \alpha_d \alpha_e t_d t_e x_d^T x_e$$
$$\sum_d \alpha_d t_d = 0, \alpha_d \geq 0$$

<u>מה שמאפשר לנו להחליף את המיפוי ל"קרנל-טריק":</u>

$$\sum_d \alpha_d - 1/2 \sum_d \sum_e \alpha_d \alpha_e t_d t_e\ K(x_d, x_e)$$

רגולריזציה ו-Soft Margin



- רוב הדאטה שניתקל בו הוא למעשה **לא ניתן להפרדה לינארית** (דאטה "רעשי", outliers וכו'...)
- משתני **Slack** יכולים לאפשר חריגה מהשוליים (לא בהכרח מיס-קלסיפיקציה, אלא חדירה ל-safe zone **מרווח ביטחון**) של דאטה מורכב או "רועשי"
- נסמן כל חריגה בקסי (הסימון של הנחש)– המרחק מהשול הרלוונטי לנקודה הספציפית.

  ונשנה את בעיית האופטימיזציה ל**מיזעור** של: $\frac{1}{2}\|w\|^2 + \gamma \sum_d \xi_d$
- **ככל שהשוליים רחבים יותר, סכום הקסי יגדל, יהיו יותר חריגות.** על החריגות יש לנו היפר-פרמטר **גמא** שמכפיל את סכום החריגות. ככל שהגמא (סקלר) גבוה יותר אנחנו נשלם יותר על כל חריגה, ולכן הגמא מאזן בין מיקסום השוליים לבין תשלום החריגות.
- **כך האילוצים שלנו לבעיית האופטימיזציה ישתנו:** $t_d(w^T x_d + w_0) \geq 1 - \xi_d \quad \xi_d \geq 0$

  • Minimize

$$\min_{w,w_0,\xi_d} \max_{\alpha_d,\mu_d} L(w, w_0, \xi_d, \alpha_d, \mu_d) =$$

$$\min_{w,w_0,\xi_d} \max_{\alpha_d,\mu_d} \frac{1}{2}\|w\|^2 + \gamma \sum_d \xi_d - \sum_d \alpha_d (t_d(w^T x_d + w_0) - 1 + \xi_d) - \sum_d \mu_d \xi_d$$

  • Subject to:

$$\alpha_d \geq 0 \quad \mu_d \geq 0$$

  • Dual - maximize

$$\sum_d \alpha_d - 1/2 \sum_d \sum_e \alpha_d \alpha_e t_d t_e x_d^T x_e$$

$$\sum_d \alpha_d t_d = 0 \quad 0 \leq \alpha_d \leq \gamma$$

-
- **נעבור מהמצב הפרימלי למצב הדואלי בו נוכל להשתמש בקרנל טריק וברגע שנפתור את המשוואה שנמצאה לעיל נמצא פתרון עבור בעיית האופטימיזציה החדשה שלנו שכוללת את החריגות.**