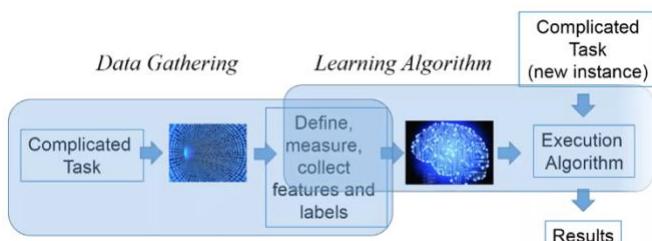


## למידה חישובית ממדיע: הרצאה 1



### מושגים בלמידה חישובית ממידע:

- אוסף של נתונים, דוגמאות, טבות ורעות (חיוביות ושליליות) שיקראו samples instances
- איסוף מידע אודות הדוגמאות הנ"ל, שיקרא features תכונות של הדוגמאות (חלקו רלוונטי וחולק איןין, לא בהכרח נדע מראש מה רלוונטי עבורנו ומה לא רלוונטי עבורנו לדעת)
- הפעלת אלגוריתם הלמידה שיצור אלגוריתם מבצע = שידוע לקבל דוגמאות חדשות ולהשאיב ערכו

### סוגי למידה :

- גורסיה : בהינתן דוגמיה { $y_i, x_i$ } נמצא פונקציה  $f(x)$  כך ש  $y_i = f(x_i)$
- קלסיפיקציה : בהינתן דוגמיה { $y_i, x_i$ } כאשר  $y_i$  0 או 1 בדעתה אימון, נקבע, עבור כל  $x$  חדש אם  $x$  שייך ל-0 או ל-1.
- Density Estimation
- Clustering

### גורסיה ליניארית

Square Feet (x)	House Price in \$1000s (y)
1400	245
1600	312
1700	279
1875	308
1100	199
1550	219
2350	405
2450	324
1425	319
1700	255

דוגמה: בהינתן גודל של בית נרצה לקבוע מחיר עבورو. להלן נתונים אימון שלנו שמכיל 10 דוגמאות ו-1 feature. מתוך נרצה למדוד פונקציה המכילה. ההנחה היא שאנו נרוצים מודל ליניארי – וכך נעשה גורסיה ליניארית – הדבר מגביל את מרחיב היפותזות שלנו לモרחב הפונקציות הליניאריות בלבד.

כיצד מייצגים את היפותזה  $f$  במרחב הפונקציות הליניאריות? על ידי משווה ליניארית: בדוגמה שלנו יש פיצ'ר אחד, מה עושים באשר לש-מספר פיצ'רים?

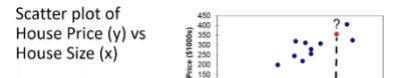
כאשר  $X$  הוא וקטור הפיצ'רים שלנו, עבור כל דוגמיה  $i$ .  

$$X^{(i)} = (X_1^{(i)}, X_2^{(i)}, \dots, X_n^{(i)})$$
  

$$\theta = (\theta_0, \theta_1, \theta_2, \dots, \theta_n)$$
  $X^{(i)} = \theta + \epsilon^{(i)}$   $\epsilon^{(i)}$  הן נוחות, נסיף פיצ'ר "קבוע" עבור כל  $i$ ,  $\theta_0$  הוא פיצ'ר קבוע,  $\theta_1, \theta_2, \dots, \theta_n$  הם פיטרים, היפותזה הליניארית שלנו תהיה מוגנת על ידי וקטור הפרמטרים  $\theta$ .

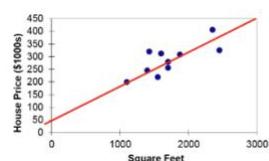
\* We say that  $\theta = (\theta_0, \theta_1, \theta_2, \dots, \theta_n)$  is the vector of parameters that defines our function (or our model)

כך שמותקינים:



#### Prediction:

Given house of size  $x$ , what would be its price  $y = f(x)$ ?



- We assume/hypothesize that the relationship between the observed and the independent/explained variable is linear and thereby conduct our search

- This is our **Hypotheses Space** – all linear functions

וקטור הפרמטרים טטה למעשה מגדר את הפונקציה שלנו (המודל שלנו), וכך למעשה במדד גובה ותראה הפונקציה שנרצה ללמד, אך עליינו למדוד את הטוטות. נרצה למצוא את הטוטות הטובות ביותר – שינוי את המודל הטוב ביותר.

#### המכפלה הפנימית של טטה באיקס החדש ( $x$ ) תנייב לנו את מחיר הבית.

הданה אימון training data תעוזר לנו להגיע לטיטה הטובה ביותר. בעולם "מושלים" היינו רוצים למצוא טטה שמניבת טעות 0, ככלומר ממש למצוא מודל שעובר בכל הנקודות, אלגוריתם למידה כזה נקרא learner. זהו לא המקורה בעולם האמתי וכן לא בדוגמה שלנו. אבל עדין נרצה את הטעות הקטנה ביותר. אז איך נמדד את השגיאה שלנו? הטעות בכל דוגמיה (instance) הינה:

$$(\theta_0 x_0^{(i)} + \theta_1 x_1^{(i)} + \theta_2 x_2^{(i)} + \dots + \theta_n x_n^{(i)} - y^{(i)})$$

cutet נבצע ממוצע על כל דוגמאות האימון במטרה לקבל את פונקציית העלות שלנו

#### cost function:

$$J(\theta) = \frac{1}{2} \cdot \frac{1}{m} \sum_{i=1}^m (\theta \cdot x^{(i)} - y^{(i)})^2$$

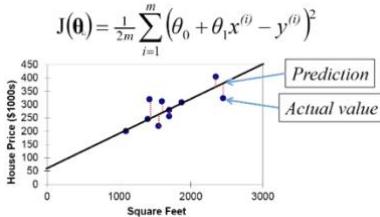
או משתמשים בשגיאה הריבועית כדי שטוענו בכיוונים שונים לא תتبטל, וגם זו פונקציה smoother שאנינה גזרה ב-0.

כך נוכל לחשב את הטוטות עבור כל טטה על ידי חישוב השגיאה. ומכיון שנרצה להגיע לטטה בעלת השגיאה המינימלית, נרצה למצוא את הטטה שמניבת ערך מינימלי של הפונקציה  $J$  – פונקציית העלות. הטוטה ה- $\theta$  המודול.

- Hence, our best hypothesis  $\theta^*$  would be the one that minimizes the cost function:

$$\theta^* = \arg \min_{\theta} J(\theta) = \arg \min_{\theta} \left[ \frac{1}{2m} \sum_{i=1}^m (\theta \cdot x^{(i)} - y^{(i)})^2 \right]$$

- How can we find it?



- The directional derivative in the direction of the vector  $u = (u_1, u_2)$  (a scalar!) can also be written as:  $D_u f(x_1, x_2) = \nabla f \cdot \bar{u} = \|\nabla f\| \cos \beta = \|\nabla f\| \cos \theta$
- Where  $\beta$  is the angle between  $u$  and  $\nabla f$
- However,  $\cos \beta \leq 1$ .
- Therefore:
  - The greatest increase in the function happens in the direction of the gradient (i.e.  $\beta=0$ )
  - The greatest decrease is in the direction  $-\nabla f$  (i.e.  $\beta=180^\circ$ )

השאלה, איך מבאים למינימום את פונקציית המחיר / הูลות, שהיא בעלי מספר משתנים?  
הדרך הראשונה למצואו מינימום של פונקציה היא לגזר אותה ולהשווות אותה ל-0.  
הדרך השנייה: גרדיאנט דיסנט.  
שתי השיטות יכולות להוביל למינימום מקומי, ולא גלובלי כפי שנראה!  
זכיר שגורת חלקית מוחשבת על ידי גזירה על פי משתנה אחד כך שכל השאר נתרים קבועים.  
הגורת הכוונית בכיוון  $u$  מוגדרת:

$$D_u f(x_1, x_2) = \lim_{s \rightarrow 0} \frac{f(x_1 + su_1, x_2 + su_2) - f(x_1, x_2)}{s} = \left( \frac{df}{ds} \right)_u$$

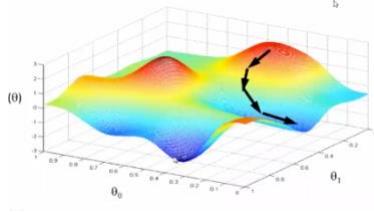
Define the **GRADIENT of  $f$ :**  $\nabla f = \left( \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2} \right)$   
הגרדיינט של פונקציה  $f$  מוגדר להיות:

$$D_u f(x) = \nabla f(x) \cdot u$$

**משפט:** עבור כל כיוון  $(u_1, u_2) = u$  ולכל נקודה  $(x_1, x_2) = x$  מתקיים:

כלומר הגורת הכוונית בכיוון  $u$  בנקודת  $x$  שווה למכפלה הפנימית של הגרדיינט של הפונקציה  $f$  בנקודת  $x$  עם וקטור הכוון  $u$ .

במטרה להגעה למינימום של פונקציהначילה בנקודת מסוימת ונלך נגד הגרדיינט. תחילה זה נקרא **גרדיינט דיסנט**.  
גדיאנט דיסנט לא מבטיח מינימום מקומי! ומוראים גם לאלגוריתמים שאינם מניבים פונקציה שהיא בהכרח לינארית.



- Start with some value for  $\theta = (\theta_0, \theta_1, \theta_2, \dots, \theta_n)$
- Repeat until you reach a minimum:

  - For all  $j$ ,  
Update  $\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$

- $\alpha > 0$  is a parameter of the algorithm called the learning rate
- Updates are simultaneous (in all  $n+1$  directions)
- In the general case this process can still be trapped in local minima!

### אלגוריתם גרדיאנט דיסנט

$\theta = (\theta_0, \theta_1, \theta_2, \dots, \theta_n)$  נתחל עם ערך רנדומלי כלשהו של טטה •

עד שנגיע למינימום ובצע את הצעד הבא: (הליכה נגד הגרדיינט) •

עבור כל  $j$ , נעדכן את טטה  $j$  להיות טטה  $j$  עם "צעד קטן" נגד הגרדיינט.

כלומר נעדכן את טטה  $j$  להיות טטה  $j$  עם "צעד קטן" נגד הגרדיינט. •

אלפא היא פרמטר של האלגוריתם שנקרא learning rate גודל צעד העידכון שМОוגדר. •

כאשר חישוב הגרדיינט של פונקציית הูลות הינו:

$$\frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1, \dots, \theta_n) = \frac{\partial}{\partial \theta_j} \frac{1}{2m} \sum_{i=1}^m (h_\theta(x^{(i)}) - y^{(i)})^2$$

$$= \frac{1}{2m} \sum_{i=1}^m \frac{\partial}{\partial \theta_j} (\theta_0 + \theta_1 x_1^{(i)} + \dots + \theta_j x_j^{(i)} + \dots + \theta_n x_n^{(i)} - y^{(i)})^2$$

$$= \frac{1}{2m} \sum_{i=1}^m 2(\theta_0 + \theta_1 x_1^{(i)} + \dots + \theta_j x_j^{(i)} + \dots + \theta_n x_n^{(i)} - y^{(i)}) \cdot x_j^{(i)}$$

$$= \frac{1}{m} \sum_{i=1}^m (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_j^{(i)}$$

**האלגוריתם הסופי להפעלת גרדיאנט דיסנט על פונקציית הูลות (השgiaה של טטה):**

מתחלים מערך התחלתי של טטה

חוורים על הצעד עד תנאי העזירה: מעדכנים טטה  $j$  חדש להיות טטה  $j$

הנוכחי פחותה learning rate (אלפה) כפול הגורת החלקית של פונקציית

הูลות ביחס לטטה  $j$ , מחושב על הטטה הנוכחי.

זכור ש-  $x_0(i) = 1$

- Initialize  $\theta = (\theta_0, \theta_1, \theta_2, \dots, \theta_n)$
- Repeat until you reach a minimum (or stop condition):
  - For all  $0 \leq j \leq n$ ,

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^m (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_j^{(i)}$$

- In words: set the new  $\theta_j$  to the current  $\theta_j$  minus the learning rate ( $\alpha$ ) times the partial derivative of the error function with respect to  $\theta_j$ , computed at the current  $\theta$ .

- Also remember that  $x_0^{(i)} = 1$

אם המטריצה  $X$  הייתה ריבועית, היינו משתמשים באינברס שלה.

#### סודו אינברס:

- Setting the gradient to zero yields the necessary condition for minimum (see notes):

$$X^T X \cdot \vec{\theta} = X^T \vec{Y}$$

- Now,  $X^T X$  is square and often nonsingular and so we can solve for  $\vec{\theta}$  uniquely as:

$$\vec{\theta} = \text{pinv}(X) \vec{Y} \quad \text{where} \quad \text{pinv}(X) = (X^T X)^{-1} X^T$$

- The  $n \times m$  matrix  $\text{pinv}(X) = (X^T X)^{-1} X^T$  is called the **pseudo inverse** of  $X$  (which is  $m \times n$ )

- If  $X$  is square it is just its inverse.

#### הערות לגבי סודו אינברס:

- תיכן מצב בו למכפלה לא יהיה אינברס, ניתן לפתור זאת עם פעולות אלגוריות אשר לא נתעמק בהן בקורס.
- הפינב לא יעבור לכל פונקציית עלות  $j$ , גרדיאנט דסנט הוא כלל יותר.
- גרדיאנט דיסנט מאפשר פרליזציה.

נשתמש בטכנית של רגסיה גם עבור פונקציות פולינומיאליות:

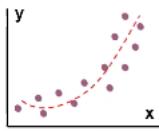
## Polynomial Regression

- We can expand our feature space by using functions of the original features.

- For example, if we want to use a cubic function feature space we can define:

$x_0 = 1, x_1 = x, x_2 = x^2, x_3 = x^3$   
then use regular regression and in essence we are learning the function

$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$$



## למידה חישובית ממכירען | תרגול 1 Linear Regression –

### וגרשייה לינארית

- המטרה היא למצוא פונקציה שמתארת באופן הכי טוב את הדadata שלנו, מה צריך להחליט כדי למצוא את הפונקציה הזו?
- סוג הפונקציה (لينארית, פולינומיאלית),
  - ממד שאומר האם הפונקציה טובה או לא טובה
  - תהליך שמשפר את הפונקציה בהתאם לממד שבחרנו

הגדירות:

$$פונקציית היפותזה - h_{\theta}(x) = \theta_0 + \theta_1 \cdot x_1$$

ההיפותזה מניבת את הדוט פרודקט = מכפלה פנימית בין  $x$  לוקטור טטה.

(בדוגממה שלנו,  $x$  הוא מחיר הדירה, טטה<sub>0</sub> מטאר את המרכיב הראשי הציגים, וטטה<sub>1</sub> הוא המקדם של  $x$ )

i.e.  $f(x^{(i)}) = x^{(i)} \cdot \theta + \theta_0 = y^{(i)}$  יי' הוא הפלט הנוכחי עבור פונקציית המטרה על האינפוצ' א', דוגמה תיאורטיבית (שלא בהכרח מתקיימת):

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$$

פונקציית עלות: בדרך כלל ריבועים פחותים (Mean Square Error = MSE) שהוא הפונקציה:

הפרדייקציה פחות הטעות target value는, מועלה בربיע ומציעים את השגיאה עבור כל  $m$  הדוגמאות.

$$\min_{\theta_0, \theta_1} J(\theta_0, \theta_1)$$

נניח שנרצה למצוא טטה<sub>0</sub> וטטה<sub>1</sub> שմבאים ל邏ינומים את פונקציית העלות, כלומר האלגוריתם הנאיובי ביותר שנדע להציג עבור בעיה זו הינו:

נחש ערכים ונדומלים התחלתיים עבור טטה<sub>0</sub> וטטה<sub>1</sub> (במקרה זה יש פיציר יחיד)

מבצע את הצעד הבא עד אשר נתכנס לערך מינימלי: נעדכן את ערכי טטה<sub>0</sub>, טטה<sub>1</sub> כך שאנו קרובים יותר ל邏ינומים

מה קורבה באשר יש לנו יותר פיציר יחיד: נניח שיש לנו פיצירם

$X$  וטטה יהיו וקטורים באורך  $m$  (ועוד 1 לביאס טרייק)

$$h_{\theta}(\bar{x}) = \theta_0 + \bar{\theta} \cdot \bar{x} = \theta_0 + \theta_1 x_1 + \dots + \theta_n x_n$$

וההיפותזה שלנו תיראה באופן הבא:

הערך בסיום זה הוא  $x$ -ים ו- $y$ -ים לכן הפלט של האלגוריתם הלמידה הוא הטעות שלנו.

### נדיר גודיאנט

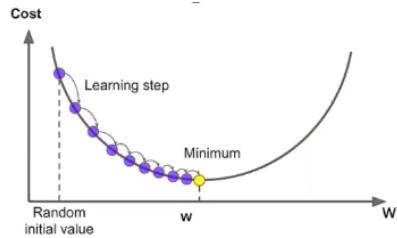
גרדייאנט הוא הכללה רב-משתנית של הנגזרת. בעוד שנגזרת יכולה להיות מוגדרת על פונקציות בעלות מסוימות משתנים, הגרדייאנט תופס את מקום הנגזרת. הגרדייאנט הוא בעל ערכים וקטוריים, בניגוד לנגזרת שהיא פונקציה בעלת ערכים סקלריים. בדומה לנגזרת, הגרדייאנט מייצג את המשיק של גרף הפונקציה. ליתר דיוק, הגרדייאנט מצבייע בכיוון בו הפונקציה "עליה ביותר", וערך הוא השיפוע של הגרף בכיוון זה.

$$\nabla f = \left( \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right) \quad y = f(x_1, x_2, \dots, x_n) \quad \text{הינו:}$$

המשמעות היגיאומטרית: גראדייאנט הוא וקטור של הנגזרות החלקיים, כמו כל וקטור, לגראדייאנט מייצג את השיפוע של המשיק לפונקציה מבחינה גיאומטרית. אלך בה אגיע לפסגה מהנקודה בה אני נמצא, הגרדייאנט מייצג את השיפוע של המשיק לפונקציה מבחינה גיאומטרית.

כיצד נוכל להשתמש בגרדייאנט במטרה למצוא את המניינים של פונקציה? נלק לכיוון "ההפוך", הכוון בו נלק לירידה התולולה ביותר והוא הכוון ההפוך אליו מצבייע הגרדייאנט.

## גרדיינט דיסנט Gradient Descent



- גראדיינט דיסנט הוא אלגוריתם אופטימיזציה איטרטיבי עבור מציאת מינימום של פונקציה.
- מתחילה על ידי איתחול וקטור הפרמטרים טטה עם ערכים רנדומליים
- לאחר מכן, בכל שלב, GD מחשב את הגרדיינט המקומי של פונקציית הולות (MSE, J...) ביחס לקטור טטה, והוא הולך בכוון בו הירידה היא הבלתי ביוור.
- האלגוריתם ממשיך עד שהוא מתקנס למינימום (בו הגרדיינט הוא 0).

**אלפא = learning rate**, שולחת בגודל הצעד אותו לוקחים לכיוון המינימום.

### Algorithm:

- Guess some value for  $\theta_0$  and  $\theta_1$
- Repeat until error is small enough
  - Update  $\theta_0 := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) = \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^m (\theta_0 + \theta_1 x_1^i - y^i) \cdot x_1^i$
  - Update  $\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^m (\theta_0 + \theta_1 x_1^i - y^i) \cdot x_1^i$

### Simultaneous updates

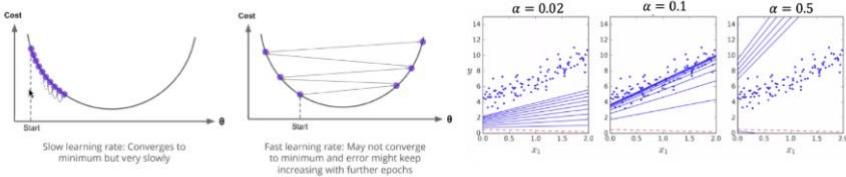
#### Correct:

- $temp0 = \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$
- $temp1 = \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$
- $\theta_0 = temp0; \theta_1 = temp1;$

#### Incorrect:

- $\theta_0 = \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$
- $\theta_1 = \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$

The cost function will have been updated



כאשר אנו נמצאים מימין למינימום, הגרדיינט חיובי ולכן נזוז שמאלה

כאשר אנו נמצאים משמאלי למינימום, הגרדיינט שלילי, ולכן נזוז ימינה

כאשר אלפא קטן מדי, האלגוריתם יתכנס מאד לאט (בדוגמה 0.02)

כאשר אלפא גדול מדי, יתרן ו"נדلغ" על המינימום וכך נקבל התוצאות (בדוגמה 0.5)

בגרף המתואר לעיל ניתן לראות כי ככל שאנחנו מתקרבים למינימום נשים לב שהצעדים לכיוונו נעשים קטנים יותר ויותר, כאמור גודל הצעד, אלפא, מוכפל בגרדיינט בכל שלב באלגוריתם בו אנו מעדכנים את הטוטו, لكن אינטואטיבית לא צריך להקטין את האלפא. אבל יש אלגוריתמים שאכן דורשים הקטנה של אלפא, מפני שיש היתכנות ל"דילוג" מעיל המינימום.

## גישה שונות לאלגוריתם גרדיינט דיסנט

### סטוכסטי / באץ' Stochastic Gradient Descent – \ Batch Gradient Descent

עד כה חישבנו את הגרדיינט התחשבנו בכל הדאטה

במקרים זאת, נכל להשתמש רק בחלק מהדאטה ולדבר יש חסרוןות ויתרונות (סטוכסטי).

חסרוןות: פחות מדויק (מן שלא כל הדאטה נלקח בחשבון)

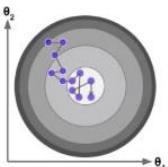
יתרונות: מהיר יותר

Batch משמשו לנו משתמשים בכל הדאטה (לפעמים אנשים אומרים Batch אבל הם מתכוונים רק לחלק מהדוגמאות בדאטה אימון) הגישה השונה היא הגישה הסטוכסטית – אינטנס יחיד בכל איטרציה.

### הגישה הסטוכסטית:

הבעיה העיקרית בגישה Batch הינה העבודה כי משתמשים בכל ה- training set במטרה לחשב את הגרדיינט בכל שלב של האלגוריתם, מה שהופך את האלגוריתם לאיטוי יותר ככל שהיא training set.

הגישה הסטוכסטית בוחרת דוגמאות רנדומלית ב- training set בכל שלב של האלגוריתם ומחשבת את הגרדיינט רק על הדוגמיה היחידה זו. מה שהופך את האלגוריתם לרובה יותר מהיר, וגם אפשר אימון על training set מאוד גדול, מכיוון שרק דוגמיה אחת צריכה להישמר בזיכרון בכל איטרציה (יעילות מחינת זיכרון).



מצד שני, האלגוריתם הסטוכסטי הרבה פחות רגולרי. במקומות לרדרת באופן מותן עד אשר נגיעה למינימום, פונקציית העלות מעלה ומטה וرك במנז'ו יורדת. לכן ברוע שהאלגוריתם נעצר, ערכי הפרמטרים הם אכן טובים, אך אינם אופטימליים.

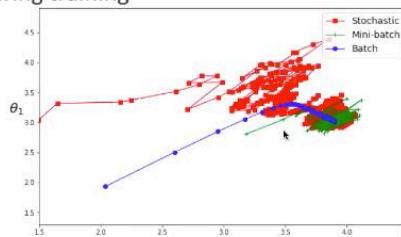
כasher פונקציית העלות אינה קעורה (כך שבין כל שתי נקודות הפונקציה עצמה נמצאת מתחתיו לקו העובר ביניהו), הדבר יכול לעזור לאלגוריתם לקפץ מוחץ למינימום המקומי. לכן שימוש בגישה הסטוכסטית יכול באופן סביר יותר למצוא את המינימום הגלובלי מאשר שימוש ב-Batch. לכן הרנדומליות של הסטוכסטית טוביה לשבר היחס מאופטימליות לוקאלית, אבל יכולה גם להיות רעה מפני שהמשמעות היא שהאלגוריתם לעולם לא יתכנס למינימום עצמו.

### מיני – בא' גרדיאנט דיסנט / Mini-Batch Gradient Descent

בכל שלב, במקום לחשב את הגרדיינט על כל הדוגמאות על כל הדוגמאות ב- training set (כפי שעושים ב-Batch GD), או על דוגמיה אחת רנדומלית (כפי שעושים ב- Stochastic GD), מיני-בא' GD מחשב את הגרדיינטים על קבוצות רנדומליות קטנות של דוגמאות שנבחרו随机地 .minibatches

היתרון העיקרי של Mini-Batch GD על פני הגישה הסטוכסטית הוא שהוא מקנה שיפור אופטימלי ביצועים של החומרה על אופרציות על מטריצות, בעיקר כאשר משתמשים ב-GPUs. התהיליך של האלגוריתם במרחב הפרמטרים הוא יותר מותן (פחות "משתוללי" כמו שמתורחש בגישה הסטוכסטית), במיוחד כאשר הקבוצות המכוגנות mini-batches הן יחסית גדולות. כתוצאה לכך, Mini-Batch GD מighb תוצאות שקרובות יותר למינימום מאשר הגישה הסטוכסטית. אבל, מצד שני, הדבר עשוי להקשות על האלגוריתם לבסוף ממינימום לוקאלי.

- The following figure shows the paths taken by the three GD algorithms in parameter space during training



- They all end up near the minimum, but Batch GD's path actually stops at the minimum, while both Stochastic GD and Mini-batch GD continue to walk around
- However, Batch GD takes a lot of time to take each step

### בחירה האלפא

שיטת החיפוש : ננסה ערכים שונים של אלפא וນשתחמם בקבוצות 'hold out', עברו טסיטים ומדידת הטיעות (ביחס לדאטה שיש לנו), ונבחר אלפא שתنبي את המינימום. אם פונקציית העלות אינה עולה או אם היא נשארת אותו הדבר בכל שלב, אז האלגוריתם GD אינו עובד.

אבל צריך להיזהר לא לחשוף אלף שמותאיימה מיידי לדאטה שיש לנו, הרעיון הוא להשאיר קבוצה בצד שלא כללת ב-set training. מאוד קל למצוא את מהיר הדירה של דירה שנמצאת במאגר שלוי, אבל כדי למצוא את האופטימום הטוב ביותר ביוורר עברו דוגמיה חדשה נרצה להימנע מ- overfitting לדאטה הקבילים. לכן כדאי "לשיטים בצד" קבוצות אימון עברו מציאות האלפא הטובה ביותר.

אם פונקציית העלות גדלה בכל שלב של האלגוריתם, סביר כי אלף שנבחרה הינה גדולה מדי.

נבדוק את הטיעות בכל 100 איטרציות וגעזר כאשר הטיעות חולכת וגדלה.

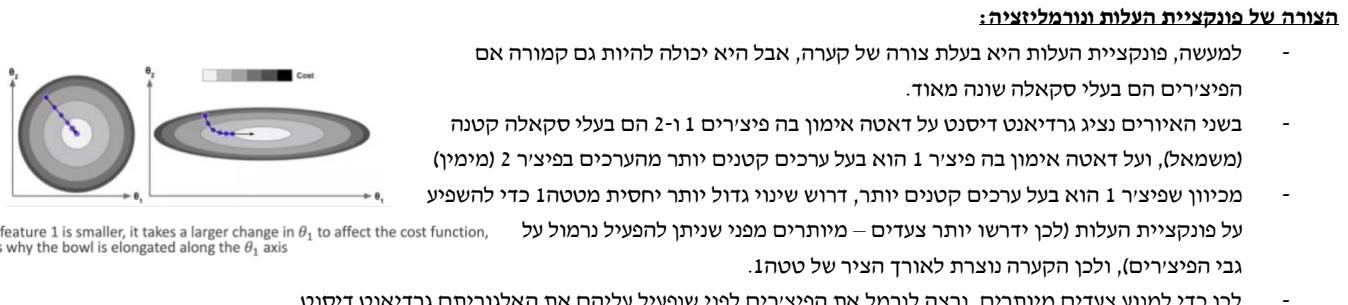
נבחר את האלפא בעלת העלות / השגיאה הפחותה ביותר.

שיטה נוספת הינה לערוך גרפ עבור ה- training error ולהתבונן על השיפוע שלו.

אם הגרפ של ה- training error עולה, אז (סביר כי) אלף גודלה מיידי. אם השיפוע של ה- training error אינו תלול מספיק, אז אלף קטן מדי.

**מה הוא תנאי העצירה?** כאמור בגרדיינט דיסנט ככל שנתקרוב למינימום הצדים שננקוט בהם ילכו וייהפכו קטנים יותר, לכן הסיכון שי"ניפול" בדיקוק על המינימום (גרדיינט ה-0) בזמן לא ארוך מיידי איןנו סביר ולכן נבדק שהירידה של פונקציית העלות באיטרציה הנוכחית מהאיטרציה הקודמת קטנה יותר מערך קטן כלשהו, אפסילון, למשל 0.003.

- מה קורה כאשר פונקציית העלות איננה נורמלית - כלומר אינגה קעורה:**
- לא כל פונקציות העלות נראות כמו קערות יפות.
  - יתרנו חורמים, ridges, plateaus, (חלקים שטוחים, בעלי שיפוע קרוב ל-0), ועוד כל מיני מאפיינים אי רגולריים, שהופכים את ההתקנסות למינימום להרבה יותר קשה.
  - הגרף המוצג מתאר את שי האתגרים העיקריים של גרדיאנט דיסנט :
    - \* אם האתחול הרנדומלי מציב את האלגוריתם בצד השמאלי, הוא יתכנס למינימום מקומי, שפחות אופטימלי מאשר האלגוריתם בימין, ייקח הזמן זמן לחצות את החלק ה"ירידוד" והכמעט \*
    - \* אם האתחול מציב את האלגוריתם בימין, ייקח הזמן זמן לחצות את החלק ה"ירידוד" והכמעט מישורי של הגרף, ואם נעצור מוקדם מדי לא נצליח להגיע למינימום הגלובלי.



- כיצד מנרמלים ? Feature Scaling**
- אחד השימושיים החשובים ביותר שיש להפעיל על הדאטה שלנו הוא feature scaling
  - עם כמה יוצאים מן הכלל, אלגוריתמי למידה חישובי אין בעלים ביצועים מוזרים כאשר הקלט הוא בעל ערכים מספריים בעלי סקללה שונה מאוד.
  - ישנן 2 דרכים להפוך את הערכים הללו לבעלי סקללה זהה : standardization ו- min-max scaling
  - ב – **min-max scaling** – הערכים משתנים ועוביים שונים בסקללה כך שהם נתחמים לטוויה  $-1 \text{ ל } 1$ . זה נעשה ע"י חישור הערך המינימלי וחילוק בערך המקסימלי מינוס המינימלי. מקופה טרנספורמר שנקרא MinMaxScaler בשם כך.
  - ב – **Standardization** – ראשית מחסרים את הממוצע/התוחלת (mean value) ואז מחלקים בסטיית התקן, כך שנקבל התפלגות בעלת תוחלת 0 ושותות יחידתית (1).
  - בשונה מ- $\text{min-max scaling}$ , standardization מנסה מצפה לערך אינפוט שהטוויה שלו הוא בין 0 ל-1).
  - אבל, standardization היא הרבה פחות מושפעת מ-outliers. (ערכים מחריגים)
  - למשל, נניח כי פיציר שהערך המרormal שלו הוא בטוחה שבין 0 ל-15, הוא בעל ערך מחירג שווה ל-100 (בטעות). במקרה זה min-max scaling תhapeוך את כל הערכים שוכנים בטוחה 0-15 ל-0-0.15, בעוד שטנדרטיזציה לא מושפעת מכך.
  - מקופה טרנספורמר שנקרא StandardScaler עבר סטנדרטיזציה.

## למידה חישובית ממכרז הרצאה 2

**עצי החלטה** הם תת-קבוצה של אלגוריתמים ממינימס - קלאסיפיקציה (Classifiers).

נקבל דאטא אימון = המון וקטורים של תכונות שלגביהם נדע האם הם פלוס או מינוס, נכניס אותם לאלגוריתם הלמידה שיניב עבורנו היפותזה, מודל, ממין (קלאסיפייר).

- Classification

- Given  $\{x_i, y_i\}$  where  $y_i \in \{0,1\}$  as training data, determine for a new  $x$  if  $x \in C_0$  or  $x \in C_1$

מושגים פורמליים (不服 מרחב נתונים (X) שייך ל $C_1$ ):

- X הוא מרחב הדגימות (X שייך ל $C_1$ ).

קונספט concept c הוא תת-קבוצה של מרחב הדגימות, כלומר c שייך לקבוצת החזקה של X = H. (נקרא גם מרחב ההיפותזה)

דאטא אימון (x, c) היא קבוצה של זוגות  $(x, c)$ , כך ש-x היא דוגמה ממרחב הדגימות X ו- $c$  שייך ל- $\{+1, -1\}$ .

אנחנו מחפשים אחר היפותזה (או מודל) h ששייך ל-H (תת-קבוצה של דגימות), כך ש- $(x, c) = (x, h)$  עברר כל x המופיע ב-D (או עבור רוב

ה- $x$ -ים בקבוצת האימון (D).

Convert from continuous to discrete can be done by quantization (binning):



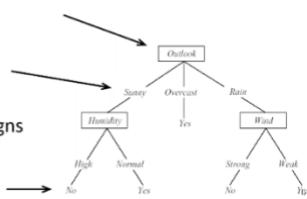
Convert from discrete to continuous can be done by encoding:

- Blue = 1, Green = 2, Brown = 3 etc.

יכולים להיות אטרייבוטים/פיצרים/נומרים = רציפים, וניתר להמיר בין משתנים רציפים לבדים ולהפך. אבל כדי שנעשה זאת בזיהירות רבה, מפני שאם נקבעו בשיטה בה הבדל בין 3 ל-2 גדול מההבדל שבין 3 ל-1 כנראה שהימרנו באופן לא נכון.

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

- Each internal node tests an attribute
- Each branch corresponds to an attribute value
- Each leaf node assigns a classification



השאלה שלנו השיעור: איך נבנה את העץ הזה באופן הטע/הטוב ביותר? איך יודעים איזו שאלה שואלים קודם?

### עצי החלטה

להלן דוגמה לגבי מר סמייט' שמחלית האם למכת לשחק טניס על פי מצב הרוח, הלחות, הטמפרטורה והתחזית (האטרייבוטים). להלן דוגמה לדגימות של מר סמייט' ועצי החלטה שנבנה מחדשתה.

### מתי משתמשים בעצי החלטה?

- כאשר הדגימות שלנו מתייחסות על ידי זוגות של תכונות וערכים.
- כאשר פונקציית המטריה שלנו היא בדידה (לא בהכרח, DT regression)
- כאשר ההיפותזה הנדרשת היא בעלת מבנה לוגי או אינטראקטיבית היא חשובה.
- כאשר יש לנו דאטא אימון שהוא possibly noisy, even inconsistent.

### יצירת העץ

נוקח את כל הדגימות וניצור מהם את השורש של העץ

נאתחול תור של הצמתים בעץ

כל עוד יש צמתים לא שלמים בתור נבצע:

- ניקח את הצמתה הבא ו-

- אם הדגימות ב- $T$  טהורות/מוניוכרומטיות (cols בצביע כחול למשל), נסמן את הצומת כשלם

ונמשיך לצומת הבא בתור.

- אחרת, נכנס ל-A את האטרייבוט (ערוך החסימה של ערבי) ה-"טוב ביותר" עבור הקבוצה ב- $T$ .

-- נשים את A כאטרייבוט ההחלטה עבור T

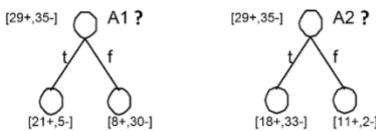
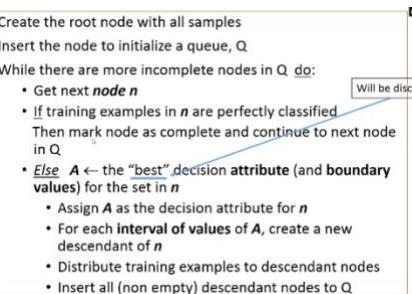
-- לכל אינטראול של ערכים של A, ניצור צומת-בן לצומת מ- $T$

-- נציג את הדגימות לילדים של T

-- נכנס את כל הילדים ללא ריקים לתור

### איך נדע מייהו אטרייבוט ההחלטה הטוב ביותר?

בדוגמה ממשאל יש לנו שני אטרייבוטים A1, A2. איך נדע איזה עדיף? נשים לב שהילדים של האטרייבוט A2 מניב ילדים-מעט-מוניוכרומטיים, מה שמקרב אותנו להתפלגות טהורה, מה שנanton לנחרווים יותר. A1 אינפורמציה. לכן, נרצה להפחית את דמות האי-טהורות (או הבילבול, או היוזאות) בצד הבא שלנו.



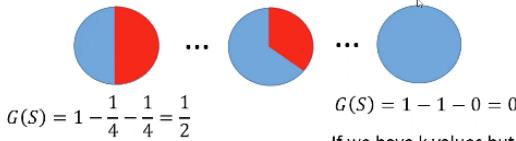
**פונקציית אי וודאות** היא פונקציה המתקבלת הטעוגות דיסקרטית ומחזירה מספר ממשי, המקיים את התנאים הבאים עבור וקטור הסתברויות:

for probability distributions  $P = (p_1, \dots, p_k) \in [0,1]^k$ :

- $\varphi(P) \geq 0$
- The minimal value is attained when  $\exists i$  s.t  $p_i = 1$ .
- The maximal value is attained when  $1 \leq \forall i \leq k, p_i = \frac{1}{k}$ .
- It is symmetric with respect to the components of  $P$
- It is smooth (infinitely differentiable) in the relevant range



מן בטווח ערכים 0-1, כך שערכים נומכרים משמשותם אי וודאות גבוהה.



If we have  $k$  values equi-distributed then  
 $G(S)=1- k(1/k)^2=1-1/k=\sim 1$

If we have  $k$  values but only one value really present then  
 $G(S)=1-(k/k)^2 = 0$

### פונקציית Gini Impurity

מודד חסר שוין בכלכלה

-

מניב טווח ערכים 0-1, כך שערכים נומכרים משמשותם אי וודאות גבוהה.

$$G(S) = 1 - \sum_{i=1}^c (p_i)^2 = 1 - \sum_{i=1}^c \left(\frac{|S_i|}{|S|}\right)^2$$

(זכור שפונקציה זו מקבלת  $S$  ממשים שהיא מקבלת את קבוצה של דגימות שנמצאים באותו החומר)

להלן דוגמה לחישוב אי וודאות על ידי פונקציית Gini: כאשר יש לנו שני צבעים  $c=2$

בזרה לנשא השיעור: אנו מחפשים אחר אטריבואט  $A$  שיניב את ממוצע משקלל הנקוט  $Gini$  הtout לאחר פריצול, וכן נגידר את פונקציית הרוחות, שמחשבת לנו את **הוודאות שהרווחנו מהפיצול על ידי אטריבואט ספציפי**,  $A$ , על לת הקבוצה של הדגימות  $S$  של הצומת הנוכחי:

$$\begin{aligned} GiniGain(S, A) &= \Delta G(S, A) \equiv G(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} G(S_v) \\ &\text{↑ Change in Impurity} \\ &\text{↑ Impurity Before Split} \\ &\text{↑ Weighted Average of Impurity of All Groups After Splitting} \end{aligned}$$

נוסחה זו ממחשבת כמה אי וודאות הייתה לי לפני הפריצול ומהזירה כמה אי וודאות משוקלתת הרוחותי אחריו הפריצול על ידי האטריבואט. אנחנו נרצה למקסם את

**פונקציית ה-Gain.** ( $Gain(S, A) = \text{כל מי שבתוך } S \text{ ועונה לערך של } v$ )

### פונקציית האנתרופופיה Entropy

נשתמש בנוסחה של אנתרופופיה כפונקציית אי וודאות עברו משתנה רנדומית  $X$  שולקן  $c$  ערכים שונים עם הסתברויות  $p_i$ :

$$H(X) = -\sum_{i=1}^n p_i \log(p_i)$$

(לוג בסיס 2, כאשר ידוע לנו כי לוג על ערכים בין 0-1 הוא שלילי ולכן המינוס)

מודד את המידע הממוצע שמתקשר לתוצאה של משתנה מסוים.

יהו קבוצה  $S$  וدادה עם  $c$  (יתכן יותר מ-2) קלאסים.  $P_i$  היא החלק היחסית של קלאס  $i$  בקבוצה  $S$ . האנתרופופיה של  $S$ :

נמיר את האנתרופופיה לנוסחת רוח (Gain(S,A)) = פונקציית הרוח הצפוי של האנתרופופיה לאחר הפריצול על ידי אטריבואט  $A$ .

**תכונות גנטופות של אנתרופופיה:**

הערך המקסימלי שלו (בזה יש מקסימים אי וודאות), הינה כאשר הטעוגות

שנמשבגה בלחמה 1. פונקציית לוג היא פונקציה "עוברת" קעורה כלפי מטה (ונזרת שונית של הינה שלילית).

Lemma 2 (Jensen's Inequality):

The following holds for any sad function  $f$ :

$\forall x_1, \dots, x_k$  and  $\forall \lambda_1, \dots, \lambda_k \in [0,1]$  s.t  $\sum \lambda_i = 1$ .

$$f\left(\sum_{i=1}^k \lambda_i x_i\right) \geq \sum_{i=1}^k \lambda_i f(x_i).$$

$$H\left(P = \left(\frac{1}{k}, \dots, \frac{1}{k}\right)\right) = \log k.$$

יוניפורמיות/אחדה כיאה לפונקציית אי וודאות:

**נרצה להוכיח איפה אנטרופיה מקסימלית:**

נשתמש בלחמה 1. פונקציית לוג היא פונקציה "עוברת" קעורה כלפי מטה (ונזרת שונית של הינה שלילית).

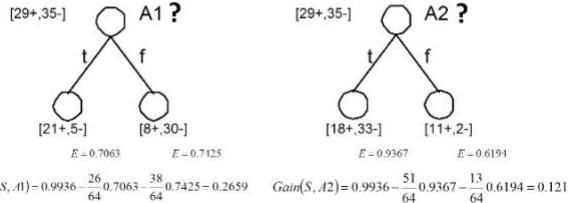
ובלחמה 2: אי שוויון ינסן (מושיע משMAIL<>) אשר הוא נכון לכל פונקציה עצובה.

$$\begin{aligned}
H(p_1, \dots, p_k) &= -\sum_{i=1}^k p_i \log(p_i) \\
&= \sum_{i=1}^k p_i \log\left(\frac{1}{p_i}\right) \\
&\leq \log\left(\sum_{i=1}^k p_i \cdot \frac{1}{p_i}\right) \\
&= \log k
\end{aligned}$$

- הצעד האחרון בהוכחה:
- השתמשנו בשתי הلمות לעיל במטרה להראות היכן אנטרופיה מקסימלית.
  - בחרזה לדוגמה שלנו: נחשב את הרוח האנטרופי עבור שני האטראיביטים **A1, A2** כך נדע איזה מבין האטראיביטים עדיף, נשים לב שהרווח גובה יותר עבור A1 כלומר הוא מניב עבורנו או וודאות פחות מאטראיביט A2.

Which attribute is best?

$$\text{Entropy}([29+, 35-]) = -\frac{29}{64} \log_2\left(\frac{29}{64}\right) - \frac{35}{64} \log_2\left(\frac{35}{64}\right) = 0.9936$$

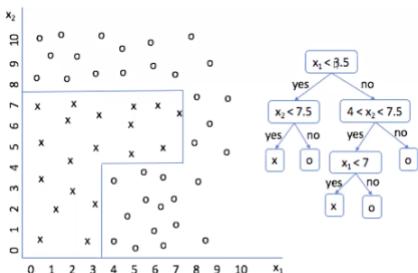


The information gain values for the 4 attributes are:

- Gain(S, Outlook) = 0.247
- Gain(S, Humidity) = 0.151
- Gain(S, Wind) = 0.048
- Gain(S, Temperature) = 0.029

where S denotes the collection of all training examples

## DTs in continuous space

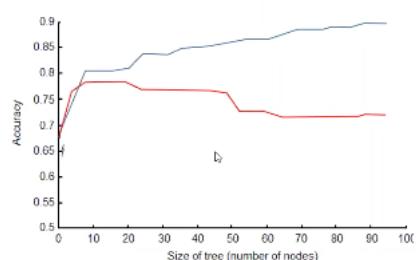


**תמיד נרצה להימנע מ-Overfitting!** ! לנוכח להרחב את העץ יותר מדי כדי שלא יתאים לדatos הנתון יותר מדי ויכול להגיע להחלטה עם מצב חדש. כדי לעשות זאת משתמש בdataset סט נוסף שיקרא "validation". נלמד את הדאטה על ידי הדאטה אימון אבל מדוד את הטעות על ה- validation set (להלן גוף המתאר את הדיווק), וכך יותר ככל שיש לנו יותר אטראיביטים מכאן שנרצה להפסיק להרחיב את העץ לפני שנגיע למונוכרומטיות מושלמת (לפניהם שגיאה 0), או להרחיב את העץ לשגיאה 0 ואז לבצע קיצוץ post-prune, או לשלב בין שתי השיטות.

**גישה Chi Square** ששותאלת – האם פיצול לפי אטראיביט מועמד לנו התפלגות לפי

כלאש שיש לה יותר כוח מהונכית? האם הבנים מאוד דומים לאבא? הם הרוחם הוא

סטטיסטיות משמעותי? (הרחבה בתרגול 2)



**בעיות נוספת**

- התמודדות עם **ערכים חסרים** של חלק מהתatribוטים : השלמת דата או ליקיחת הממוצע
- חיפוש אחר ערך חותך, נקודת הסף, **אטatribוטים רציפים**
- פיצול אינפרמציה ו- **Gain Ratio** עבור אטatribוטים עם ערכים רבים - מבחינת החישוב יש יתרון עבור אטatribוטים שיש להם מספר ערכים גבואה יותר וכן נגידר Gain Ratio (הרחבה בתירגול)
- הצללת העלות עבור שמירת האטatribוטים, ופונקציית שגיאה ממושקלת
- גבולות מורכבים

## למידה חישובית ממדיע: תרגול 2

**חזרה קצרה על מה שנלמד בתרגול הקודם:**

- גרדיאנט - וקטור הנגורות החלקיות

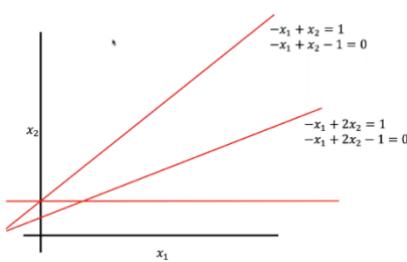
$$\nabla f = \left( \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right) \quad f(x_1, x_2, \dots, x_n)$$

הגרדיינט עבורה יהיה:

הגרדיינט מצביע לכיוון העלייה התולדה/הגדולה ביותר ביחס לנקודה בה הוא מחושב, וערכו הוא השיפוע בכיוון זה. כך שם נלך נגד כיוון הגרדיינט, נתקדם לכיוון המינימום (גרדיינט דיסנט).

בנוסף לדברנו על הצורך ב-learning rate, לפחות, במקרה מסוים (משקל אופטימל). העלא את הדיוון הבא: מצד אחד לא נרצה אף גודלה מדי כדי שלא נגיע למצב בו "דילגנו" על פני המינימום, וכן מצד שני, לא נרצה אף קטנה מדי כדי להגיע למינימום בזמן סביר.

**תזכורת באלגברה:**



מפרידلينארוי הוא למשהו היפר-מיישור במרחב שמודרך על ידי הווקטור טטה, וכל הנקודות על ההיפר-מיישור הניל פותרות את המשוואה  $\theta_1 x_1 + \dots + \theta_n x_n = b$  ( $= \theta_0$ ) כאשר  $x$  הן הקואורדינטות של הנקודת.

ההיפר-מיישור מפריד את המרחב לשני חללים, כל הנקודות שהמשוואה פותרת הן מעל  $b$ . הטעות שולטות בזווית של המישור.

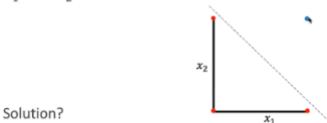
נרצה למצאו מפרידلينארוי: כל נקודה עלינה עם תוצאה שגדולה מ-0, תהיה שייכת למחלקה + או (-). כל נקודת תחתונה עם תוצאה נמוכה מ-0, תהיה שייכת למחלקה - (או +).

לכן עלינו למצוא את הווקטור טטה (שייש בו +1 מערבים, 0 משקלים וביאס טטה 0).

$$\text{נמצא } 1 \text{ אם } \sum_{i=1}^n \theta_i x_i + \theta_0 > 0 \text{ ואחרת } -1.$$

**להלן דוגמאות:**

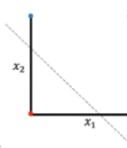
•  $X_1$  AND  $X_2$



• Solution?

- If  $1 \times X_1 + 1 \times X_2 - 1.5 > 0$  predict 1
- Otherwise predict -1.
- i.e.  $\theta_0 = -1.5, \theta_1 = 1, \theta_2 = 1$

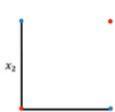
•  $X_1$  OR  $X_2$



• Solution?

- $X_1 + X_2 - 0.5 > 0$  predict 1
- Otherwise -1

•  $X_1$  XOR  $X_2$



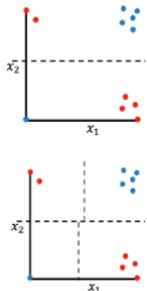
- Solution?
- There is no solution
- Many functions cannot be represented using a linear separator, i.e., they are not linearly separable

העניין הוא שלא כל DATA ניתן להפרדהリンארית (להלן בעיית הקסוך).

ולכן נצטרך כלים קצץ יותר חזקים ממפרידיםリンאריים (נרחיב עליהם בהמשך).

**עצי החלטה**

**האינטואיציה של עצי החלטה:** האטריבוטים שיבחרו, יבחרו על פי מי שAKER אוטנו ביותר להפרדה מלאה בילדים. נרצה למצוא חלוקה לבנים, כך שלאחר החלוקה נהיה כמו שיוצר קרוביים לטהורות לבנים.



• עליה הוא תשובה לקלסיפיקציה

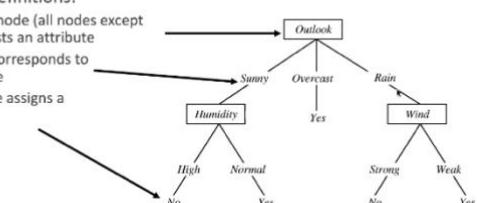
• כל ענף מייצג ערך של האטריבוט הנבחר

• כל צומת פנימי בוחנים אטריבוט.

• יודע לטפל גם באטריבוטים רציפים, עבורם נבחר threshold במטרה לחלק לאטריבוטים.

• Decision Tree definitions:

- Each internal node (all nodes except the leaves) tests an attribute
- Each branch corresponds to an attribute value
- Each leaf node assigns a classification



**האלגוריתם לבניית העץ:** (דוגמא לעץ שנבנה בתור, יש אפשרות לבנות עץ גם ברקורסיה)

- כל עוד יש צומחות בתור נבצע :
- קח את הצומחות הבא בתור.
- אם הדגימות בצומת זה טהורים (שייכים כולם לאותו הקלאס) המשך לצומת הבא
- אחרת : נכניס ל- A את האטריביוט הטוב ביותר קבוצת הדגימות בצומת מ, נמנה את A לאטריביוט הבחירה של הצומת מ, ולכל ערך של האטריביוט A ניצור בן חדש לצומת מ. נפלג את הדגימות של צומת מ לבנים חדשים שלו ונכנס את הצמתים הבנים לתור.
- כשאינו יותר לבנים בתור נסימן את הלולאה

איך בוחרים את האטריביוט הטוב ביותר שיקר卜 אותנו כמה שיותר לטהורות/וזאות? כדי לענות על שאלה זו וראשת עליינו לבחור ממד שיגיד לנו האם הפיצול הוא אכן טוב. עלינו למודוד כמה אנחנו רוחקים מטוהרויות (perfect classification), מדייה זו נקבעת impurity (מחושב על צומת):

- Impurity / אי טהורות גבולה ממשועורה – אנחנו רוחקים מקלסיפיקציה מושלמת (כל שיותר קרוביים להתפלגות יוניפורמיית נתרחק יותר מקלסיפיקציה מושלמת)
- Impurity / אי טהורות נוכחה ממשועורה – אנחנו קרוביים לקלסיפיקציה מושלמת (בהערכתה מופיעה הגדרה פורמלית יותר)

#### בחירה האטריביוט הטוב ביותר

- נחשב את רמת הטהורות Impurity עבור הצומת הנוכחי (בו אנו נמצאים)
- נחשב את הממוצע המשוקל של רמת הטהורות על הצמתים-הבנים לאחר פיצול לפי האטריביוט שנבדק
- נחסיר את השינוי(הממוצע המשוקל) מהראשון(רמת הטהורות של האבא) ונקבל את הפרש ה- impurity
- נבחר את האטריביוט שמניב את הפרש ה- Impurity המקסימלי, הגבוהה ביותר
- הפרש ה- **Impurity** בין רמת הטהורות של צומת האב לבין ממוצע רמת הטהורות שמתקבלת עבור הבנים לאחר הפיצול נקבעת **.Goodness of split**

הנוסחה שמחשבת את הפרש-*Impurity* מקבלת שני אינפוטים ; את האטריביוט הנבדק A ותת קבוצת הדגימות אשר בצומת הנוכחי :

$$\Delta\varphi(S, A) = \varphi(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} \varphi(S_v)$$

\* Where  $\varphi$  is the impurity measure

רמת הטהורות בצומת האב פוחתת הממושך הממושך של הטהורות בבנים הנוצרים מהפיצול על פי A (יש לשים לב שהאימפיאורייטי מחושב על הקלאס ואינו שום קשר לערכי האטריביוט כזוה מגע לחישוב האימפיאורייטי בלבד הפיצול לבנים על פי האטריביוט).

**עובדות חשובות :** (האות היונית פ' מסמלת את-*impurity*)

- ממד אי הטהורות (impurity measure) מודד לפי התפלגות הקלאסים בצומת כלשהו
- דוגמאות מפצלות לפי ערכי האטריביוט הנבדק – A
- זאת אומרת שאנו מפצלים את הדגימות לפי ערכי האטריביוט ואז מחשבים את רמת הטהורות לפי ערכי הקלאסים.

#### טיב הפיצול / Goodness of Split

ישנם 2 יישומים עיקריים עבור קритריון הטהורות :

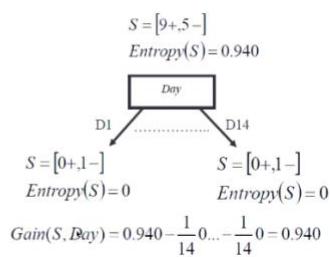
	Gini	Entropy
Impurity	$GiniIndex(S) = 1 - \sum_{i=1}^c \left( \frac{ S_i }{ S } \right)^2$	$Entropy(S) = - \sum_{i=1}^c \frac{ S_i }{ S } \log \frac{ S_i }{ S }$
Goodness of split	$Gini Gain = GiniIndex(S) - \sum_{v \in Values(A)} \frac{ S_v }{ S } GiniIndex(S_v)$	$Information Gain = Entropy(S) - \sum_{v \in Values(A)} \frac{ S_v }{ S } Entropy(S_v)$

- מסמל את מספר הקלאסים C
- הערך המקסימלי שגיאני אינדקס יכול לקבל הוא 1, ככל שכמות הקלאסים עולה הסכום שואף ל-0. כאשר יש שני קלאסים המקסימים שנייני יקבל יהיה 0.5
- הערך המינימלי של אנטרופיה עבור 2 קלאסים הוא 1, עבור 4 קלאסים הוא 2. מה שניתן להבין מכאן הוא שהוא אנטרופי לא חסום.
- ערכם המינימלי הוא 0.
- את שתי הנוסחאות נמיר לפורמוללה של גודנס אוֹס ספליט ובוחן נשתמש (שורה שנייה בטבלה).

## התמודדות עם אטריביזיט עם ערכים רבים

- Imagine using the attribute DAY=[D1,...,D14]

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	Yes
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Overcast	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No



בහינתו DATAה של ציונים בקורס, יש לנו 100 דוגמאות של סטודנטים עם 5 אטריביזיטים : תעודות זהות, מין, ממוצע בוגרות, שעות למידה, מזל (1-10). נרצה לחזות מי עבר את המבחן ומילא.

באיזה אטריביזיט יבחר InformationGain ולחזות? נשים לב שבחירה האטריביזיט של תעודה הזהות מפני שהוא בעל 100 ערכים שונים, הוא ענייך לך תשובה לכל סטודנט. יש לנו תשובה ייחודית והוא יוריד את האי וודאות ב-100%. אבל כמובן שהוא לא טוב, ובאופן כללי נרצה להכנס DATA חדשה ולהזות עליה, מה שלא ניתן עבור ת"ז שלא נמצא במינימום.

יש נטייה באלגוריתם של גודנס או ספליט לבחיר אטריביזיט בעל ערכים רבים.

**הדרך להימנע מנטייה זו היא להשתמש בו GainRatio :**

נוקח את ה-InformationGain ו"נורמל" אותו על ידי חילוק ב-SplitInformation(S,A)

$$SplitInformation(S, A) = -\sum_{a \in A} \frac{|S_a|}{|S|} \log \frac{|S_a|}{|S|}$$

כאשר SplitInformation(S,A) הוא האנתרופיה ביחס לאטריביזיט A

עד כה חישבנו אנטרופיה עבור **קלאס**, כתע נחשב אנטרופיה **בהתיחס לאטריביזיט**. ככל שייהיו לנו יותר ערכים באטריביזיט ככה נגיעה למקרה שהוא יותר יוניפורמי. (ככה "מענים" אטריביזיט בעל ערכים רבים)

דוגמה לחישוב ספליט אינפורמיישן וגיאון רשי עבור הדוגמה המוצגת לעיל בתמונה, ועבור האטריביזיט Day.

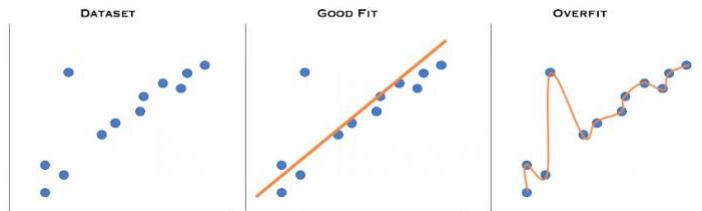
$$SplitInformation(S, Day) = -\sum_{i=1}^{14} \frac{1}{14} \log \frac{1}{14} = -\log \frac{1}{14} = 3.8074$$

$$GainRatio(S, A) = \frac{0.94}{3.8074} = 0.2469$$

## OVERFITTING / OVERGENERALIZATION

בשלב מסוים באלגוריתם הלמידה כל שהופכים את המודול שלנו למורכב יותר, כך אנחנו מתאימים יותר את המודול שלנו לדאטתנו. אנחנו נרצה להגיע למודול כמה שייותר כדי כך שכאשר תגעה דוגימה חדשה, המודול ידע להתאים עבורה תוצאה נכונה ביותר.

**דוגמה לאוברפיטינג בדוגשיה:**



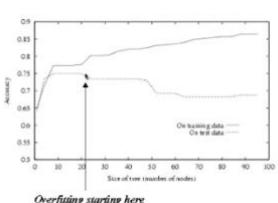
**עצי החלטה נוטים ל-overfit.** המשמעות של כך היא שהעצים שלנו עושים יותר ויותר ספציפיים עבור ה-training data.

הדרך שלנו להתמודד עם נטייה זו נקראת Pruning.

### Pruning

בגישה זו עליינו לחקצ ענפים מהעץ על מנת להפוך אותו לפחות לטוב יותר, ובכך לקבל test error טוב יותר.

אפשר להחליט מראש על גובה עץ מסוימלי לפי ניסוי וטעייה. נסמן גבהים שונים ונחשב עבור כל גובה את ה-validation accuracy



אפשריה נוספת נקראת **post pruning** : נעלם מהעץ מהעלים לכיוון השורש, עברו כל צומת נחליט האם להשאיר אותו או לחקוץ.

אפשריה נוספת נקראת **Chi Square Test** : Chi Square Test

אפשר גם לחקוץ בזמן בניית העץ = לא ניצור עד צמתים בנים.

## Chi Square Test

זהוי בדיקה סטטיסטית שמטרתה להגיד לנו האם היפוכו של מבחן עבורנו התפלגות שהיא רנדומלית לחולטן או האם יש לה כוח חיזוי כלשהו. לכן, נבדוק אם פיצול לפי האטריביוט הנבחר מוביל עבורנו התפלגות שהיא ממש רנדומלית. אם יש לי בצדומת 100 אינסטנסים, אם חלק אותה רנדומלית, נצפה שהפיזול הרנדומלי יניב בערך את אותו היחס שהוא באב. האם הילדים שומרים על היחס בין הקלאסים שיש באבא – אם כן, נגוזם, לא הרוחנו וודאות! ככל שאחננו יותר ורוחקים ממה שציפינו = החלוקה היא יותר רנדומית. אם אני רוחק מרנדום, נבע את החלוקה. בדיקה זו מבצעים תוך כדי הבניה של העץ – בודקים את כוח החיזוי.

### הבדיקה מבוצעת באופן הבא:

עבור כל ילד נחשב ונסכום – כמה דגימות יש לנו בקלאס 0 פחות כמות הדגימות שנפוצה לקבל בקלאס 0 בטיבו, חלק לצפי הדגימות בקלאס 0. ווד מספר הדגימות בקלאס 1 פחות כמות הדגימות שציפינו שנתקבל בקלאס 1 בטיבו, חלק לצפי הדגימות בקלאס 1. (חלוקת היא סתם נרמול)  
המוחות היא כמה אני קרוב לצפי שלי  
נקבל מספר כלשהו, ונשאלו את עצמנו מה זה אומר? בשביל כך יש לנו את טבלת ה-Chi Square-test.

- The test itself (assume Y can only take values of 0 \ 1):

- $P(Y = 0) \approx \frac{\# Y=0 instances}{\# Instances}$
- Call  $D_f = \text{number of instances where } x_j = f$
- $p_f = \text{number of instances where } x_j = f \ & \ Y = 0$
- $n_f = \text{number of instances where } x_j = f \ & \ Y = 1$
- $E_0 = D_f * P(Y = 0), E_1 = D_f * P(Y = 1)$
- So Chi Square statistic is:

$$\chi^2 = \sum_{f \in \text{values}(x_j)} \frac{(p_f - E_0)^2}{E_0} + \frac{(n_f - E_1)^2}{E_1}$$

## Chi Square-table

מה שמשמעותו הם המספרים בתוך הטבלה.

.Degree of rhythm

**DegreeOfRhythm = (numOfClasses – 1)(numOfValsInA – 1)**

בכל צומות שאנו נמצאים יש כמות אחרת של ערכים לאטריביוט הנבחר. כולם יתכן שלאטריביוט "תיזיות" יש 4 אפשרויות (מעון, מעון חלקית,شمיש, גשם), אבל בזומת הנוכחי שלנו אין אפשרות ל"תיזיות" שימוש. לכן  $\text{DegreeOfRhythm} = 3$ . מפני שזה מספר הערכים האפשריים של האטריביוט בזומת הנוכחי.

## מהו האלפא רиск ?Alpha Risk

אני רוצה להיות רוחק מרנדום

בהת恭נות מסוימת (השגיאה שאני

מוכנה לסייע למשל

0.05

**Table of Probabilities for the Chi-Squared Distribution**

Alpha Risk														
df	0.995	0.990	0.975	0.95	0.9	0.75	0.5	0.25	0.1	0.05	0.25	0.01	0.005	0.001
1	0.000039	0.000157	0.000982	0.00393	0.0158	0.102	0.455	1.323	2.706	3.841	4.323	6.635	7.879	10.828
2	0.010	0.020	0.051	0.103	0.211	0.575	1.386	2.773	4.605	5.991	2.773	9.210	10.597	13.816
3	0.072	0.115	0.216	0.352	0.584	1.213	2.366	4.108	6.251	7.815	4.108	11.345	12.838	16.266
4	0.207	0.297	0.484	0.711	1.064	1.923	3.357	5.385	7.729	9.488	5.385	13.277	14.860	18.467
5	0.412	0.554	0.831	1.145	1.610	2.675	4.351	6.626	9.236	11.070	6.626	15.086	16.750	20.515
6	0.676	0.872	1.237	1.635	2.204	3.455	5.348	7.841	10.645	12.592	7.841	16.812	18.548	22.458
7	0.989	1.239	1.690	2.167	2.833	4.255	6.346	9.037	12.017	14.067	9.037	18.475	20.278	24.322
8	1.344	1.646	2.180	2.733	3.490	5.071	7.344	10.219	13.362	15.507	10.219	20.090	21.955	26.124
9	1.735	2.084	2.700	3.325	4.168	5.899	8.343	11.389	14.684	16.919	11.389	21.666	23.582	27.877
10	2.156	2.558	3.247	3.940	4.865	6.737	9.342	12.549	15.987	18.307	12.549	23.209	25.188	29.588
11	2.603	3.053	3.816	4.575	5.578	7.584	10.341	13.701	17.275	19.675	13.701	24.723	26.757	31.264
12	3.074	3.571	4.404	5.226	6.304	8.438	11.340	14.845	18.549	21.026	14.845	26.217	28.300	32.909
13	3.565	4.107	5.009	5.892	7.042	9.299	12.340	15.984	19.812	22.362	15.984	27.688	29.819	34.528
14	4.075	4.664	5.629	6.571	7.790	10.165	13.339	17.117	21.064	23.685	17.117	29.141	31.319	36.123
15	4.601	5.229	6.262	7.261	8.547	11.037	14.339	18.245	22.307	24.998	18.245	30.578	32.801	37.697
16	5.142	5.812	6.908	7.962	9.312	11.912	15.338	19.369	23.542	26.296	19.369	32.000	34.267	39.252
17	5.697	6.408	7.564	8.672	10.085	12.792	16.338	20.489	24.769	27.587	20.489	33.409	35.718	40.790
18	6.265	7.015	8.231	9.390	10.865	13.675	17.338	21.605	25.989	28.869	21.605	34.805	37.156	42.312
19	6.844	7.633	8.907	10.117	11.651	14.564	18.338	22.718	27.204	30.144	22.718	36.191	38.582	43.820
20	7.434	8.260	9.591	10.851	12.443	15.452	19.337	23.828	28.412	31.410	23.828	37.566	39.997	45.315
21	8.034	8.897	10.283	11.591	13.240	16.344	20.337	24.935	29.615	32.671	24.935	38.932	41.401	46.797
22	8.643	9.542	10.982	12.338	14.041	17.240	21.337	26.039	30.813	33.924	26.039	40.289	42.796	48.268
23	9.260	10.196	11.689	13.091	14.848	18.137	22.337	27.141	32.007	35.172	27.141	41.638	44.181	49.728
24	9.886	10.856	12.401	13.848	15.659	19.037	23.337	28.241	33.196	36.415	28.241	42.980	45.559	51.179
25	10.520	11.524	13.120	14.611	16.473	19.939	24.337	29.339	34.382	37.652	29.339	44.314	46.928	52.620
26	11.160	12.198	13.844	15.379	17.292	20.843	25.336	30.435	35.563	38.885	30.435	45.642	48.290	54.052
27	11.808	12.879	14.573	16.151	18.114	21.749	26.336	31.528	36.741	40.113	31.528	46.963	49.645	55.476
28	12.461	13.565	15.306	16.928	18.939	22.657	27.336	32.620	37.916	41.337	32.620	48.278	50.993	56.892
29	13.121	14.256	16.047	17.708	19.768	23.567	28.336	33.711	39.087	42.557	33.711	49.588	52.336	58.301
30	13.787	14.953	16.791	18.493	20.599	24.478	29.336	34.800	40.256	43.773	34.800	50.892	53.672	59.703
40	20.707	22.164	24.433	26.509	29.051	33.660	39.335	45.616	51.805	55.758	45.616	63.691	66.766	73.402
50	27.991	29.707	32.357	34.764	37.689	42.942	49.335	56.334	63.167	67.505	56.334	76.154	79.490	86.661
60	35.534	37.485	40.482	43.188	46.459	52.294	59.335	66.981	74.397	79.082	66.981	88.379	91.952	99.607
70	43.275	45.442	48.756	51.739	55.329	61.694	69.334	77.577	85.527	90.531	77.577	100.425	104.215	112.317
80	51.172	53.540	57.153	60.391	64.278	71.145	79.334	88.130	96.578	101.879	88.130	112.329	116.321	124.839
90	59.196	61.754	65.647	69.126	73.291	80.625	89.334	98.650	107.565	113.145	98.650	124.116	128.299	137.208
100	67.328	70.065	74.222	77.929	82.358	90.133	99.334	109.141	118.498	124.342	109.141	135.807	140.169	149.449

### דוגמה לחישוב האנטרופיה ולאחר מכן הוכנה לנוסחת ה-Information Gain

- What calculations are needed to find the feature to split the root of the decision tree using Information Gain

- Reminder:

$$\text{Information\_Gain} = \text{Entropy}(S) - \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

$$\text{Entropy}(S) = - \sum_{i=1}^c \frac{|S_i|}{|S|} \log \frac{|S_i|}{|S|}$$

- $c$  – number of classes
- $\text{Values}(A)$  – all the values in the A feature

- We need to calculate:

- Entropy(root)
- Weighted average of the Entropy according to "Attraction"
- Weighted average of the Entropy according to "Weather"

Instance	Attraction	Weather	Classification
1	Swim	Hot	-
2	Dance	Hot	+
3	Casino	Hot	+
4	Golf	Hot	-
5	Swim	Mild	-
6	Casino	Mild	-
7	Dance	Mild	+
8	Golf	Mild	-
9	Ski	Mild	+
10	Ski	Cold	+
11	Casino	Cold	-
12	Dance	Cold	-

- Entropy(root)

$$\text{Entropy}(root) = - \left( \frac{7}{12} \log \frac{7}{12} + \frac{5}{12} \log \frac{5}{12} \right)$$

- Weighted average of the Entropy according to "Attraction"

$$\sum_{v \in \text{Values}(Attraction)} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

$$= - \left( \frac{2}{12} \left( \frac{2}{2} \log \frac{2}{2} \right) + \frac{3}{12} \left( \frac{2}{3} \log \frac{2}{3} + \frac{1}{3} \log \frac{1}{3} \right) + \frac{3}{12} \left( \frac{1}{3} \log \frac{1}{3} + \frac{2}{3} \log \frac{2}{3} \right) + \frac{2}{12} \left( \frac{2}{2} \log \frac{2}{2} \right) \right. \\ \left. + \frac{2}{12} \left( \frac{2}{2} \log \frac{2}{2} \right) \right)$$

- Weighted average of the Entropy according to "Weather"

$$\sum_{v \in \text{Values}(Weather)} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

$$= - \left( \frac{4}{12} \left( \frac{2}{4} \log \frac{2}{4} + \frac{2}{4} \log \frac{2}{4} \right) + \frac{5}{12} \left( \frac{2}{5} \log \frac{2}{5} + \frac{3}{5} \log \frac{3}{5} \right) + \frac{3}{12} \left( \frac{1}{3} \log \frac{1}{3} + \frac{2}{3} \log \frac{2}{3} \right) \right)$$

Put it all together in the Information Gain formula

$$\text{Information Gain}(root, \text{Attraction})$$

$$= - \left( \frac{7}{12} \log \frac{7}{12} + \frac{5}{12} \log \frac{5}{12} \right)$$

$$+ \left( \frac{2}{12} \left( \frac{2}{2} \log \frac{2}{2} \right) + \frac{3}{12} \left( \frac{2}{3} \log \frac{2}{3} + \frac{1}{3} \log \frac{1}{3} \right) + \frac{3}{12} \left( \frac{1}{3} \log \frac{1}{3} + \frac{2}{3} \log \frac{2}{3} \right) \right.$$

$$\left. + \frac{2}{12} \left( \frac{2}{2} \log \frac{2}{2} \right) + \frac{2}{12} \left( \frac{2}{2} \log \frac{2}{2} \right) \right) = 0.36$$

$\text{Information Gain}(root, \text{Weather})$

$$= - \left( \frac{7}{12} \log \frac{7}{12} + \frac{5}{12} \log \frac{5}{12} \right)$$

$$+ \left( \frac{4}{12} \left( \frac{2}{4} \log \frac{2}{4} + \frac{2}{4} \log \frac{2}{4} \right) + \frac{5}{12} \left( \frac{2}{5} \log \frac{2}{5} + \frac{3}{5} \log \frac{3}{5} \right) \right.$$

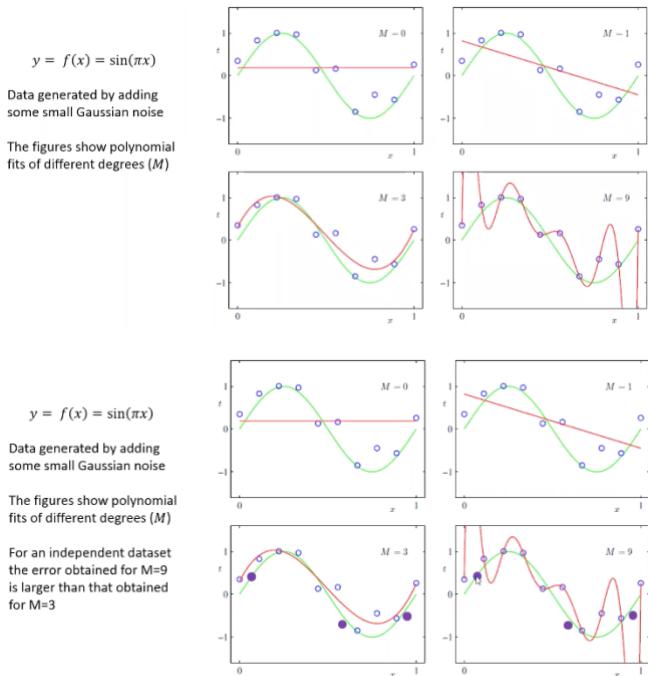
$$\left. + \frac{3}{12} \left( \frac{1}{3} \log \frac{1}{3} + \frac{2}{3} \log \frac{2}{3} \right) \right) = 0.0085$$

## למידה חישובית מפדיין: הרצאה 3 – פולינומיאליות וOVERFITTING

### Introduction to Overfitting

- **טעות השערוך:** מודד את הטעות בין הפרדיקציה לבין הערך האמיתי. מגדנו טעות זו עד כה על הדאטה אימון.
- נקראות גם טעות האימון.

- **טעות הכללה:** מודד את הטעות על דאטה חדש, לא על הדאטה אימון. בלמידה חישובית אנו מעוניינים בטעות הכללה.
- **אובייפיטיניג ברגression פולינומיאלית:**



- נלקחה הפונקציה סינוס(פאאי'איקס) ונדרגו ממנה 10 נקודות עליהן חושבה הפונקציה, אלה הן הנקודות החולות, אולם זה לא ערכן המדויק, והוסףנו "רעש" גאוסיאני (שמפלג נורמלית) לערך. אם לא היינו מוסיפים רעש הדגימות החולות היו נמצאות על הפונקציה הירוקה (זו הפונקציה שאכן יצרה את הדאטה עבורי).
- בעולם האמיתי אנחנו לא יודעים איך נראהית הפונקציה שאנו צריכים למצוא – הגרף הירוק הוא לא כorrect בalthי נראאה.
- ראשית התחלנו בניסוי לחזות את הנקודות על ידי פונקציה ממולחה 0 (הקו הירש בגרף הימני למעלה), עשינו זאת גם עבור מעללה 3 וגם עבור מעללה 9.
- בפוליגום ממוללה 9 יש 10 משוואות עם 10 נעלמים ולכל ה-  $m$  (הטעות) והוא 0, אבל הוא מתאים מדי לדאטה שלנו!
- אם איננו יודעים איך נראה הגרף הירוק איך נדע להחליט ש-  $m=3$  טוב יותר מאשר  $m=9$ ?
- **נבעול וליידיצית:** נגיד ל 3 נקודות חדשות (הסגולות המלאות באירוע משמאל) שלא השתמשו בהן ב-training, ועליהן נמדד את ה- MSE, את ה-  $\text{MSE}$  (באותיו האופן שבו הוגרלו 10 הנקודות של האימון).

- Consider the error of a hypothesis/model over:

- The training set data:  $\text{error}_{\text{train}}(h)$
- The entire distribution  $F$  of data:  $\text{error}_F(h)$  ("true error" or "generalization error")

Hypothesis  $h \in H$  overfits training data if there is an alternative hypothesis  $h' \in H$  such that

$$\text{error}_{\text{train}}(h) < \text{error}_{\text{train}}(h')$$

and

$$\text{error}_F(h) > \text{error}_F(h')$$

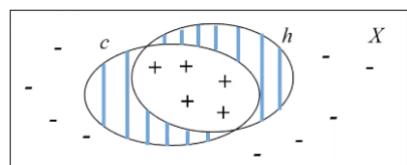
### הגדרה פומאלית ל-Overfitting :

נתבונן בשגיאה של ההיפותזה/המודול מעל :

- קבוצת הדאטה אימון :  $\text{ERROR train}(h)$
  - מעל החתפנות  $F$  של הדאטה :  $\text{ERROR F}(h)$  ("הטעות האמוטית" או "טעות הכללה")
- היפותזה  $h$  מותוך מרוחב הhipotheses  $H$  מקיים על הדאטה האימון אם **קיימת היפותזה אלטרנטיבית  $h'$  מותוך מרוחב הhipotheses  $H$  כך שמותקיים**

$$\text{error}_F(h) > \text{error}_F(h') \quad \text{וגם} \quad \text{error}_{\text{train}}(h) < \text{error}_{\text{train}}(h')$$

כלומר, הטעות של היפותזה  $h$  על הדאטה אימון קטנה מארה הטעות של היפותזה  $h'$  על הדאטה אימון וגם, הטעות של היפותזה  $h$  על כל הדאטה גודלה מהטעות של היפותזה  $h'$  על כל הדאטה = היפותזה  $h$  מותאמת מדי לדאטה אימון!



### הטעות האמוטית (במקרה של סיווג/קלסיפיקציה) :

הטעות האמוטית של היפותזה  $h$  ביחס לקונספט המטריה  $c$  היא החסתברות  $h$ - $c$  טיטה בסיווג לקלאס של  $c$

$$\text{error}(h) = \Pr_{x \sim F}[c(x) \neq h(x)]$$

דגימה שנשלפה באופן רנדומלי מותוך החתפנות של הדאטה  $F$  וו למשה החסתברות להימצא בשווה המוקוק וכחול באירוע המזג. השגיאה הזו תלויה מאוד בחתפנות הדאטה! (ב- $F$ )

## הערכת סטטיסטית Statistical Estimation

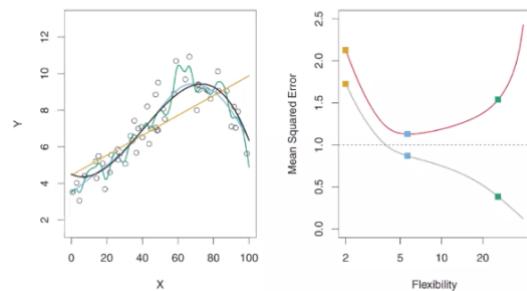
- נוכל להשתמש ב- test set במטרה להעריך את השגיאה האמיתית של היפוטזה מועמדת/מודל מועמד.
- אם אס-set הוא היתרה של X נדע את השגיאה האמיתית! אבל זהו מצב לא ריאליסטי כמובן – אנחנו חיבים להסתמך על דוגימות.
- נגידר את טעות הדגימה, עבור סט של דוגימות באופן הבא:

  - כל שקבוצת הדוגימות S גודלה יותר, כך החערכה תהיה טוביה יותר. נרצה להבין את האיכות של ההערכה שלנו.
  - הדבר נכון לשאלת הבאה בסטטיסטייה: הערכו את הקבוצה היחסית של האוכלוסייה (אחיזה מהאוכלוסייה) בעלת תוכונה מסוימת.
  - במקרה שלנו, התוכונה של כל x באוכלוסייה X היא שההיפוטזה שלנו  $\hat{y}$  מושגת את x באופן שני.

- **התפלגות של טעות הדגימה – התפלגות ביןומית (הצלה וכישלון של קבוצת ניסויים בלתי תלויים)**

  - עבור דגימה ספציפית א, נסמן את ההסתברות לניסי-קלסיפיקציה המוגדרת על ידי error F (h) באאות  $\alpha$ .
  - נניח כי קבוצת המוגדים שלנו מכילה  $n$  דוגימות רנדומליות שהוגלו באופן בלתי תלוי מ-X.
  - هي R משתנה מקרי שמודר להיות מספר השגיאות (הניסי-קלסיפיקציות) שתניב ההיפוטזה  $\hat{y}$  כאשר נפעיל אותה על קבוצת המוגדים שלנו.

## Using a validation set



$$\text{Prob}(R = k) = \frac{n!}{k!(n-k)!} \cdot p^k (1-p)^{n-k}$$

אזי:

אבל אנחנו לא יודעים את  $p$ ? לכן علينا לחשב הערכה עבור  $p$ .

### התהליך של ההערכת סטטיסטית Statistical Estimation Procedure

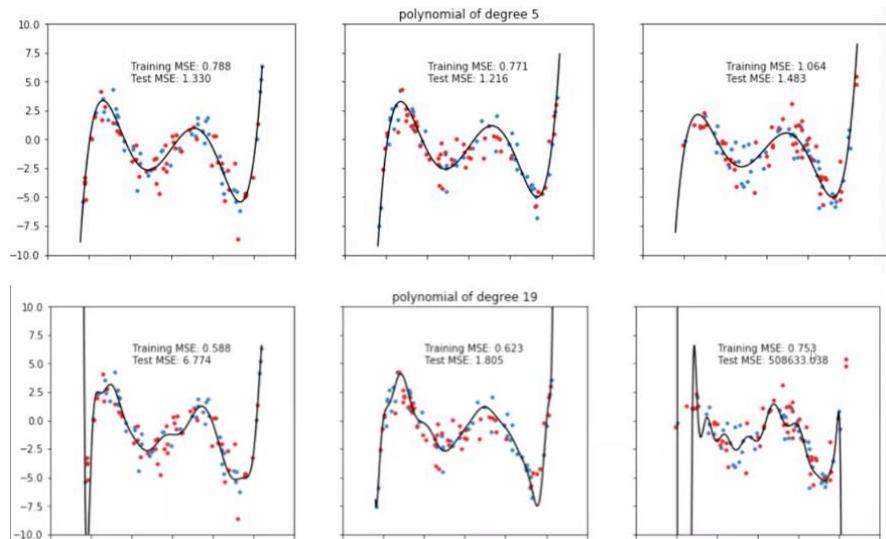
- השתמש ב"test set" בגודל  $T = |S|$ . ונניח כי מספר הטעויות הוא  $r$ .
- נוכל להראות כי  $T/r$  הוא הערכה עבור הטעויות המומכללת.
- בתלות על הגודל של סט הבדיקה שלנו, נוכל להפיק הבטחה סטטיסטית כגון:

$$\frac{r}{n} + \varepsilon$$

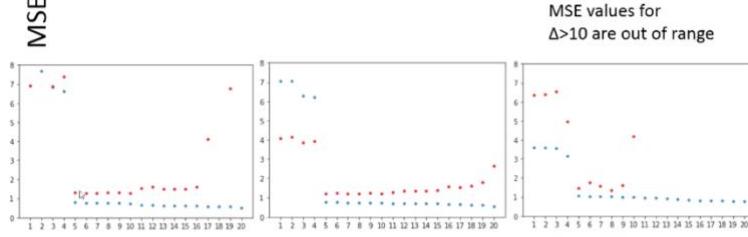
תהליך זה נקרא גם confidence intervals for proportion estimates.

### דוגמא נוספת:

יש כאן 50 דוגימות אדומות שמהוות טסט ו-50 דוגימות כחולות, של פולינום ממעלה 5 עם רעש. training test MSE מיצג עבורנו את טעות ההכללה (טעות הכללה האמיתית מחושבת על אינספור גבולות). ניתן לראותה שהיא MSE הטעות ביותר הוא כמובן, במעלה 5, שכן את זה רק לאחר שנמשיך למעלות הבאות (להלן מעלה 19 שמניב MSE עם ערך גבוה מאוד).



MSE



$\Delta$  →

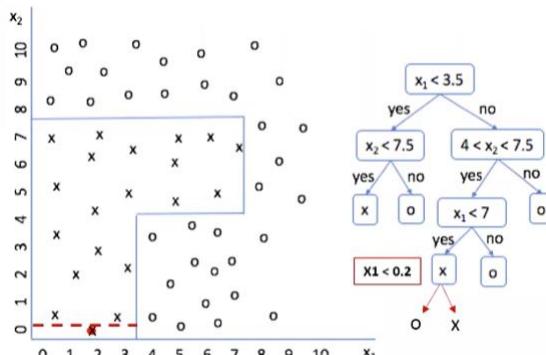
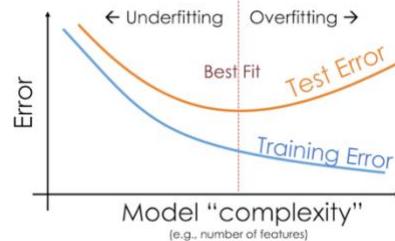
— בדוגמה לעיל הינו המעלות של הפולינומים, ובעצמי החלטה המורכבות של המודל מתרבطة בעומק העץ/גובה העץ.

— **Best Fit** – הנקודה בה השגיאה על ה-*training set* קטנה ועל ה-*test set* היא מינימלית. בנקודה זו כדאי לעצורקדם את מורכבות המודל. נקודה זו תלויה בגודל הדadata.

נציג את ה-**MSE** כפונקציה של השינוי במעלה של הפולינומים

שבדקו

ניתן לראותות שבעמלוות נוכחות 1-4 ה-**MSE** הוא מאוד גבוה בשני הסתמים, לעומת 5 אנחנו כבר ירודים ל-**MSE** נמוך יותר מאשר בשני הסטים. אבל, בעמלוות הגבוהות ה-**MSE** של ה-*test set* הולך ונעשה נמוך יותר, השגיאה קטנה, ואנו מותאמים את המודל שלנו מידי לדאטה, מפני שבמקביל ה-**MSE** של ה-*test set* הולך וגדל!

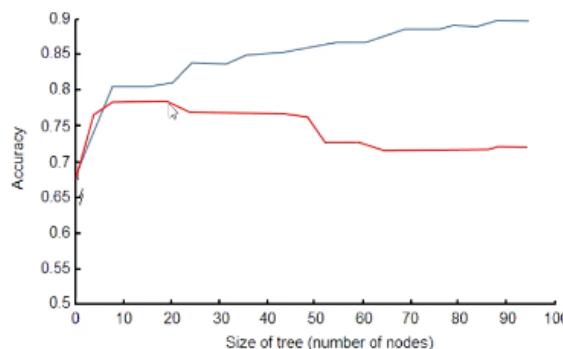
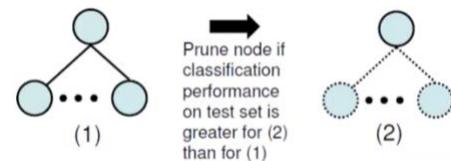


**בחזרה לעצמי החלטה** – נחזור לדוגמה זו שראינו כבר בהרצאה 2, ונניח כי הייתה לנו טעות בדadata ומדובר דוגמה שהיא עוגול כאייקס (מסומן באדום) והוא יהיה علينا לשאל עוד שאלת כדי להגיע לעלים מונוכרומטיים (טוהרים).

העץ האדום הוא **overfitting**, מפני שהוא מתאים מידי ל-*training data* שלנו, אך יש מוטיבציה לבצע **pruning** באמצעות validation set (למשל).

#### :Post-Pruning using a Validation Set

- נפצל את הדadata לקבוצות אימון ולבוצות בדיקה
- נבנה עץ על ה-*training data* (עם או בלי בסיס לקיצוץ, למשל chi-square)
- נקט צמתים אשר עוברים הביצועים של הסיווג/הקלסיפיקציה טובים יותר על הטעטט סט כאשר מבטלים את הילדיים שלהם.



הגרף הבא מတיר את הדיק בקלסיפיקציה כפונקציה של גודל העץ ( מבחינת מספר הצמתים) כפי שניתן לראותות הכו הכו מတיר את הדיק עבור ה-*training data* עבור ה-*validation set* שホールך ומשתפר בכל שטדיים את העץ ובכך נוצר overfitting. הגרף האדום מတיר את הדיק של המודל שלנו עבור ה-*test set*. נשים לב שהשלב מסויים, ככל שנעשה התאמת עבור ה-*test set*, הדיק עבור ה-*test set* הולך ופוחת, מה שמתאים לנו להגדירה של overfitting

	Less than 1.70m	1.70-1.90	Taller than 1.90
Women	4/6	4/6	8/9
Men	1/2	1/2	23/27

### הכנה להרצאה הבאה – A prolog on Bayesian Learning

**פרזנטס'** – נאספו נתונים אודוטיים שחקני כדורסל שהצלחו לפחות 5/5 סלים מהקו.  
הדאטה מוצגת בטבלה משMAIL ומפולחת על פי גובה ומין.  
האינטואיציה אומרת – פלח גדול יותר של הנשים בכל קטגוריה צלחו את הקלייעת, ולכן הנשים יותר טובות לפי אינטואיציה זו.  
אבל כאשר סוכמים את מס' הנשים שצלחו לעומת הגברים שצלחו, מקבל **שיעור גברים צלחו**.

MAP classification (next week)

### המשך הכנה להרצאה הבאה – Parameter and model estimation

- בהינתן נתונים נרצה להעריך את פונקציית הצפיפות של הדאטה = PDF
- .Probability Density Function

Classify an instance with observed properties  $\vec{x}$  as  

$$\operatorname{argmax}_i P(\vec{x}|A_i)P(A_i)$$

### הערכת סבירויות מקסימלית – MLE = Maximum Likelihood Estimation

- גישה ישירה עבור הערכת פרמטרים אשר עובדת באופן ישיר על מקרים פשוטים ויצירת בסיס רעיוני
- עבור רוב הגישות העוסקות בערכת פרמטרים.
- בהינתן סט דוגמאות  $\{x_1, \dots, x_m\} = D$  ומודול וקטור מועמד של הפרמטרים של המודל זהה, טהה.
- נגידיר את הסבירות (Likelihood) של כל מודל מועמד בהינתן הדאטה להיות  $(D | \theta)$ ,  $L(\theta)$ , למען זאת משתמש ב- $\operatorname{log}$ -. $\operatorname{theta}$ .
- היחסטריות לראות את הדאטה בהינתן הוקטור  $\operatorname{log}-\operatorname{likelihood}$   $LL(\theta) = \operatorname{log} P(D | \theta)$
- נרצה למינס את הסבירות, אם וקטור טהה ממקסם סבירות הוא גם ימקסם את  $\operatorname{log}-\operatorname{likelihood}$ . ולכן MLE אנחנו מחפשים את :

$$\theta^* = \operatorname{argmax}_{\theta \in \Omega} LL(\theta)$$

### MLE for independent identically distributed instances

- לרוב נניח כי דוגמאות הדאטה נוצרות ממשתנים מקרים שהם i.i.d.
- $\theta^* = \operatorname{argmax}_{\theta \in \Omega} \sum_{i=1}^m P(x_i | \theta)$
- לכן, עליינו למצוא את :

### זוגמה של הטלה מטבע (לא נדע מהו ה- $p$ של המטבע וננסה להעריך אותו)

- נניח כי יש לנו מטבע בעל הסתברות  $p$  להיות (H) ו- $1-p$  להיות (T).
- נזרוק את המטבע  $m$  פעמים, ונتابון בקבוצה של H-ים ו-T-ים.
- Observation

$$L(\Theta) = \operatorname{log} P(D | \Theta) = \operatorname{log} p^m (1-p)^{N-m} \\ = m \operatorname{log} p + (N-m) \operatorname{log}(1-p)$$

- חסר  $N$  choose  $M$  =  $N! / (M!(N-M)!)$ , אבל זה קבוע, זה לא ישנה את הטלה הממקסמת ולכןazon אותה (אנו מושווים ל-0).
- נרצה למצוא את ה- $p$  שմביא את הנוסחה לעיל למינס ולבסוף גוזרים את הליליקליאוד לפי  $p$  ומשווים ל-0:

$$\frac{dL(\Theta)}{dp} = \frac{d(m \operatorname{log} p + (N-m) \operatorname{log}(1-p))}{dp} = \frac{m}{p} - \frac{N-m}{1-p} = 0$$

$$\rightarrow p = m/N$$

### לוגמה של ההסתגלות הגאוסיאנית

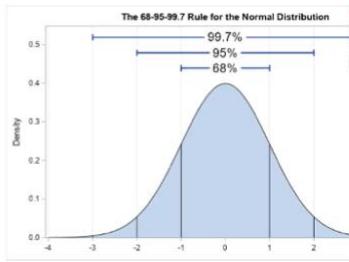
נתבונן במשפחה המשתנים שמתפלגים נורמלית/אוסיאני אשר מאופיינים על ידי שני פרמטרים מיו = תוחלת, וסיגמא = סטיית תקן. (כל שחשsigma יותר קטנה = עמוק יותר צר). נתבונן במדידות  $\{x_1, \dots, x_n\} = D$  אשר אנו מניחים שהן מוגעות מتوزע ההסתגלות הנורמלית. במקרה זה

$$L(\theta) = \prod_{i=1}^n \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x_i - \mu)^2}{2\sigma^2}\right) \text{ : likelihood}$$

מתקיים כי,  $\theta = (\mu, \sigma^2)$ . וכן, פונקציית ה-log-likelihood :

## Normal Distribution

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$



- Normal distributions are determined by two parameters:  $\mu$  and  $\sigma$ .
- Given  $m$  values of a variable  $X$ , we want to estimate the mean and variance of its normal approximation:

יותר נוח לעבד עם ה-log-likelihood :

$$LL(\theta) = -n \ln(\sigma\sqrt{2\pi}) - \sum_{i=1}^n \frac{(x_i - \mu)^2}{2\sigma^2}$$

כדי למצוא את המקסימום נחשב את הגרדיאנט ונשווה ל-0 :

To find a max point for this function we set the gradient to 0:

$$\frac{\partial}{\partial \mu} : \sum_{i=1}^n \frac{(x_i - \mu)}{\sigma^2} = 0 \Rightarrow \hat{\mu} = \frac{1}{n} \sum_{i=1}^n x_i$$

$$\frac{\partial}{\partial (\sigma^2)} : -\frac{n}{2\sigma^2} + \sum_{i=1}^n \frac{(x_i - \mu)^2}{2\sigma^4} = 0 \Rightarrow \hat{\sigma} = \left( \frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2 \right)^{1/2}$$

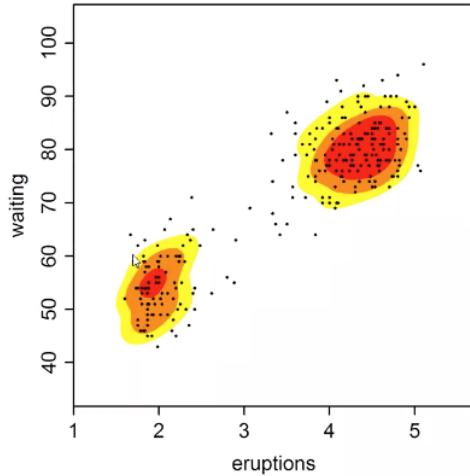
עבור מיו-קובע נוכל לחשב מפניהם שהדגימות נتوנות לנו. אך קיבל הערכה למיו.

עבור סיגמא-האט נציב את המיו-קובע בתוך הנסחה שהתקבלה.

לכן בהכרח קודם חיבר לחשב את הערכה לתוחלת ונציב אותה בהערכתה של סטיית התקן (סיגמא).

### לוגמה נוספת – התפרצויות הר הגעש Old Faithful Wyoming

למד אלגוריתם EM על מנת להעריך את ההתפרצויות הללו.



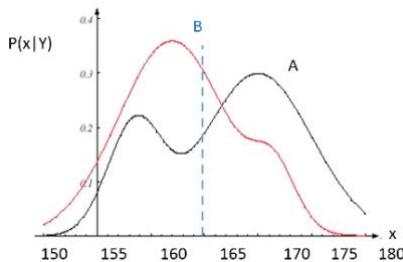
## lec4: MAP Classifier and Bayesian Learning

### למידה בייסיאנית - Bayesian Learning

בכלטיפקציה קיבל  $x$  חדש ונרצה לדעת לסוגו אותו לקלאס המתאים לו. בלמידה הסתברותית נתיחס לדגימות שלנו בעל הטעחות משותפת כלשהי וכן נדע לסוגו אותם. נניח שקיים טופס של מבחן ונרצה לנחש האם מדובר בבחן של סטודנט או סטודנטית. לצורך העניין נניח כי יש **60% ננים ו-40% ננות**, שכן לנו זה בלבד נחש שזו **מבחן של סטודנט – מינימום סיון, הסתברות לטעות = 40%**.

- בהינתן שני קלאסים A ו-B, ואשר אנחנו יודעים את ה **prior-probability** של המחלקות  $P(A)$ ,  $P(B)$
- Classify A if  $P(A) > P(B)$ ,
- Classify B otherwise :
- **נסוג מקרה חדש באופן הבא :**
- **נשים לב כי סיווג זה אינו לוקח בחשבון את המידע שיש לנו אודות הדגימה  $x$  (פיצרים).**
- **הסתברות הטעות =  $1 - \max(P(A), P(B))$**
- **לרוב לא נרצה לסוג על פי ה-*prior* אלא נרצה לנ��וט בגישה מתקדמת יותר : likelihood**

נניח כי ידוע לנו גובהם, ההסתברויות ( $A|x$ )  $P$  ו( $B|x$ )  $P$  למשל :  $P(\text{height}|\text{female})$  ו-  $P(\text{height}|\text{male})$  לדוגמה. נניח כי  $x = 163\text{cm}$  אך סביר יותר שהבחן הוא של סטודנטית מכיוון של ה-*height* של קלאס B בנקודה  $x=163$  הינה גבוהה יותר מאשר קלאס A.



**מדוע סיווג זה הינו בעייתי? אין התחשבות ב-*prior***

$P(H>1.9   NBA) = 0.85$	$P(H<1.9   NBA) = 0.15$	$P(H>1.9   R) = 0.1$	$P(H<1.9   R) = 0.9$
-------------------------	-------------------------	----------------------	----------------------

פגשנו ברוחב אדם  $1.93$ , וידוע לנו שההסתברות של שחקן NBA להיות מעל  $1.9$  היא  $85\%$  אך נסיק שהאדם הוא NBA, אבל איך שחקני ה-NBA מתחזק אוכטסיטי האנשים בעולם הוא זניח ולכן סיווג באופן זה הינו שגוי.

• What we want is the rule:

• "Classify A if  $P(A|x) > P(B|x)"$

• Not the same – why?

• What about prior probabilities?

• In our example (male/female)  $P(A) \cong P(B)$ .

But, in more general cases, even if  $P(x|A) > P(x|B)$  it may be the case that  $P(A) \ll P(B)$  (i.e. the probability of A is very very small in the first place even though the specific value  $x$  is much more common in A than in B).

**אלגוריתם MAP: Maximum A-Posteriori**

אנחנו רוצים להעריך את ( $x|A$ )  $P$  ו( $x|B$ ), כולם – בהינתן  $x$  (הדגימה)

נרצה לדעת את המცב "האומוטי הטבעי" הסביר ביותר.

המסוג שלנו ציריך לסוגו באופן הבא :

-- נסוג A אם ( $x|A$ )  $>$  ( $x|B$ )

-- אחרת נסוג B

אבל, אנחנו לא יודעים באופן ישיר את ההסתברויות ה-"posterior" הללו.

$$P(A|x) = \frac{P(x|A)P(A)}{P(x)}$$

לכן, משתמש בנוסחת בייס :

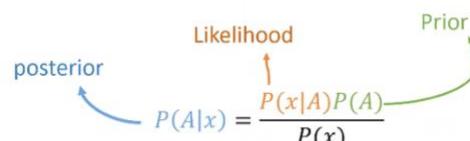
### הקומפוננטות של נוסחת בייס

לא הגדרנו את ( $x|A$ )  $P$  מפני שהוא משותף לשני הקלאסים.

$$P(A|x) = \text{Posterior}$$

$$P(x|A) = \text{Likelihood}$$

$$P(A) = \text{Prior}$$



$$C(x) = \max_{i=1..k} \frac{P(x|A_i)P(A_i)}{P(x)}$$

נסוג דגימה בעלת וקטור פיצרים  $x$  על ידי ( $k$  מספר המחלקות) :

$$C(x) = \max_{i=1..k} P(x|A_i)P(A_i)$$

אבל נוכל לחושט את ( $x|A$ ) מהנשחה, מהיות שחשבון נעשה ביחס ל-i :

## העקרונות של סיוג בייס / Principles of Bayes Classification

- הסיווג תלוי ב-likelihood (הסתברותה בהינתן הקלאס) כמו ה- prior (הסתברות הפרIORית).
- חוק בייס מניב:
- נסוע A אם  $P(A|x) > P(B|x)$
- אחרת נסוע B
- ណזון בסיווג של multiclassess בהמשך.
- נשים לב כי  $P(x)$  מוסר מן המכנה משני הצדדים מפני שבשני הצדדים הוא אותו הדבר.

### סיוג בעל שגיאה מינימלית

בכל פעם שאנו מותבוננים בערך  $x$ , ההסתברות לשגיאה תהיה:

- אם נחליט B אז  $P(A|x) = P(error|x)$
- אם נחליט A אז  $P(B|x) = P(error|x)$
- החלטת הביס היא זו שתביא למינימום את ההסתברות של הטעות (error rate).
- שימוש ב- $\min[P(B|x), P(A|x)]$  : Byes decision

$$Error_p(h) = \int P(error|x) dP(x)$$

: אילו ממש ידעונו את מבנה ההסתברות השלמה (ואנחנו לא) היינו יכולים להעריך

### פונקציית המחיר / Loss = Cost of Wrong Decision

- נניח שיש לנו  $k$  מחלקות (klassים) שונות  $A_1, \dots, A_i, \dots, A_k$ .
- על פי התובנות בדגימה  $x$ , علينا לסועג את הדגימה זו לאחד המחלקות  $A_i$  על ידי יישום גישת הסיוג Bayes/MAP. נשים לב שהחלה שגיה מניבה loss!
- Loss יכול להיות תלוי באיזה  $j$  סוג באופן שגוי ל- $i$  (מה המחיר לכך שדגימה אשר שייכת למציאות  $A_j$  ושייכת על ידי

$$\lambda_{ij} = \lambda(\text{Choose } A_i | A_j) = \begin{cases} 1 & \text{if } i \neq j \\ 0 & \text{if } i = j \end{cases}$$

האלגוריתם שלנו למחלקה  $i$ . למשל, דוגמה לפונקציית loss עבור סיוג למחלקות 0-1.

- 0 אם צדקנו (המחיר הוא 0), 1 הוא המחיר אם טעה.
- הסיכון הוא המחיר הצפוי / the expected loss, המוחושבת לפי הסתברויות אפוסטטריאניות.
- הסיכון להחלטת  $A_i$  כאשר אנו מותבוננים בדגימה  $x$  הינו:

$$R(\text{choose } A_i | x) = \sum_{j=1}^k \lambda_{ij} P(A_j | x) = 1 - P(A_i | x)$$

המעבר הראשון נכוון לכל פונקציית loss זהה תוחלת המחיר שאנו נשלם אם נחליט  $i$ .

$$\sum_{j=1}^k P(A_j | x) = 1$$

המעבר השני מתייחס ספציפית לפונקציית loss 0/1. מעבר השליishi והאחרון נסתמך על כך שהסכום על כל  $j$  הינו 1 ונחסיר מסכום זה את ההסתברות של  $P(A_i|x)$ . המשוג שלו מביא לMINIMUM את תוחלת הפסד שלנו.

לכן הסיכון המינימלי לסיוג במקרה שלו (כאשר פונקציית המחיר היא 0/1 או 0/0) יהיה:

Choose  $A_i$  such that  $P(A_i|x) > P(A_j|x)$  for all  $j \neq i$

תחת פונקציית zero-one loss השוואות הבאים שקולים

$$g_i(x) = P(A_i | x) = \frac{P(x | A_i) P(A_i)}{\sum_{j=1}^k P(x | A_j) P(A_j)}$$

נשים לב שהחישוב של  $P(A_i)$  מוחשבים מתוך הדאטה אימון (softmax כמנה) אינסטנסים מתוך הדאטה אימון הם בעלי התכונה ומחלקים במספר הדגימות, מסתמכים על כך שהוא מיצג. את ההסתברות  $P(x|A_i)$  נתנו לנו למשל בדוגמה של ה-NBA מפורשות, ובדוגמה של המבקרים והגבאים ניתנה לנו פונקציית הצפיפות עבור הסתברות זו.

$$g_i(x) = P(x | A_i) P(A_i)$$

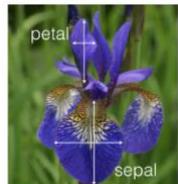
$$g_i(x) = \ln P(x | A_i) + \ln P(A_i)$$

עלינו ללמידה את החתפולוגות של ההסתברות המותנית, נבצע חישוב מתוך הקבוצה המקיימות (התנאי). נדגים את הקבוצה המקיימות את התנאי ומוכחה נחשב את ההסתברות המותנית על ידי הדגימות המקיימות את הפיצרים.

מקרה פרטי: עבור קלאסים בעלי prior זהה

- כasher כל הקלאסים השווים הם בעלי הסתברויות prior שות (prior  $P(A_i) = P(A_j)$  לכל  $i, j$ , נוכל לوتור על ה-"prior" גם ולקבל  $MLC$
- Maximum Likelihood Classifier:** שימושו נבחר  $A_i$  עבור כל  $i$  אשר שווה מ- $j$  המקיימים ( $A_j > P(x|A_i)$ ) או
- Log-Likelihood Classification:**  $\ln(P(x|A_i)) > \ln(P(x|A_j))$

### דוגמה: דאטת האירוסים של פישר



קיים שלושה סוגים של אירוסים: סטוסה, איריס וירגיניקה ואיריס ורשיולר. אלו הם הקלאסים שלנו ולכן יש לנו 3 קלאסים. נמדדנו 4 פיצרים עבורי 50 דוגימות מכל אחד מהסוגים של הפרחים המצוינים לעיל. ארבעת הפיצרים הם: **אורך ורוחב פטלי וספלי בס"מ**. لكن כל דגימה, אינסטנס א, בדאתה אימנו לנו, מחזיקה 4 משתנים (ווקטור של 4 ערכים).

$$P(\text{sepal length} = x | \text{Setosa})$$

$$= (\text{count of Setosa w sepal length} = x) / (\text{total setosa})$$

איך נחשב את  $P(x|A_i)$ ? הדרך הפחותה ביותר היא לספר:

**הבעיה בדרך זו היא:** שכאותה מחר יכול להציג פרח חדש והאורך שלו יהיה 6.1 ס"מ ולא נראה בדתאה שלנו אורך ספלי כזה, אז ההנחה היא שהסתברות לפרא כזה על פי ספירה כזו היא 0.

אבל בעולם האמיתי, כਮון שיתכן וימצא פרח כזה.

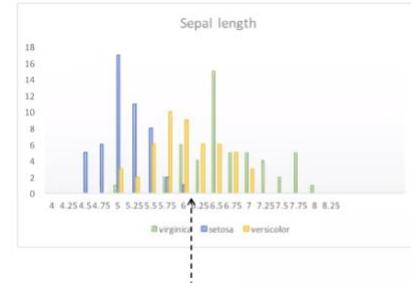
**ולכן עליינו "להתפרק" מעת.** דרך אחרת לעשות זאת היא להמיר את הדאטה **להיסטוגרמה**, מחלקים את ציר האיקס ל-bin-ים וסופרים כמה מהדאטה של נכנס לתוך ה-bin הזה. יש לנו חישובו כי עליינו לשומר מספרים במספר המ-bin-ים ששמרנו.

- In 1D we have  $m$  real values and we divide the real line into  $k$  non-overlapping bins:  $[x_i - h, x_i + h]$

$x_i$  is the center of the  $i$ -th bin.

The resulting density estimate will be:

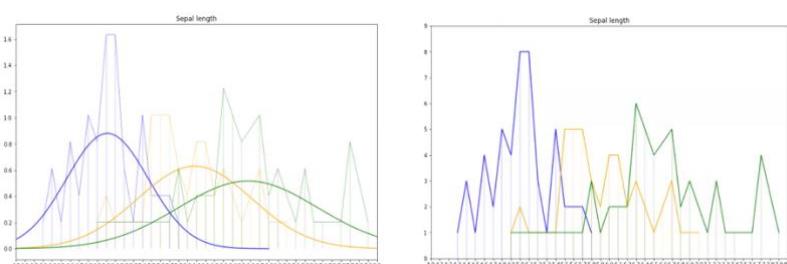
$$p(x) = \frac{\{\text{number of samples in the bin containing } x\}}{\{\text{total number of samples}\}}$$



What if the sepal length of a new instance is 6.1?

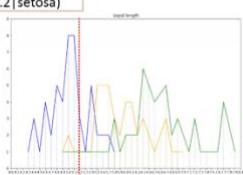
פתרון יותר אלגנטiy יהיה:

לנרטם את הדאטה/לבצע קירוב גאוסיאני על הדאטה. ניקח את כל הדאטה הכלול למשל נבעע עליו MLE, נמצאו את מיו וסיגמא ובכך ניצור עבורו קירוב גaussiano. להלן דוגמה על הפיצר אורך ספלי, מימין הדאטה לפני הנירמול ומשמאלו לאחר הקירוב gaussiano. תהליך הלמידה זהה אילץ אותנו לשומר 2 משתנים בלבד (מיו וסיגמא) עבור כל סגמנט בדאטה ולכן יש סך הכל 6. ואלה למעשהamushe the-conditional likelihood probabilities של מודנו מחדאטה, ככלומר the-h-likelihood probabilities



לכן כעת נשימוש בנוסחת ביסס כדי לחשב את ההסתברויות הללו. בהנחה שהדאטה מייצג, ה-prior הוא  $1/3$  עבור כל פלאח בדאטה (נקחו 50 דוגימות מכל סוג) ומכיון שה-prior זהה למעשה קבוע לפי ה-likelihood הסיווג עבור הדגימה המתוארת: ה-likelihood בנקודת 5.2 הוא הגבוה ביותר בדאטה הכלול, ככלומר בהינתן שהפרח הוא סטוסה, ההסתברות שאורך הספלי הוא 5.2 הוא הגבוה ביותר ועל כן, מסווג על פי הליליקיhood: סטוסה.

1.  $P(\text{sepal length} = 5.2 | \text{versicolor})$
2.  $P(\text{sepal length} = 5.2 | \text{virginica})$
3.  $P(\text{sepal length} = 5.2 | \text{setosa})$

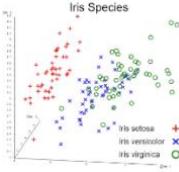


לכן כעת נשימוש בנוסחת ביסס כדי לחשב את ההסתברויות הללו. בהנחה שהדאטה מייצג, ה-prior הוא  $1/3$  עבור כל פלאח בדאטה (נקחו 50 דוגימות מכל סוג) ומכיון שה-prior זהה למעשה קבוע לפי ה-likelihood הסיווג עבור הדגימה המתוארת: ה-likelihood בנקודת 5.2 הוא הגבוה ביותר בדאטה הכלול, ככלומר בהינתן שהפרח הוא סטוסה, ההסתברות שאורך הספלי הוא 5.2 הוא הגבוה ביותר ועל כן, מסווג על פי הליליקיhood: סטוסה.

True Species	Classified Species		
	versicolor	virginica	setosa
versicolor	31 (20%)	14 (9%)	5 (3%)
virginica	12(8%)	37 (25%)	1 (0.7%)
setosa	11 (7.3%)	0 (0%)	39 (26%)

להלן מטריצת הטיעות של ניסוי זה **"confusion matrix"**

- Wrong classification - error
- Correct classification



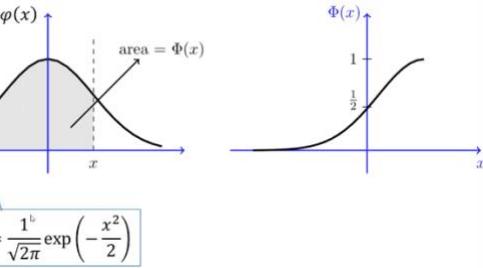
עד כה דיברנו רק על מודד אחד – האורך הספלי, יש המון אדומים בטבלה ונרצה למתן זאת, עדין לא השתמשו בתוכנות נוספות מלבד האורך הספלי. لكن עליינו לפתח את המנגנון שלנו – עליינו ללמידה התפלגיות לא לתכונה בודדת אלא **להסתברויות רבות-מימדיות**.

### התפלגיות של מספר משתנים – תוצאות / רענון

הטלת שתי קוביות היא התפלגות רב משתנית. התפלגותינו שלנו מוגדרת מעל מרחב כל הזוגות  $(i,j)$  ש- $i = 1, \dots, 6$ ,  $j = 1, \dots, 6$ .

כאשר אנו מתייחסים לשתי הקוביות הן הוגנות ובלתי תלויות, אויב פונקציית התפלגות היא יוניפורמיית מעל 36 תוצאות אפשריות.

אם אנו לא מתייחסים שאין תלות, כלומר לכארה יש תלות, ניתן להגיד שההתפלגיות השוליות עדין הוגנות (סוכמות לשישית) אך החהתפלגיות הפנימיות אינן 1/36. ככלור החהתפלגות המשותפת אינה יוניפורמיית!



$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

נאמר כי איקס מתפלג נורמלית סטנדרטית, אם איקס מתפלג נורמלית ומיו = 0, סיגמה = 1. האינטגרל של פונקציית צפיפות חד ממדית הוא 1. בפונקציה זו לכל איקס אנו יודעים את הצפיפות.

$\Phi(x)$  = פונקציית ההסתברות המצטברת (השיטה האפור בגרף מתאר את ההסתברות לערך קטן מאייקס)

### פונקציית הצפיפות הוגוסיאנית

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp\left[-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})' \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu})\right]$$

מיו מייצג את וקטור התוחלות של התפלגיות השוליות ( $d$ -ממדי)

סיגמה מתאר את מטריצת השונות המשותפת (covariance) מטריצה ריבועית  $d \times d$ , נדרש שתהייה הפיכה

משום שאין מבעאים עליה חזקה  $= \mathbf{A}^{-1}$ . המטריצה הזו היא תמיד סימטרית, חיובית ו-semidefinite.

$d$  מייצג את מספר המימדים.

לפני האקספוננט, במכנה, הסיגמה בערך מוחלט זו הדטרמיננטה של מטריצת השונות המשותפת.

בחזקה חזק = שורש הדטרמיננטה של מטריצת השונות המשותפת.

ז היא דוגימה אשר מיוצגת על ידי וקטור  $d$ -ממדי

$\boldsymbol{\mu} - \mathbf{x}$  הוא וקטור  $d$ -ממדי מפni שזיהו חישור בין שני וקטורים  $d$ -ממדיים, מופיע פעמיים

בצורתו המקורית ולפni מופיע בצורה transpose (על כן חזקה  $t$ )

הנפח מתחת לפונקציה זו הוא 1. ערך הפונקציה (גובה האוכל) לא תמיד יהיה קטן מחד, אבל במקרה זה כן (נורמלית סטנדרטית).

פונקציה זו מקבלת וקטור (עם מספר ערכים) ועליה להחזיר סקלר (כלומר זוהי פונקציה המפה מ- $R^d$  ל- $R$ ). הסקלר הוא הגובה באוותה הנקודתית וזהו הצפיפות של הווקטור.

ישנם בסופו של דבר 2 פרמטרים אשר מגדרים את התפלגות זו גם גם: בפונקציית צפיפות חד-ממדית מיו מזוי

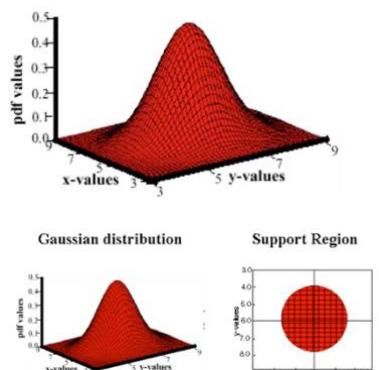
את הפעמו לאורך ציר האיקס (ימינה ושמאליה) וסיגמה מגדרה את הרוחב (עמוקן צר או רחב). בפונקציית צפיפות

רב ממדית מיו יקבע את מרכז "האובל". סיגמה = מטריצת השונות המשותפת, מגדרה את הצורה של

"האובל". אם המטריצה היא מטריצת היחידה או אלכסונית בעלת ערכים שווים באלכסון, אז האובל יהיה כיפה

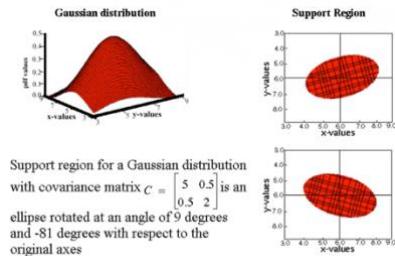
מושלמת. עבור מטריצה שאינה אלכסונית – הערכים באלכסון יקבעו את היחס בין שני הצלרים של האליפסה,

הزوויות של האליפסה נקבעת על פי הערכים שמחוץ לאלכסון.



Gaussian distribution with identity covariance matrix has equal variances in all directions

Support region for a Gaussian distribution with identity covariance matrix  $C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$  is a circle



Support region for a Gaussian distribution with covariance matrix  $C = \begin{bmatrix} 5 & 0.5 \\ 0.5 & 2 \end{bmatrix}$  is an ellipse rotated at an angle of 45 degrees and -81 degrees with respect to the original axes

- למעשה אנחנו מחשבים את גובה האוחל שרוואה הוקטור  $x$  (דו ממד) ולכן נצפה לקבל סקלאר נשים לב כי באקספוננט, אנחנו מכפילים קודם את המטריצה באינברס בווקטור  $\text{min}-x$  ואז נקבל ווקטור בימיד  $d$  שנכפיל אותו בווקטור  $\text{min}-x$  בצורת  $x$  transpose כך שלמעשה מדובר במכפלה פינימית של שני הוקטורים.

- נשים לב כי "הראשינו" העצמנו לשים במטריצה ערכים ריבועיים (סיגנומה 1 ביריבוע למשל) מהיות שאנו יודעים כי מטריצת השונות המשותפת מחייבת להיות סימטרית, חיובית ו-semidefinite. התקבלה מכפלה של פונקציית צפיפות גאוסיאנית.

$$\begin{aligned} p(x; \mu, \Sigma) &= \frac{1}{2\pi \begin{vmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{vmatrix}^{1/2}} \exp \left( -\frac{1}{2} \begin{bmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \end{bmatrix}^T \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}^{-1} \begin{bmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \end{bmatrix} \right) \\ &= \frac{1}{2\pi(\sigma_1^2 \cdot \sigma_2^2 - 0 \cdot 0)^{1/2}} \exp \left( -\frac{1}{2} \begin{bmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \end{bmatrix}^T \begin{bmatrix} \frac{1}{\sigma_1^2} & 0 \\ 0 & \frac{1}{\sigma_2^2} \end{bmatrix} \begin{bmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \end{bmatrix} \right) \\ &= \frac{1}{2\pi\sigma_1\sigma_2} \exp \left( -\frac{1}{2\sigma_1^2}(x_1 - \mu_1)^2 - \frac{1}{2\sigma_2^2}(x_2 - \mu_2)^2 \right) \\ &= \frac{1}{\sqrt{2\pi}\sigma_1} \exp \left( -\frac{1}{2\sigma_1^2}(x_1 - \mu_1)^2 \right) \cdot \frac{1}{\sqrt{2\pi}\sigma_2} \exp \left( -\frac{1}{2\sigma_2^2}(x_2 - \mu_2)^2 \right). \end{aligned}$$

- MAP  $\Rightarrow \arg \max_i P(A_i | x) = \arg \max_i \frac{P(x | A_i)P(A_i)}{\sum_j P(x | A_j)P(A_j)}$
- Dropping  $P(x) \Rightarrow \arg \max_i \{P(x | A_i)P(A_i)\}$
- ML - Assuming  $P(A_i) = P(A_j) \Rightarrow \arg \max_i \{P(x | A_i)\}$
- Using log probability  $\Rightarrow \arg \max_i \{\ln P(x | A_i) + \ln P(A_i)\}$
- Now : Naive Bayes - assuming  $P(\bar{x} | A_i) = \prod_j P(x_j | A_i) \Rightarrow \arg \max_i \{P(A_i) \prod_j P(x_j | A_i)\}$

נשתמש בכלים שלמדו כדי באמצעות **סיגומת האירוסים** לפי אורך ורוחב פטאלי וスペאלי לאחד מ-3 הסוגים. עליינו ללמידה **אוסטיאן 4-מדדי**, לשם כך נלמד 4 נווט-ים, לאחר מכן עליינו למצוא MLE למידת המטריצה סיגומה. لكن עליינו ללמידה 4  $\text{min}-i$ ים לכל סוג פרח, **סיגמה היא סימטרית** ולכן עליינו למצוא 10 פרמטרים (האלכסון, 4 פרמטרים, והערכים מעליין) למטריצה של סוג אחד. סה"כ 14 ערכים לכל סוג ולכן בטוטאל עליינו ללמידה  $3 \times 14 = 42$  ערכים. הרבה פעמים, הדאטה שלנו לא מספיק גדול כדי ללמידה את כל הערכים הללו ולכן אנחנו מניחים אי תלות בגישה שנקראת **Naïve Bayes**.

#### גדר אי תלות מותנית:

**דוגמה** : מספר הסרטים שהבנו צפה במשך חיים לעומת רמת הולסטורול בדם, מבחינות כל הדאטה וכל הגילאים שני המשתנים המתוארים הם תלויים בגיל. **הם בלתי תלויים בהינתן הגיל של הבנאים**, ככל שאתה מבוגר יותר ככל הנראה שצפטי ביוטר סרטים וכן ככל שאתה מבוגר יותר הסבירות שרמת הולסטורול בדם שלו יהיה גבוהה מאשר אדם צעיר. לכן כאשר אנו ממקדים את קבוצות המדומים שלנו בקבוצות גיל מסוימות (למשל רק אנשים בני 32-30) הם בלתי תלויים, ועל כן הם **משתנים בלתי תלויים בתנאי**.

**סיכום**:  $x$  בלתי תלוי ב- $y$  בהינתן הקלאס (גיל וכו'...)

**הגדרה**: הפיצרים הינם בלתי תלויים בתנאי (conditionally independent) (בhinntnu הקלאס, אם לכל ווקטור (רב-ממד) ערכי פיצרים  $x$  ולכל ערכי

$$P((x_1, x_2, \dots, x_d) | A_i) = \prod_{j=1 \dots d} P(x_j | A_i)$$

הקלאסים האפשריים, מותקיים:

הסתברות של חיותוך הערכים בהינתן הקלאס שווה למכפלת ההסתברויות של כל אחד מהערכים בהינתן הקלאס. זהה הנחת אי תלות. לכן עליינו ללמידה עבור כל מכפלה בצד ימין גאוסיאן חד ממד ולבסוף לככל מכפלה יש ללמידה 2 פרמטרים (צד ימין). בצד שמאל עליינו ללמידה 55 פרמטרים. לכן ברור שהלמידה של הצד ימין הינה יותר גבוהה!

```
Naive_Bayes_Learn(D-examples)
For each target class A_i
    P(A_i) ← estimate P(A_i) from D
For each attribute value x_j ∈ V_j
    P(x_j | A_i) ← estimate P(x_j | A_i) from D
```

Learning

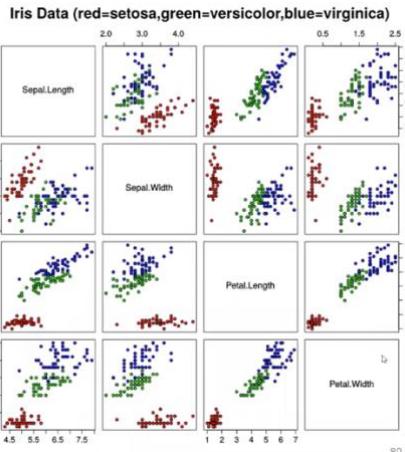
```
Naive_Bayes_Classify(x)
return A_NB = arg max_i P(A_i) ∏_{x_j ∈ x} P(x_j | A_i)
```

Execution

בסיוג על ידי Naïve Bayes אנו עושים הנחה שימושית ומפשטת, כי ערכי הפיצרים הן בלתי תלויים בתנאי בהינתן הקלאס. **אבל הנחת אי תלות בתנאי הינה לא תמיד נכונה**. ישנה דוגמה בשיעורי הבית.

#### האלגוריתם הלומד של נאיב בייס

- עלינו ללמידה הסטברויות: נרצה לקחת את הדאטה, להסיק ממנו את priors.
- ולכל ערך באטtribוט ללמידה גאוסיאן חד ממד (2 פרמטרים המגדירים את הגאוסיאן)



### הדעתה של פישר (scatterplots של הדעתה אודות האירוסים):

- אם האורך הפטאלי והרוחב הפטאלי הם בלתי תלויים בתנאי?
- ראשית כל נשים לב, הם אינם בלתי תלויים מפני שدادטה בלתי תלוי נראה כמו עננה, וכך נראה יחסית קו ישר. אז אינם בלתי תלויים
- אבל האם הם בלתי תלויים בתנאי? בשליל מהם יהיה בלתי תלויים בתנאי – כלל אחד מהצבעים יראו כמו עננים. הען הירוק דומה מידי לכאן, הכהול והאדום ייחסית נראים מעוננים.
- גם אורך פטאלי ורוחב ספלי אינם בלתי תלויים בתנאי משומש שכל אחד מהצבעים נראה ב嚷גמה של קו ולא דוווקע ענן מפוזר.
- למרות שנראה שאין כאן בהכרח אי תלות בתנאי, גישת נאיב בייס עדין הייתה עובדת על הדעתה של פישר.

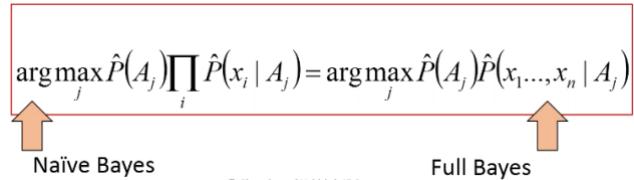
לרוב, ההנחה הנאיבית מופרת (למשל כפי שנאמר לעיל רוחב פטאלי ואורך פטאלי אינם נראים בהכרח

$$\hat{P}(x_1, x_2, \dots, x_n | A_j) \neq \prod_i \hat{P}(x_i | A_j)$$

בלתי תלויים בתנאי מפני שהם אין היה נראה יותר עני) :

אבל, בפרקтика, המשערך הזהעובד באופן מפתיע די טוב. נשים לב, כי בפועל, אנחנו לא באמצעות ציריכים שההנחה הזו תתקיים ותהיה נכונה. אנו רק ציריכים שהבא יתקיים (תנאי יותר חלש) :

אנו צריכים שה- $j$ -שסתוברו שמתאפשר בנאיב בייס ומקיים הסתבותות מקסימלית, יהיה אותו ה- $j$ -שסתובROL ללא ההנחה של נאיב בייס (בפוג' בייס). ככלומר אנחנו לא צריכים שככל אחד מהאיברים  $j$  יהיה שווה, אם אכן מתקיים השוויון, או ברור שהמקסימום מתקבל באופן המקיים וערך זהה. אנחנו צריכים שהמקסימום יתקבל עבור  $j$  שהוא מתקבל בשתי הגישות וזה דרישת הרבה יותר חלה.



בשיעור בית נראה מקרה בו נאיב בייס באמת לא יעבד.

## למודה חיישונית ממכ不失, תרגול 3

### אלגוריתמים הסטברותיים

האלגוריתם האינטואיטיבי ביותר הוא להחזיר את הקלאס הרוב, או במקרים אחרים להחזיר את הקלאס הסביר ביותר לדאטה אימון. היום נלמד אלגוריתמים הסטברותיים – אלגוריתמים שימושים בטכניקות הסטברותיות כדי לסייע את הדגימה החדשה לקלאס.

### חוזה קצהה בסטברות:

- מרחב מדגם – הוא קבוצה של מאורעות events, שזו רשימה של כל התוצאות האפשרות של המאורע. מרחב הדגימות יכול להיות גם רציף (גבאים וכו')
- מאורע – תת קבוצה של מרחב המדגם. זריקה של שתי קוביות שהסכום על הקוביות הוא 7.
- משתנה מקרי – פונקציה מרוחקת המדגם ומחזירה ערך רציף. למשל סכום של שתי קוביות. לעילו ניתן לשאול מה ההסתברות שאיקס שווה לתוצאה מסוימת. למשל מה ההסתברות שאיקס = 1 עבור איקס המתאר את סכום על שתי הקוביות (ההסתברות היא 0).
- על משתנים מקרים אפשר לשאול מה התוחלת שסכום ב-n ביוונית (expected value),

$$E[X] = \sum_x xp(x)$$

עבור משתנה בדיד התוחלת הינה

$$E[X] = \int_{-\infty}^{\infty} xf(x)dx$$

כאשר f היא פונקציית הצפיפות.

- שונות – מסומנת על ידי סיגמא בריבוע. מוגדרת להיות:

$$\sigma = \sqrt{var(X)} = \sqrt{E[(x - \mu)^2]}$$

סטטיסטית – היא שורש השונות

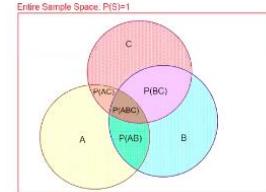
- $P(A \cup B) = ?$   

$$P(A) + P(B) - P(A \cap B)$$
- $P(A \cup B \cup C) = ?$   

$$P(A \cup B) + P(C) - P((A \cup B) \cap C) =$$
  

$$P(A) + P(B) - P(A \cap B) + P(C) - P(A \cap C) - P(B \cap C) + P(A \cap B \cap C) =$$
  

$$P(A) + P(B) + P(C) - P(A \cap B) - P(A \cap C) - P(B \cap C) + P(A \cap B \cap C)$$



### חוזה קצהה על הסטברות מותנית

#### דוגמה עבור הסטברות מותנית:

נתון כי ההסתברות לעبور את המבחן היא  $P(\text{pass}) = 90\%$  ולהיכשל  $P(\text{fail}) =$

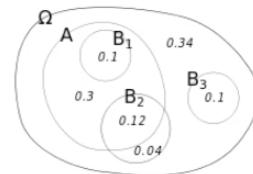
- $P(\text{Learn for the test}|\text{Pass}) = 90\%$
- בונוס ידוע לנו כי:  $10\% \times 90\% = 9\%$
- $P(\text{Learn for the test}|\text{Fail}) = 5\%$
- $P(\text{Did not learn}|\text{Fail}) = 95\%$

#### Conditional probability:

- $P(A|B) = \frac{P(A \cap B)}{P(B)}$
- $P(A|B_1) = ?$   

$$\frac{P(A \cap B_1)}{P(B_1)} = \frac{0.1}{0.1} = 1$$
- $P(A|B_2) = ?$   

$$\frac{P(A \cap B_2)}{P(B_2)} = \frac{0.12}{0.16} = 0.75$$



- $P(\text{Pass} \cap \text{Learn for the test}) = P(\text{Pass}) \times P(\text{Learn for the test}|\text{Pass}) = 90\% \times 90\% = 81\%$
- $P(\text{Pass} \cap \text{Did not learn}) = P(\text{Pass}) \times P(\text{Did not learn}|\text{Pass}) = 90\% \times 10\% = 9\%$
- $P(\text{Fail} \cap \text{Learn for the test}) = P(\text{Fail}) \times P(\text{Learn for the test}|\text{Fail}) = 10\% \times 5\% = 0.5\%$
- $P(\text{Fail} \cap \text{Did not learn}) = P(\text{Fail}) \times P(\text{Did not learn}|\text{Fail}) = 10\% \times 95\% = 9.5\%$
- $P(\text{Learn for the test}) = P(\text{Pass} \cap \text{Learn for the test}) + P(\text{Fail} \cap \text{Learn for the test}) = 81\% + 0.5\% = 81.5\%$

מה ההסתברות של מבחן אחד למכות  $0.815$  להלן החישובים הדרושים לפתרון

- $P(\text{Pass}|\text{Learn for the test}) = \frac{P(\text{Pass} \cap \text{Learn for the test})}{P(\text{Learn for the test})} = \frac{81\%}{81.5\%} = 99\%$
- $P(\text{Fail}|\text{Learn for the test}) = \frac{P(\text{Fail} \cap \text{Learn for the test})}{P(\text{Learn for the test})} = \frac{0.5\%}{81.5\%} = 1\%$
- $P(\text{Pass}|\text{Did not learn}) = \frac{P(\text{Pass} \cap \text{Did not learn})}{P(\text{Did not learn})} = \frac{9\%}{18.5\%} = 49\%$
- $P(\text{Fail}|\text{Did not learn}) = \frac{P(\text{Fail} \cap \text{Did not learn})}{P(\text{Did not learn})} = \frac{9.5\%}{18.5\%} = 51\%$

#### Independent events

- If  $P(A \cap B) = P(A)P(B)$  then A & B are independent
- From conditional probability we get:

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

↓

$$P(A \cap B) = P(A|B)P(B)$$

- If A & B are independent:

$$\begin{aligned} P(A)P(B) &= P(A \cap B) = P(A|B)P(B) \\ P(A) &= P(A|B) \end{aligned}$$

\* And also  $P(B) = P(B|A)$

#### מתבקש לשאול, ולהלן החישובים

- מה ההסתברות שסטודנט עבר בהינתן שלמד?
- מה ההסתברות שסטודנט נכשל בהינתן שלמד?
- מה ההסתברות שסטודנט עבר בהינתן כי לא למד?
- ומה ההסתברות שסטודנט נכשל בהינתן כי לא למד?

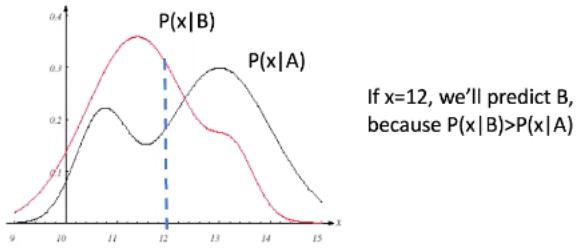
#### גדר מאורעות בלתי תלויים:

אם ההסתברות לחיצוך שווה לכפלה ההסתברויות אזוי המאורעות בלתי תלויים.  
המסקנה: זה אחד מהם קrho, לא גורע ולא מוסיף למאורע השני.

כעת, נוכל להתקדם למושגים שייתר רלוונטיים עבור Bayes

#### הסתברות פrioriy / Prior Probability

וזיא ההסתברות של המקרים בדעתם שיש לנו, האלגוריתם הנאי שтвор עמוד הקודם משתמש בהסתברות prior. כמובן, חלוקה הפנימית של מאורעות, למשל 40% בניים ו-60% בנות.



#### הסתברות הליליקיליהו / Likelihood Probability

זו היא הסתברות שמותנית על ידי כלל: **ההסתברות של דגימה x בהינתן הקלאס**. למשל עבור דגימה  $x=12$   $x$  ושני קלסים אפשריים A ו-B: האודומה היא ההסתברות של B והשchorה היא A. נחשב את ערך ההסתברות בשני הגרפים עבור  $x=12$  ונראה כי ההסתברות יותר גבוהה עבור קלאס B ולכן נסוג B. אם נזכיר לדוגמה עבר/נכשל, הסתברות זו היא כמו לשאול מה ההסתברות שימושו למד למבון בהינתן שעבר. אבל, אנחנו רוצים לדעת מה ההסתברות לעבור או להיכשל בהינתן שלמדנו. אז אנחנו צריכים דרך לעבור מההסתברות posterior probability - likelihood probability.

#### הסתברות הפוסטורייר וחוק בייס / Posterior Probability and Bayes rule

חוק בייס מיניב עבורהו את ההסתברות הפוסטורייר = מה ההסתברות לקלאס, בהינתן האינסטנס.

כדי לחשב את הפוסטורייר נצטרך להשתמש בהסתברות הליליקיליהו והפריר (במונה) ובהתברות המכנה (אווידנס).

- We will classify A if

$$\begin{aligned} P(A|x) &= \frac{P(x|A)P(A)}{P(x)} > \frac{P(x|B)P(B)}{P(x)} = P(B|x) \\ P(x|A)P(A) &> P(x|B)P(B) \end{aligned}$$

- Note that  $P(x)$  is removed from both denominators simply because it is the same

מסוג שיסוג A אם מתקיים  $(x|A) > P(B|x)$ , הוא מסוג שմביא למקסימום את הסתברות הפוסטורייר – MAP Classifier. ה-MLP Classifier תלויה בפריר והיליליליהו, האוודנס לא עניין אותנו מפני שהוא במכנה והוא שווה בכל החישובים שלו.

#### שווייה מינימלית במסוג מסוג MAP

But, we classify B only if  $P(B|x) > P(A|x)$ , and therefore the probability of the error is minimal

$$P(\text{error}|x) = \min[P(A|x), P(B|x)]$$

ישפה אכן, זה לא שאנו מזערים את הטיעות הגלובלית העולמית של הפרידקציה, אלא בתיחס להסתברויות של מדנו מהזאת הטיעות היא מינימלית. כמובן אם ההסתברויות של הדעתה אכן מייצגות את העולם האמיתי, אכן השגיאה היא מינימלית ביחס לעולם האמיתי.

0-1 loss (the simplest one):

$$\lambda_{ij} = \lambda(\text{Choose } A_i | A_j) = \begin{cases} 1, & \text{if } i \neq j \\ 0, & \text{if } i = j \end{cases}$$

הגדרת פונקציית loss: אם בחרתי את  $A_i$  בהינתן שהקלאס המכון היה  $A_j$  ניתן דוגמה

של 0/1 loss

**בhinuten פונקציית loss נגידיר את הסיכון risk:** זהו בעצם הסיכון לבחור קלאס מסוים, למשל Ai בהינתן האינסטנס, זהו החפסד כפול החסתברות של הקלאסים. אנחנו עושים סכום על כמות הקלאסים k, של הלויס על כל אחד של הפוסטוריורים של הקלאסים.

$$R(\text{Choose } A_i | x) = \sum_{j=1}^k \lambda_{ij} P(A_j | x) = \sum_{j \neq i} P(A_j | x) = 1 - P(A_i | x)$$

דוגמה לSTITCON במקורה שפונקציית הלויס הינה 0/1 loss:  
nocel להגידיר פונקציות לוס הרבה יותר מורכבות.

מסוג שרוצה למזער את הסיכון יבחר Ai כך ש:

$$g_i(x) = P(A_i | x) = \frac{P(x|A_i)P(A_i)}{\sum_{j=1}^k P(x|A_j)P(A_j)} = P(x)$$

הסתברות הפוטוריור המלאה (לפני שנשميית את (x) (P) הינה על פי ביסיס:

ולאחר שנשמיית את (x) (P) (המכנה) יתקיים:

### בינט עבור קלאסים רבים

כדי להפוך את תהליך הסיווג ליותר יעיל nocel להשתמש ב-(ln):

שימוש ב-(ln) עוזר להפחית את ההכפלות במספרים נוכנים (בין 0-1) (חוקי לוגריתמים, והופכת מכפלות לסכוםים) וכן עוזר להתמודד יותר טוב עם פונקציית הצפיפות הנורמלית (x) (e). אנחנו יכולים להשתמש ב-(ln) מפני שהיא פונקציה מונוטונית עולה.

### היפותזות מקסימום ליליאיה Hypothesis

- נרצה לבחור את ההיפותזה שיש לה את ההיפותזה הכימבריה (בעל התהסתברות המקסימלית) בהינתן הדאטה, ובכתיבת התהסתברות

$$P(h|D) = P(D|h)P(h)$$

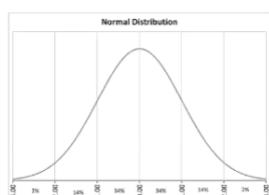
נחשף אחר התהסתברות הפוטוריורית הבאה המקסימלית:

- nocel להניא שכל היפותזות במרחב היפותזות יש את אותו הפרירוי (P), ולכן nocel למצוא את ההיפותזה הכימבריה h על פי ה-

$$h_{ML} = \underset{h \in H}{\operatorname{argmax}} p(D|h)$$

: maximum likelihood

לכן למעשה היפותזה שמקסימת את הליליאיה היא היפותזה בעלת התהסתברות המקסימלית בהינתן הדאטה.  
נרצה למצוא את ML-h. nocel להניא שכל האינסטנסים הם בלתי תלויים זה בזה ועל כן מתקיים:



$$h_{ML} = \underset{h \in H}{\operatorname{argmax}} p(D|h) = \underset{h \in H}{\operatorname{argmax}} \prod_i P(y_i|h)$$

$$= \underset{h \in H}{\operatorname{argmax}} \prod_i \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(e_i - h)^2}{2\sigma^2}}$$

$$= \underset{h \in H}{\operatorname{argmax}} \prod_i \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(h(x_i) - y_i)^2}{2\sigma^2}}$$

$$h_{ML} = \underset{h \in H}{\operatorname{argmax}} \prod_i \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(h(x_i) - y_i)^2}{2\sigma^2}}$$

$$h_{ML} = \underset{h \in H}{\operatorname{argmax}} \ln \left( \prod_i \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(h(x_i) - y_i)^2}{2\sigma^2}} \right)$$

$$h_{ML} = \underset{h \in H}{\operatorname{argmax}} \sum_i \ln \left( \frac{1}{\sqrt{2\pi\sigma^2}} \right) - \frac{1}{2} \left( \frac{h(x_i) - y_i}{\sigma} \right)^2$$

$$= \underset{h \in H}{\operatorname{argmax}} \sum_i -\frac{1}{2} \left( \frac{h(x_i) - y_i}{\sigma} \right)^2$$

$$= \underset{h \in H}{\operatorname{argmax}} \sum_i -(h(x_i) - y_i)^2$$

$$= \underset{h \in H}{\operatorname{argmin}} \sum_i (h(x_i) - y_i)^2$$

- **כעת נקבל:** (משמאלי)

הנחנו אי תלות וnocel השורה הראשונה מתקיים.

הנחנו שהטעות מתפלגת נורמלית וnocel והסתברות לקלבל את ה-target value, שווה להסתברות של הטעות וכניב את נוסחת התהסתברות של הטעות המתפלגת נורמלית עבור כל אינסטנס (שורה שנייה).

בשורה השלישית אנחנו מציבים את המרכיבים שלנו מהטעות, כי היפותזה נתונה.

לאחר כל זאת nocel להפעיל מנו. שיחפה את הפאי, המכפלה, לסקום. ונשמייט ערכיהם קבועים כי הם לא משתנים את המקסימום.

**לבסוף נקבל כי היפותזה הכימבריה maximum likelihood hypothesis היא זו שמצוירת את**

**MSE** = שורה אחורונה.

## סיכום עד כה:

- Prior classifier:  $P(A) > P(B)$
- ML classifier:  $P(x|A) > P(x|B)$  – assuming  $P(A) = P(B)$
- MAP classifier:  

$$P(A|x) = P(x|A)P(A) > P(x|B)P(B) = P(B|x)$$

\* Drooping  $P(x)$  from the denominator

### **איך מחשבים/משעלים את הסתברויות: הליקיליהוד, חפסטריוו**

- **Parametric estimation** – אם ידוע לנו שאנו יוכלים לנחש את סוג ההתפלגות נוכל להעריך את הפרמטרים של ההתפלגות. למשל אם נוכל לנחש שמשתנה מקרי מסוים מותפלג נורמלית נוכל לשערך עבورو את ה- $\mu$  ו- $\sigma$ , או אם הוא מותפלג פואסוני נוכל לשערך עבورو את הלמדא.
- **Non parametric estimation** – אם אנחנו לא יכולים להניח אף סוג של התפלגות על הדאטה שלנו נשתמש בהיסטוגרמה (=ספרה או ב- $K$ -smooth histogram) (שזו למעשה היסטוגרמה חלקה = Kernel Density Estimation).

### **שיעורן פרמטרי Parametric Estimation**

עבור כל קלאס נשערך את הפרמטרים של ההתפלגות על פי הדאטה אימון. אם אנחנו מדברים על ההתפלגות הנורמלית, علينا לשערך את התוחלת ואת השונות של כל קלאס:

ואם נרצה לסוג לפיה הסתברות הגבוהה ביותר בהינתן ההתפלגות הנורמלית (הליקיליהוד):

$$P(x|A_i) = \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{1}{2}\left(\frac{x-\mu_i}{\sigma_i}\right)^2}$$

הבעיה היא שכזאת מתחייב רק לאטריביוט/קלאס יחיד, מה אם יש יותר אטריביוטים? או במקרה כזו הסתברות הליקיליהוד תחושב לפי **התפלגות נורמלית רב-ממדית**. לשם כך נצטרך את וקטור התוחלות (כל ממד יהיה התוחלת של אטריביוט מסוים) ואת מטריצת השונות המשותפת (the covariance matrix) (במטריצה, השוניות שלא באלכסון הן ההסתברויות המשותפות) **Multivariate normal distribution**

$$\mathbf{S} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \dots & \sigma_{1d} \\ \sigma_{21} & \sigma_{22} & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{d1} & \sigma_{d2} & \dots & \sigma_{dd} \end{bmatrix} = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \dots & \sigma_{1d} \\ \sigma_{21} & \sigma_2^2 & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{d1} & \sigma_{d2} & \dots & \sigma_d^2 \end{bmatrix}$$

$|S|$  - is the determinant of the covariance matrix

$S^{-1}$  - is the inverse matrix of the covariance matrix

$$P(\bar{x}|A_i) = \frac{1}{\sqrt{(2\pi)^d |S|^2}} e^{-\frac{1}{2}(\bar{x}-\bar{\mu}_i)^T S^{-1} (\bar{x}-\bar{\mu}_i)}$$

### **שיעורן לא פרמטרי Non Parametric**

מכיוון שבשיעור זה איננו יודעים להניח את ההתפלגות, علينا למצוא דרך לשערך את הפרירור ( $P(A_i|A)$  ו-  $P(A|A_i)$ ). הסתברות הפרירור ( $P(A_i|A)$ ) ניתנת לשערוך מתדיירות הקלאסים בדאטה אימון (מספר המופיעים של הקלאס לחלק במספר הדגימות). לגבי הליקיליהוד, ראיינו בהרצאה שנייתן להשתמש בהיסטוגרמה וב-interpolation, ושבובע הבא נראה גישה נוספת.

#### **התמודדות עם multiple features**

כדי להעריך נכון את הליקיליהוד עבור דגימה נתונה علينا להוכיח דאטה-סט עצום: אם בידינו  $d$  אטריביוטים בדים,  $1-k$  קלאסים, מספר

האפשרויות של הליקיליהוד  $P(x_1, x_2, \dots, x_d|A_i)$  הבא הינו  $k \cdot |V_1| \cdot |V_2| \cdots |V_d|$ . עבור רק 2 אטריביוטים וקלאס אחד נוכל להגיע ל- $2^d$ .

לכן אנחנו צריכים דרך / הנחה שתעזר לנו להתגבר על בעיה זו.

לכן נוכל להניח שהאטריביוטים הם בלתי תלויים בהינתן קלאס – ברגע שנניח זאת נוכל להמיר את המשוואה למכפלת ההסתברויות של כל אטריביוט בנפרד. בדאטה מופיעים לנו כל הערכים של האטריביוטים האלה וכך הדבר ניתן לחישוב!

### הנחת נאיבית Naïve Bayes

אנו מניחים שכל האטריבוטים הם בלתי תלויים **בhinתן הקלאס**, ונקבל:

$V_{NB} = \operatorname{argmax}_i P(A_i) \prod_{j=1}^d P(x_j|A_i)$  : (Maximum A-Posterior Probability) MAP  
הקלאס שיבחר הוא זה שמקסם את הביטוי הנוכחי.

ברגע שאנו מניחים כך, אנו מורידים משמעותית את הגודל של הדאטה-סט שנטרך הבדיקה ל-

נתן להניח זאת גם במקרה הרציף, ניתן להשתמש בו גם בשיעור הפרמטרי.

## הרצאה 5 – מודלים מיקסチャרים: Bayes Variations and EM

### Density Estimation, Gaussian Mixture Models, EM

- בהינתן סט דוגמאות  $x$  נמצאת פונקציית הצפיפות (Probability Density Function) PDF – **Density Estimation**  
שמשמעותה באופן הטוב ביותר את הדadata.
- מדובר על מודלים המתאימים מצב בו יש שתי שכבות השולטות בהתפלגות. במקרה כזה ראשית יש להחליט לאיזה "ענמה" אני שוייך, ומיתוכו להחליט על סיווג סופי.

### אלגוריתם EM = Expectation Maximization

- שיטה איטרטטיבית עבור הערכת פרמטרים כאשר שכבות של DATA חסרים מה-observation.
- אלגוריתם זה כולל שני שלבים: Expectation (E)- ו- Maximization (M):  
נדיר את המשמעות באלגוריתם EM: D הוא קבוצה של DATA points (ה- $x$ )  
טטה היא וקטור הפרמטרים, EM הוא אלגוריתם איטרטטיבי עבור מציאת טטה (maximum likelihood).
- אנחנו נניח שיש שתי רמות של DATA. יש את DATA ה- $x$ , "augmented data" – ואנו לא נראתה את DATA ה- $x$ .  $x$  הוא DATA שאנו רואים "observable data" – ו-  $z$  הוא ה- $z$ .
- נראתה ה- $z$  DATA הנסתה מעניין (לא ממש חסר). התוצאות שלנו יהיו בתוך  $D$ .

$C = (X, Z)$

+ C: complete data ("augmented data")  
+ X: observable data ("incomplete" data)  
+ Z: hidden data ("missing" data)

+ D: the actual observed X values (from a sample)

### בחירה רנדומלית בין 2 מטבעות

יש לנו 2 מטבעות עם הסתברויות Pa, Pb (אינם משלימים! יכולם להיות 0.65-1 0.45).  
**אחד** המטבעות יבחר **עם** הסתברויות Wa, Wb (משלימים אחד את השני 1-Wa = 1 -Wb). לאחר מכן מטילים את המטבע הנבחר 10 פעמים. נתבון בתוצאות של הניסוי ונערך אותו מספר פעמים. אם ידוע לנו איזה מטבע נזרק בכל סט, אז נוכל לבצע MLE ולקבל את ה- $P$ -ים ואת ה- $W$ -ים. אבל אנחנו לא יודעים זאת. לכן EM יוכל לעזור כאן.

אנחנו רואים את התוצאה של 10 הטעויות 8 פעמים אבל אנחנו לא יודעים בכל פעם מהפעמים הללו איזה מטבע הוטל. נרצה להסביר מהניסיומים האלה את המטבע הבा – 10 הטעויות שהוא ייניב. האIOR עם הכתומים-ירוקים אינו ידוע לנו, אנחנו נמצאים במצב בו כל המטבעות כחולים – hidden – עבורנו.

### אלגוריתם EM – התהליך הרעויוני

- נתחל בסיס של הפרמטרים ההתחלתיים של המודל כולל הפרמטרים של  $z$  – בדוגמה שלנו יש 3 פרמטרים Pa, Pb, Wa, Wb (Pa, Pb, Wa, Wb הם ל-1 של a). קלומר "נכח" ערכי ההתחלתיים עבור הפרמטרים של המודל.
- נשתמש בפרמטרים אלו כדי "לשערך" את הערכים של DATA החובי  $z$ , לפי התצפית שלהם (ה- $x$ ) – (observed data point) – עבור כל אחת מנקודות DATA נשאל האם הניחוש שעשאה מותאים?
- השתמש בDATA ה-"שלם" כדי לעדכן את כל הפרמטרים (וגם של  $z$  וגם של  $z$ ). נמשיך בתהליכי זה עד שנגיע להתקנסות.

### בחזרה לדוגמה שלנו עם 2 המטבעות

1. **натחל**  $W_A = 0.5, P_B = 0.5, P_a = 0.6, P_b = 0.4$ .
2. **מחשב את ה-** responsibilities – נחשב עבור כל אינסנס, כל ניסוי של 10 הטעויות, נחשב את הפוסטוריור: נציב את הניחוש שלנו בהתקפות ביןימית בהינתן שאנו יודעים איזה מטבע נזרק, לכן יש שני חישובים לכל ניסוי:

$$P_A(x_i) = w_A \binom{10}{9} 0.6^9 0.4^1 = 0.04$$

$$P_B(x_i) = w_B \binom{10}{9} 0.5^9 0.5^1 = 0.01$$

עבור השורה -  
חישוב לדוגמה

чисוב זה הוא "כמעט הפוסטוריור" חסר כאן החלקה באוידיינט, המכנה. המכנים יהיו שונים لكن נחלק במכנים, לומר נחלק את

$$r(x_i; A) = \frac{0.04}{0.05} = 0.8$$

$$r(x_i; B) = \frac{0.01}{0.05} = 0.2$$

התוצאות בסכום של שתי ההסתברויות שקיבלו בדוגמה ובכך קיבל את הפוסטוריור האמיתי –

.3 נציג את התוצאות בטבלת Responsibilities העמודה השמאלית מייצגת את מטבע A, והימנית את מטבע B.

	HHHHHTHHHHH	0.8	0.2
--	-------------	-----	-----

טבלת ה-responsibilities מတרכת כמה הטללה החושבת שהיא הגעה מטבע A, וכמה היא החושבת שהיא הגעה מטבע B. הדבר הגיוני מפני שהענוקנו עבור מטבע A הסתברות גבוהה יותר ל-H מאשר מטבע B. לכן האחוריות באויה השורה תמיד תסכום ל-1.

משיק למלא את הטבלה באותו האופן המתואר בסעיף 2 וכן את הטבלה הבאה: .4

וחתקבל ווקטור ה-responsibilities של A ווקטור ה-B.

מהווקטורים שהתקבלו מסתמן שהסבירות לשני המטבעות אינם חצי-חci כפי שניחשו את ה-W-ים ויש נטייה יותר ל-A.

#### 5. לבן, בשלב הבא יהיה עליינו לעדכן את ה-W-ים. העדכו מتبצע כך:

ניקח את המוצע המשוקל של וקטורי האחוריות של כל מטבע ונחפוץ את התוצאה ל-W החדש של כל מטבע: מימיון הנוסחה הכללית ומשמאלו החישוב המתאים לדוגמה שלנו.

$$\text{New } w_A = \frac{1}{8} \sum_{i=1}^8 r(x_i, A) = 0.65$$

$$\text{New } w_B = \frac{1}{8} \sum_{i=1}^8 r(x_i, B) = 0.35$$

$$\text{New } w_A = \frac{1}{N} \sum_{i=1}^N r(x_i, A)$$

$$\text{New } w_B = \frac{1}{N} \sum_{i=1}^N r(x_i, B)$$

	HHHHHTHHHHH	0.8	0.2
	THHHHHHHHTH	0.76	0.24
	HHHHHHHHHTHH	0.8	0.2
	HHHTHTHHHTT	0.45	0.55
	HHTHRRHHHTH	0.76	0.24
	HTTHHTHHHTT	0.45	0.55
	HTHHHTHHHHT	0.55	0.45
	HTHHHTHHHHHT	0.64	0.36

Init  $p_A = 0.6$   
 $p_B = 0.5$   
ws are 0.5

Coin A responsibilities  
Coin B responsi

#### לאחר עדכון ה-W-ים, עליינו לעדכן את Pa ו-Pb. העדכו מتبצע באופן הבא:

אנחנו עושים סוג של voting, כל אחד מהניסיונות קיבל "צביע" ציון אחריות, כמו הוא חושב שהוא הגיע ממטבע A וכמה הוא חושב שהוא הגיע ממטבע B, לכן נבצע voting ממושך; נחשב את ה-P = MLE = הפוטריר לכל ניסוי, אם ניסוי 1 הניב 9 heads הוא יחשב ש-P של המטבע שנבחר הינו 0.9 (9 לחלק 10). ונעניק לתוצאה זו משקל על פי כמה ניסוי 1 חושב שהוא הגיע ממטבע A, ככלمر 0.8, וביחסו השני משקל 0.2. החישוב המלא של ה-MLE מופיע בוקטור האדום המופיע ב-(i).

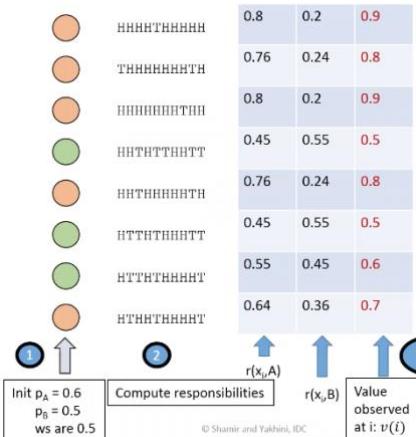
על כן החישוב הכללי יהיה מימין, והחישוב עבור הדוגמה שלנו מופיע משמאלו:

$$p_A = \frac{1}{(New w_A)N} \sum_{i=1}^N r(x_i, A)v(i)$$

$$p_B = \frac{1}{(New w_B)N} \sum_{i=1}^N r(x_i, B)v(i)$$

$$p_A = \frac{1}{5.2} \sum_{i=1}^8 r(x_i, A)v(i) = 0.745$$

$$p_B = \frac{1}{3.8} \sum_{i=1}^8 r(x_i, B)v(i) = 0.48$$



#### cut, יש לנו Ws חדשים, Pb-Pa.

על כן נחזיר על כל התהיליך החל מסעיף 2 עד התוכנות (לא הגדרנו באופן רשמי תנאי עצירה).

באיטרציה הבהה והקטור האדום לא ישנה, והוא למעשה לשנה לא ישנה אף איטרציה! זאת מושם שהחישוב

The EM algorithm for two coins

mbous על הדגימות שלנו בלבד, שאין משתנות. שאר העמודות לעומת זאת

אכן ישנו, וכך ימשיכו להשתנות ממשיק "לעבוד".

$w_A = 0.65$
$w_B = 0.35$
$p_A = 0.745$
$p_B = 0.48$

- Consider a set of starting parameters, including the parameters of Z
- Use these to "estimate" the values of the missing data, per observed data point.
  - + Compute responsibilities using MAP (using the current ws as priors)
- Use the "complete" data to update all parameters (of both Z and X|Z)

$$\text{New } w_A = \frac{1}{N} \sum_{i=1}^N r(x_i, A)$$

$$p_A = \frac{1}{(New w_A)N} \sum_{i=1}^N r(x_i, A)v(i)$$

$$\text{New } w_B = \frac{1}{N} \sum_{i=1}^N r(x_i, B)$$

$$p_B = \frac{1}{(New w_B)N} \sum_{i=1}^N r(x_i, B)v(i)$$

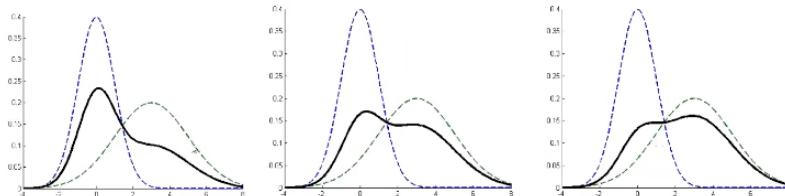
## Gaussian Mixture Models

$$f(x) = \sum_{i=1}^k w_i f_i(x)$$

**וגדר:**  $X$  הוא משתנה מקרי Gaussian Mixture (תערובת גאוסיאנית) אם פונקציית הצפיפות של ההתקלות של  $X$  הינה:  
 $k$  מוגדר מראש – למשל תערובת גאוסיאנית עם  $k=5$ ,  $w_i$  הם משקלים עבור  $i$  פונקציות, כך שסכום ה- $w_i$ -ים סוכם  $1$ .

$$f_i(x) = \frac{1}{\sigma_i \sqrt{2\pi}} e^{-(x-\mu_i)^2/2\sigma_i^2}$$

כאשר לכל  $i$  פונקציות הצפיפות היא פונקציית הגaussiana (לכן כל אחד מהם הוא גאוסיאן):



להלן דוגמאות של השפעת המשקלים  $w_i$  על פונקציית התערובת הגאוסיאנית (בשחור) בהינתן שיש בידינו שני גאוסיאנים  $k=2$  (ירוק וכחול)

## EM for GMMs

### שלב 1 : (E-step) Expectation

נשען את ה-"responsibilities" עבור כל נקודת DATA  $x_i$ , לכל גאוסיאן ( $k$  גאוסיאנים), על ידי שימוש בפרמטרים הנוכחיים (מה פוטורי של כל הגאוסיאנים בהינתן נקודת הדadataה זו).

### שלב 2 : (M-step) Maximization

נשען מחדש את הפרמטרים (המשקלים  $W_k$ , התוחלות  $\mu_k$ -ים, וסטיות התקן  $\sigma_k$ -ות) בעזרת ה-*responsibilities* הקיימים.  
 ככלומר – כל נקודת DATA,  $x$ , תורמת לפרמטרים של כל קומפוננט (מרכז) בגאוסיאן,  $G_k$ ,יחס לרסתונסיביליטי שלו:  $r(x, G_k)$ .

אם יש לנו 4 גאוסיאנים, יש לנו 11 פרמטרים: 2 של כל גאוסיאן (8 סה"כ), והסתברות לקבל כל אחד מהם (4), אבל הריבוע משלים את השלושה  $-1, 0, 1$ , ולכן סה"כ צריך 3 הסתברויות).

1. לאחר אתחול הפרמטרים (ניחוש) נבצע חישוב ב-*responsibilities* שהו למעשה הפוטורי של גאוסיאן  $k$  בהינתן  $x$  (נקודת DATA)

$$r(x, k) = \frac{w_k N(x | \mu_k, \sigma_k)}{\sum_{j=1}^K w_j N(x | \mu_j, \sigma_j)}$$

עדץ על ידי הנוסחה הבאה:

N הוא סימון להסתברות של (במונה)  $x$  בהינתן הגaussiano  $k$ , שהוא בעל הפרמטרים מיום  $k$  וSigma $_k$ . ומכיון שהוא פוטורי עלינו לחלק בסכום הממושקל של ההסתברויות N של דגימת הדadataה x בהינתן גaussiano j עבור j מ-1 עד k.

$$\text{New } w_j = \frac{1}{N} \sum_{i=1}^N r(x_i, j)$$

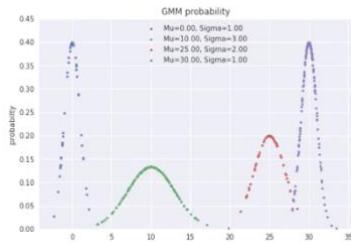
עדכן את ה- $w$ -ים :

עדכן את ה- $\mu$ -ים של כל גaussiano  $k$ :

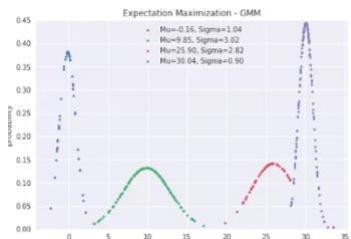
$$\text{New } \mu_k = \frac{1}{(\text{New } w_k)N} \sum_{i=1}^N r(x_i, k) x_i$$

$$(\text{New } \sigma_k)^2 = \frac{1}{(\text{New } w_k)N} \sum_{i=1}^N r(x_i, k) (x_i - \text{New } \mu_k)^2$$

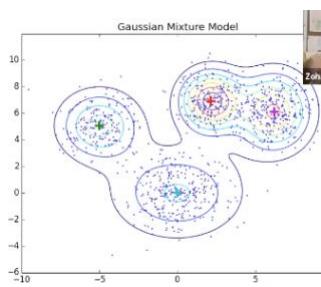
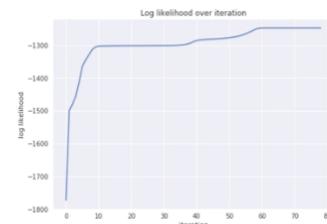
עדכן את הסיגומות של כל גaussiano  $k$ :



**דוגמת הרצה:** ייצרנו 4 גאוסיאנים עם פרIOR ( $w$ ) 0.25 לכל גאוסיאן, ונרצה לנסות לחזות אותם על פי נקודות שהגרלו מותוכן. נרצה לדעת מאייז גאוסיאן הגיעו הנקודות הללו (במציאות אנחנו לא רואים את הצבעים אלא רק את הנקודות). למעלה מופיע גרף המתאר את הדאטה של מודנו, ולמטה זו תוצאה של למידת EM על אחר 80 איטרציות, ניתן לראות שהגענו לשערוך דומה של ערכי הסיגמאות וה-טונותים של הגאוסיאנים שיצרנו מლכתה הIGIN (הגענו למספרים דומים מאוד).



נשים לב שהപסקנו להריץ לאחר 80 איטרציות מכיוון שניתן לראות, שביחסית התכנסות של פונקציית log-likelihood כי השינוי החל להיעשות בלתי ניטן להבחנה סיבוב 80- ~ולහן גרען המתאר את שינוי ההתנהגות של פונקציית loglikelihood לאורך האיטרציות השונות של האלגוריתם.



פונקציית הצפיפות של תערובת גאוסיאנית  $\hat{p}$  ממדיית היא מהצורה: כאשר כל  $i$  היא פונקציית צפיפות גאוסיאנית  $p$ -ממדית.

**כמה פרמטרים יהיה علينا ללמוד במקורה של תערובת גאוסיאנית רב ממדית?**

- $K$  משקלות –  $w$  (למעשה  $1-K$  מפני שהאחרון משלים לו  $1-w$ )
- $d$  תוחלות לכל גאוסיאן.
- מספר הערכים שנצחρיך ללמידה עבור ערכי מטריצת השונות המשותפת יהיה:

$$\begin{aligned} & \binom{d+1}{2} \text{ matrix entries:} \\ & d \text{ variances} + \binom{d}{2} \text{ covariances per Gaussian} \end{aligned}$$

### הערות על האלגוריתם של EM

אחד השימושים של EM הוא עבור Clustering (נחוור לנושה זה בהמשך).

EM לא מחייב את מספר הקומponentות של המודל שאנו לומדים, בנוסף לא מבטיח לנו מקסימום גלובלי (כמו כל תחילה איטרטיבי אחר למשה...), לא מעניק ביטחון שגיאו לאופטימום – למורדות שברתקיטה כו גיאו קרוב לשם. לא תמיד יהיה נתנו לנו הניסחאות המתמטיות עבור מקרה ונכטרך לפתח נוסחה עבור מקרה ספציפי בשביל להפעיל עליו EM.

### יתרונות של EM

התכונות: בכל איטרציה של האלגוריתם, ה-likelihood משתפר מהאיטרציה הקודמת.  
EM מתאים עבור (רובה) משפחות המודלים ועבור כל מספר של פרמטרים.

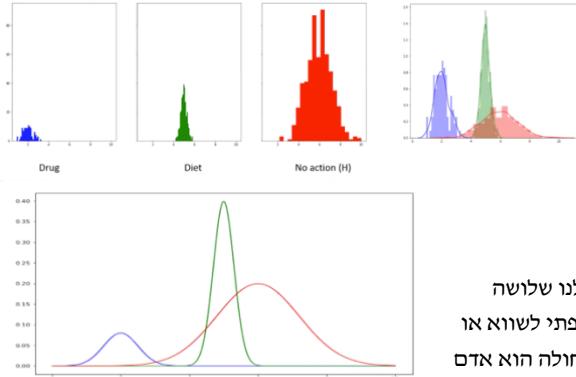
### חסרונות של EM

התכונות יכולות להיות איטריה מאוד עבור דגימות מסוימות וכן ההתכנסות תלויות מאוד בנסיבות המידע החסר. כמו כל גישת למידה, אנחנו עובדים על training data. ומאוד חשוב להימנע מ-overfitting (למשל מספר הערכים של  $2$ ). שערוך מודלים הם תלויים משימושם, ולא נקבעים על פי עקרון. אין הבטחה ל-global optimum (תיכון היתקעות על מקסימום מקומי למשל). הערכים ההתחלתיים (הניחוש) חשובים ונדרשת הפעלה על מספר "ניחושים" או קבוצת "ניחושים" כדי להגיע להתכנסות מותאמת.

## Variations on Bayes Classifiers

### פונקציית מחיר כליליות

עד כה למדנו רק על פונקציית  $0/1$  loss. נניח שיש צוות שפתח בדיקה, עבור סוג שלמחלה, המניבת מספר. באוכטוסייה שלנו **יש אנשים בריאם** שלא צריכים התערבות, **יש אנשים שאם עושים דיאטה יוכל להחליפו**, ויש אנשים שיחליפו על ידי **טיפול רפואי**. נרצה לפתח מסובג בייסיאני אשר מקבל דוגמה (אדם מהאוכטוסייה) וידיע להגיד האם הוא בריא או צריך דיאטה או צריך טיפול רפואי. ייצור עס 100 דוגמאות מ투ך התפלגות המיצגת אנשים שזוקרים לטיפול רפואי, 300 מ투ך התפלגות המיצגת אנשים הזוקרים לדיאטה, 600 מ투ך התפלגות המיצגת בריאם. מכאן שאנו כבר מושיקים מהו הפרIOR עבר כל קבוצה: 0.1 עבור דיאטה, 0.3 עבור טיפול רפואי, 0.6 עבור בריאם.



לhn הדעתה שהוגלה וכן ההיסטוגרמה של ה-class-conditional בהינתן אדם, נרצה לדעת לאן הוא שייך: לגרף האדים (בריא), לגרף הירוק (דיאטה) או לגרף החול (טיפול רפואי). לשם כך נחשב פוטסיטוריום, את הפרIORים כבר יש לי, מה שחרס לנו הוא ה-likelihoods. נוכל להפעיל MLE של גאוסיאנים על מנת למצוא את הפוטסיטוריום. ולאחר מכן נרצה להשתמש במסובג בייסיאני.

### Bayes Classification using a loss function

על רקע הדוגמה, כਮון שפונקציית מחיר  $0/1$  לא ממש מספקת "להעניש" על שגיאה מפני שיש לנו שלושה קלאסים, ובאופן גס פחות תיאורטי – לא נרצה לסוגו מישחו חוליה כבריא ולהעניק לו טיפול רפואי לשווה או שמא לגורם לו לעשות דיאטה למזרות שהוא בריא, וכן גס הדבר אותו שווה ערך לשערך כי אדם חוליה הוא אדם בריא! לכן נרצה להשתמש בפונקציית מחיר כללית יותר.

**נדיר פונקציית loss = המחיר עבור החלטה שגויה:**

נניח כי יש לנו  $k$  קלאסים, כך שלכל  $A_i$ ,  $i \in 1-k$ . על פי התבוננות בדוגמה  $x$  עלינו להחליט לאן דוגמיה זו משתייכת מבין הקלאסים  $A_i$  (על ידי הפעלת גישת בייס או MAP). החלטה שגויה מובילה להפסד loss! הפסד תלוי באיזה  $j$  סוג באופן שגוי ל- $i$ . נציג זאת ע"י פונקציית מחיר:

$$\lambda_{ij} = \begin{cases} 0 & i=j \\ 1 & i \neq j \end{cases} \text{ (למשל, ניתן את פונקציית loss } 0/1 : \text{ Cost}(h(x) = A_i \wedge x \in A_j)$$

$$R(\text{Choose } A_i | x) = \sum_{j \neq i} \lambda_{ij} P(A_j | x)$$

נרצה לחשב את הסיכון בפונקציית loss כללית:

המסוג הבייסיאני הוא זו שمبיא לMINIMUM את מחיר השגיאה. כמובן, באופן כללי, בהינתן דוגמיה  $x$  נסוג אותה לערך:

$$C(x) = \operatorname{argmin}_i \sum_{j \neq i} \lambda_{ij} P(A_j | x)$$

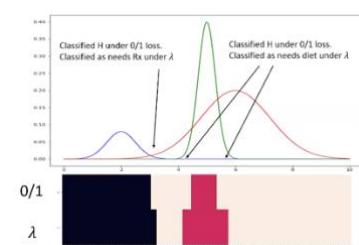
ההבדל הוא למעשה לא באלגוריתם הלמידה זהו אותו האלגוריתם שלמדו כבר, אלא באלגוריתם המבצע שיבצע ויסוג באופן שונה, מפני שכעת אנחנו מתייחסים לפונקציית מחיר שונה. בגלל שהמחיר שונה – ההחלטה תהיה שונה.

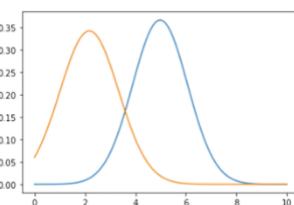
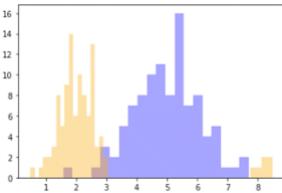
כדי להשיג את ה-risk, expected risk, לכל נקודה בDATA 101 נקודות DATA חדשה במקורה של הקוד ( $x$ ) יהיו 3 ערכים ל-3 הקלאסים, הקלאס שקיבלאת הערך סיכון הנמוך ביותר הוא הקלאס אליו נסוג את הדוגמיה. על ידי הכפלת המטריצות הניל נקבל את מטריצת risk-ים.

**דוגמה לחישוב הסיכון:** להלן מחיר "האמת" מול הפרידקציה, כך שערך האלבסטון מייצגים סיווג נכון (אמת = פרידקציה) ועל כן המחיר הוא 0. שורה 0 ומودה 1 מותארת את המחיר שעילה לסוגו אדם שזוקק ב"אמת" לדיאטה,adam שזוקק טיפול רפואי = המחיר הוא 100. אנחנו אלה שקובעים את המטריצה הזו ואיתה עושים את הפרידקציה.

$$\lambda = \begin{pmatrix} \text{TRUE} & & \\ Rx & Diet & H \\ \hline 0 & 100 & 300 \\ 800 & 0 & 100 \\ 1000 & 800 & 0 \end{pmatrix}$$

להלן התוצאות של האלגוריתם סיווג מצד אחד תחת loss  $0/1$  לעומת התוצאות של האלגוריתם תחת מדיא. ניתן לראות כי מיהו שמחירו לסוג לא נכון אדם שמיועד לטיפול רפואי כדמים בריא והוא מאוד גבוה (1000) אזו לעומת סיווג  $0/1$ , הסיווג "עדין יותר/מצויץ יותר" מאשר loss  $0/1$ .





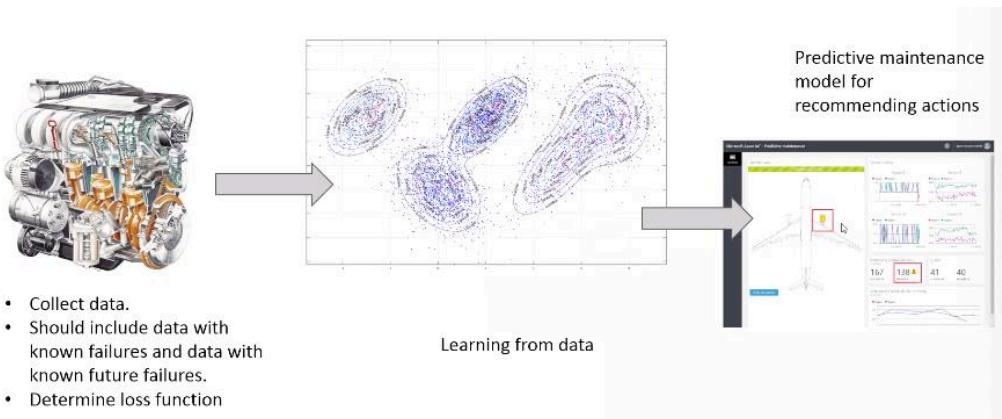
**GMM Bayes**  
נניח כי יש לנו שני קלאסים A ו-B וליהלן ה training data עבורם עם prior חצי-חצי. ונרצה לבנות מסוג בייסיאני על הדadata זהה, لكن הדבר הראשון שנעשה הוא Gauss MLE על הדadata במטרה לקבל את ה-posteriors של הדadata ומכיון שהפרויורים שוים ניתן לסוג על ידי ה-likelihood.

אבל אם במקום גaus MLE נלמד מודל תערובת גaussיאנים על ידי EM נקבל את הסיווג הבא (class-conditionals) שיכובן ישב סיווג טוב יותר.

כפי שניתן לראות בהשוואה המסווגת הבאה (סיווג שגורר מול כותם)

התוצאות זו עשויה להתרחש גם  
בממדים שאינם ייחודיים (גם בדו ממדים  
וכדומה)

הערה כללית עבור למידה כזו במטוסים: (למידה חישובית ומסוגים מסווג זה בעולם האמיתי)



## נאייב ביאס ו-ME – 4 תרגול

### Naïve Bayes

לכורה האלגוריתם שלמדונו עובד כמו שצרכיך. אבל, למה שהוא יעבוד, אם השערוך של likelihood הוא לא טוב – אז, עניינו על כך בהערכתה – הקלאס שמניב הסתברות מקסימלית בשני המקרים, הוא אותו הקלאס. כך שלמעשה אנחנו צרכיכים:

$$\operatorname{argmax}_i P(A_i) \prod_{j=1}^d P(x_j|A_i) = \operatorname{argmax}_i P(A_i) P(x_1, x_2, \dots, x_d|A_i)$$

### Discrete Naïve Bayes

$$P(x_j|A_i) = \frac{n_{ij}}{n_i}$$

כעת נראה דוגמה לחישוב ה-likelihood ע"י ספירה של הדאטה:

- $n_{ij}$  – הוא מספר הדגימות בדאטה אימונו עם הקלאס  $A_i$  והערך  $x_j$  באטריבואט הרלוונטי
- $n_i$  – הוא מספר הדגימות בדאטה אימונו עם הקלאס  $A_i$  מכלים ולכך

נשאלת השאלה מה הבעה כאן? יכול להיות שבעת ה-*training set*-הוינון קלאס לא בטוח שככל ערכי האטריבואטים הם מלאים ולכך הסתברות likelihood 0 (בגלל שאחננו מכפילים) למקרים האמתניים והסתברות 0 לא מתרחשת. אסור לנו להניח שם שלא ראיינו (בדאטה אימונו שלנו), לא מותקינים. לכן, כדי לפתרור את בעיה זו ולתת איזון ונשתמש ב- Laplace estimation .

### Laplace estimation – תיקון לפלאס

$$P(x_j|A_i) = \frac{n_{ij} + 1}{n_i + |V_j|}$$

זה תיקון מאוד פשוט אשר מוסיף אחד במונה ואת מספר הערכים של האטריבואט במכנה. כאשר:

- $n_{ij}$  – הוא מספר הדגימות ב-data-עם הקלאס  $A_i$  והערך  $x_j$  באטריבואט הרלוונטי.
- $n_i$  – הוא מספר הדגימות ב-data-עם קלאס  $A_i$ .
- $|V_j|$  – הוא מספר הערכים האפשריים של האטריבואט הרלוונטי.

העיקרון נשמר בתיקו זה, הוא שיישארו כאותן הסתברויות בין 0-1 אשר נסכמאות ל-1, אך זו נוסחה valid –ית להסתברות.

**דוגמה:** נניח שנרצה לסוג בין שני טיפולים לחולים = קלאסים A ו-B. יש לנו מטופלים בעלי history data אשר מכיל 4 אטריבואטים: מין, לחץ דם, גיל והטיפול שהמטופל קיבל. נרצה לסוג מטופל חדש עם איבר מסוים. להלן הדאטה שלנו וכן החישובים הכלולים את תיקון לפלאס.

Gender	Blood Pressure	Age	Treatment
Male	Normal	Young	A
Male	High	Old	A
Male	High	Old	A
Female	High	Young	A
Female	Normal	Young	A
Female	High	Old	A
Male	Low	Young	B
Male	Low	Old	B
Male	Normal	Old	B
Female	Low	Young	B
Female	Normal	Old	B
Female	Normal	Old	B

$P(A) = \frac{6}{12} = \frac{1}{2}$	
$P(\text{male} A) = \frac{3+1}{6+2} = \frac{4}{8} = \frac{1}{2}$	$P(\text{female} A) = \frac{3+1}{6+2} = \frac{4}{8} = \frac{1}{2}$
$P(\text{high} A) = \frac{4+1}{6+3} = \frac{5}{9}$	$P(\text{normal} A) = \frac{2+1}{6+3} = \frac{3}{9}$
$P(\text{low} A) = \frac{0+1}{6+3} = \frac{1}{9}$	
$P(\text{young} A) = \frac{3+1}{6+2} = \frac{4}{8} = \frac{1}{2}$	$P(\text{old} A) = \frac{3+1}{6+2} = \frac{4}{8} = \frac{1}{2}$
$P(B) = \frac{6}{12} = \frac{1}{2}$	
$P(\text{male} B) = \frac{3+1}{6+2} = \frac{4}{8} = \frac{1}{2}$	$P(\text{female} B) = \frac{3+1}{6+2} = \frac{4}{8} = \frac{1}{2}$
$P(\text{high} B) = \frac{0+1}{6+3} = \frac{1}{9}$	$P(\text{normal} B) = \frac{3+1}{6+3} = \frac{4}{9}$
$P(\text{low} B) = \frac{3+1}{6+3} = \frac{4}{9}$	
$P(\text{young} B) = \frac{2+1}{6+2} = \frac{3}{8}$	$P(\text{old} B) = \frac{4+1}{6+2} = \frac{5}{8}$

Gender	Blood Pressure	Age	Treatment
Male	Normal	Young	A
Male	High	Old	A
Male	High	Old	A
Female	High	Young	A
Female	Normal	Young	A
Female	High	Old	A
Male	Low	Young	B
Male	Low	Old	B
Male	Normal	Old	B
Female	Low	Young	B
Female	Normal	Old	B
Female	Normal	Old	B

לכן למשל עבור דוגמה חדשה / מטופל חדש שהוא גבר, צעיר, ובעל לחץ דם גבוה. נחשב (כנראה שבשורה הראשונה יש טעות בשווין האחרון) :

*male, young, high*

- $P(A|\text{male, young, high}) = P(A) \cdot P(\text{male}|A) \cdot P(\text{young}|A) \cdot P(\text{high}|A) = \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{5}{9} = \frac{5}{54}$
- $P(B|\text{male, young, high}) = P(B) \cdot P(\text{male}|B) \cdot P(\text{young}|B) \cdot P(\text{high}|B) = \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{3}{8} \cdot \frac{1}{9} = \frac{3}{288}$

After normalization:

$$\bullet P(A|\text{male, young, high}) = \frac{\frac{5}{54}}{\frac{5}{54} + \frac{3}{288}} = 0.9 \quad P(B|\text{male, young, high}) = \frac{\frac{3}{288}}{\frac{5}{54} + \frac{3}{288}} = 0.1$$

ונקבל כי ההסתברות שהאדם יgive יותר טוב לטיפול A היא גבוהה יותר ועל כן נסוגג אותו לטיפול A.

## אלגוריתם EM

אלגוריתם איטרטיבי שבני שני שלבים Expectation ו-Maximization. נסמן ב-D את קבוצת נקודות הדadata שלנו (observed data), טטה יהיה וקטור הפרמטרים שלנו שנרצה שיניב עבורנו ML = maximum likelihood =  $\text{maximum likelihood}$ , כלומר אותו נחשף. מושתמשים ב-EM כאשר אנחנו רוצים לחשב את  $P(x|z|\theta)$ , אבל חישוב כזה באופן ישיר הוא לא פשוט. חישוב  $P(x,z|\theta)$  הינו פשוט יותר, כאשר  $z$  הינו איזשהו דאטה חבוי hidden data. אנחנו מניחים שהדadata החבוי נקבע ע"י משתנה מקרי כלשהו  $Z$ , שהוא חלק מהמודל. הערכה: המודל, תחת הוקטור טטה, שולט גם ב-X וגם ב-Z. ואבל ב-D אנחנו נראה רק את הערכים של X.

(כאן ניתנה הדוגמה שניתנה גם בהרצאה מס' 5 – עם ההטלה של 2 המטריות)

$$p_A = \frac{1}{(New w_A)N} \sum_{i=1}^N r(x_i, A)v(i)$$

הערה לגבי דוגמה זו שעלתה בתרגול – נכון ה-P-ים מכפיל את הסכום ב-Responsibilities, כלומר בנוסחה זו :

מה שנמצא מוחץ לSigma 1 (חלק W\_A החדש כפול N), הוא למעשה חילוק בסכום ה-Responsibilities של A ונitinן לראות זאת בנוסחה הבאה:

$$New w_A = \frac{1}{N} \sum_{i=1}^N r(x_i, A)$$

שבה אם מכפילים ב-N גודלה את שני האגפים נקבל מצד שמאל את המכנה שמחשב את  $p_A$  ומצד ימימן את סכום ה-

. כאמור, A, responsibilities של A.

## EM for GMMs

### .1. שלב 1 : (E-step) Expectation

נשערק את ה-"responsibilities" של כל נקודה דאטה לכל גאוסיאן באמצעות הפרמטרים הנוכחיים.

### .2. שלב 2 : (M-step) Maximization

נשערק מחדש את הפרמטרים ( $w$ -ים, מיואים, סיגומות) בעזרת ה-"responsibilities" הקיימים. ככלומר – כל נקודה דאטה, x, תורמת

לכל מרכיב גאוסיאן,  $G_i$ , ביחס לאחריות שהיא קיבלה:  $r(x, G_i)$

- Responsibilities:

$$r(x, k) = \frac{w_k N(x|\mu_k, \sigma_k)}{\sum_{j=1}^K w_j N(x|\mu_j, \sigma_j)}$$

כאשר הנוסחאות עבור אלגוריתם זה הין:

- Weights:

$$New w_j = \frac{1}{N} \sum_{i=1}^N r(x_i, j)$$

- Mean:

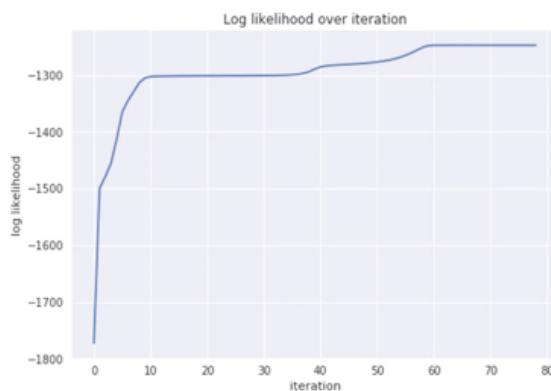
$$New \mu_j = \frac{1}{(New w_j)N} \sum_{i=1}^N r(x_i, j) x_i$$

- Variance:

$$(New \sigma_j)^2 = \frac{1}{(New w_j)N} \sum_{i=1}^N r(x_i, j) (x_i - New \mu_j)^2$$

(ניתנה דוגמת ההרצתה של 4 הגאוסיאנים שראינו בהרצאה)

נארה הערכה לגבי הגרף שמתאר את **המיקסום של הלוג-לייקלhood** לאורך האיטרציות, יש איזושהי התכנסות שאינה גלובלית בהרצתה מוקדמות 10-40 ואז נשזה עוד שיפור בין האיטרציות 60-40 ובסביבות האיטרציה ה-80-85 כבר נגיע להתקנסות.



## כטביה חישובית ממחזור הרצאה 6

### Logistic Regression (LOR)

זהו אלגוריתם קלסיפיקציה, בוגר לוגיסטי ליניארית אשר מניב ערכים רציפים לכל דגימה, אלגוריתם זה מניב סיווג בלבד. כמו בכל מסוג,

$$\{(\vec{x}^{(i)}, y^{(i)})\}_{i=1..m} : \text{labels}$$

הดาטה אימנו במקורה הביטרי, מכילה וקטורי פיצרים ואת הליבלים המותאים (labels)  $h: \mathbb{R}^k \rightarrow \mathbb{R}$

אנחנו נלמד מודל שהוא פונקציה  $h$  המכבלת וקטור פיצרים  $x$  (בממד  $k$  למשל) כך ש- $h$  מקיימות:

מודל logistic regression חושב לשיטות למידה מודרניות יותר וכן עבור deep learning.

אנחנו נרצה שהפונקציה שמנצאת,  $h$ , תחזיר עבורו הסתברויות כך שלמעשה היא תחזיר ערכים מהקטע  $[0, 1]$  ולאו דווקא מ-R.

#### ב-**logistic regression** אנחנו מניחים את התנהלה המרכזית הבאה:

נניח כי ההסתברות ( $X$  |  $Y$ )  $P$  (הסתברות להיות מסויך ליביל מסויים  $Y$  בהינתן דגימה  $X$ ) יכולה להיות משוערת כפונקציית sigmoid

המיושמת על קומבינציה ליניארית על ה- $h$ -input features. באופן מתמטי, עבור נקודת DATA ( $\vec{x}$ ), אנו מניחים ב-LoR כי:

$$z = \theta_0 + \sum_{j=1}^k \theta_j x_j \quad \sigma(z) = \frac{1}{1 + e^{-z}} \quad (\text{זוהי פונקציית sigmoid})$$

(קומבינציה ליניארית של וקטור  $x$  והוקטור  $\theta$  המקורי)

לכן  $z$  הוא למעשה סקלאר, מספר ממשי, מפני שיש לנו מכפלת פנימית של הווקטור טטה עם הווקטור  $x$ .

ההצבה היא להלן, כאשר השווון השני מתקיים מפני שאחננו מכפילים מונחים ומכוון ב- $(x^T \theta)^*$  (theta transpose).

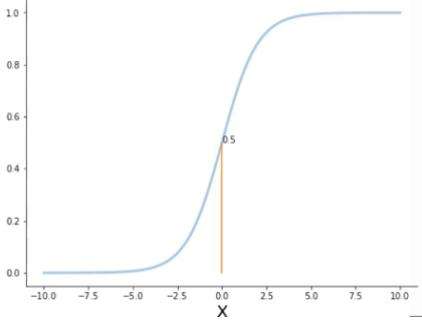
• הפונקציה הזו מונוטונית עולה ממש

$$\text{הטווה של הפונקציה הוא } [0, 1], \text{ הפונקציה ראשית כל חיובית, המכנה גדול מהמונה ולכל}$$

$$\sigma(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}} = \frac{e^{\theta^T x}}{1 + e^{\theta^T x}}$$

קטינה מ-1.

• הגבולות באינסוף: 0 במינוס אינסוף ו-1 באינסוף.



Derivative:  
 $\sigma'(x) = \sigma(x)(1 - \sigma(x))$

תמונה נוספת של פונקציית סיגמוד שיכולה לסייע הינה:

כאן, שנוכל להפוך את פונקציית ההיפותזה שלנו / המודל שלנו לצורה הבאה:

$$h_\theta(\vec{x}) = P(y = 1 | \vec{x}) = \sigma(\theta^T \vec{x})$$

מכפלת פנימית של טטה עם הרוחה של הווקטור איקס בתוך הפונקציה של סיגמוד. כאשר השיוויון logistic regression

השני נובע מנהנת ה-m

if  $h_\theta(\vec{x}) > 0.5$  then 1

else 0

כasher הפרדיקציה / הקלסיפיקציה תערך באופן הבא:

עלינו למלוד את הטtotות שמניבו את הניבוי הטוב ביותר – נקבע תהליך למידה עבורן. טטה הוא וקטור פרמטרים באורך  $k+1$  (למעט טטה 0), אנחנו נלמד את הערכים של תחת מטרת maximum likelihood. לשם כך נשתמש ב-training data set שנסמכו אותה ב-D שיש בה  $m$  דוגמאות. כל דגימה תכיל וקטור של פיצרים  $x$  מממד  $k$  וערך בינארי  $y$  שנקרא label.

### Maximum Likelihood

ראשית, נשים לב כי תחת התנהלה המרכזית שאנו מניחים ב-LoR, מתקיים עבור נקודת DATA ייחידה:

$$P(y|x; \theta) = (h_\theta(x))^y \cdot (1 - h_\theta(x))^{1-y} \quad \text{או:} \quad P(x, y; \theta) = (h_\theta(x))^y \cdot (1 - h_\theta(x))^{1-y} P(x)$$

אבל זה מתקיים עבור DATA אחת, לכן כתעתüber הדאטה אימון D עם  $m$  דוגמאות שנניח כי הן בלתי תלויות, נרצה למצוא את המודול

שמקסם את ה-likelihood. עבור כל טטה, הליליהוד שלה בהינתן הדאטה,  $P(D|\theta)$ , היא ביחס ל-

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

נרצה למצוא את הטטה שمبיאה זאת למינימום.

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

$$\begin{aligned} \underset{\theta}{\operatorname{argmax}} \prod_{d=1}^m\left(h_{\theta}\left(x^{(d)}\right)\right)^{y^{(d)}} \cdot\left(1-h_{\theta}\left(x^{(d)}\right)\right)^{1-y^{(d)}} = \\ \underset{\theta}{\operatorname{argmax}} \ln \left(\prod_{d=1}^m\left(h_{\theta}\left(x^{(d)}\right)\right)^{y^{(d)}} \cdot\left(1-h_{\theta}\left(x^{(d)}\right)\right)^{1-y^{(d)}}\right) = \\ \underset{\theta}{\operatorname{argmax}} \sum_{d=1}^m \ln \left(\left(h_{\theta}\left(x^{(d)}\right)\right)^{y^{(d)}}\right)+\ln \left(\left(1-h_{\theta}\left(x^{(d)}\right)\right)^{1-y^{(d)}}\right) = \end{aligned}$$

$$\begin{aligned} \underset{\theta}{\operatorname{argmax}} \sum_{d=1}^m y^{(d)} \cdot \ln \left(h_{\theta}\left(x^{(d)}\right)\right)+\left(1-y^{(d)}\right) \cdot \ln \left(1-h_{\theta}\left(x^{(d)}\right)\right) = \\ \underset{\theta}{\operatorname{argmin}} \sum_{d=1}^m-y^{(d)} \cdot \ln \left(h_{\theta}\left(x^{(d)}\right)\right)-\left(1-y^{(d)}\right) \cdot \ln \left(1-h_{\theta}\left(x^{(d)}\right)\right) \end{aligned}$$

מכאן, נוכל להתחיל להריץ גרדיאנט דיסנט מפני שכל מה שכתוב בגרדיינט נמצא בשונו (בנהוג שיש לי נירוש התחלמי לוקטור טה).  
הנגזרת הcyונית לפיטה $\hat{z}$  המוחשבת בנקודה טה שהיא וקטור  $k+1$  ממדי.

**לכן גזר ונשווה ל-0.** השורה הראשונה מועתקת מהעמודה הקודם. בשורה השנייה אנחנו רוצים לבטל מכפלות כדי שיהיה קל יותר לכפול ולכט אנו מפעלים לון (מוניוניות עלה ולכן איננה משנה את המקסימום). בשורה השלישי המכפלה הפכה לסכום והופכל לון נוסף על שתי החזקות כדי שנוכל להוריד את  $y^{(d)}$  מהחזקה – שורה רביעית.  
בשורה החמישית והאחרונה אנחנו הופכים סימן למינוס כדי למצוא את ה- $\text{cost}$ .  
כפי זה ניתן לחשב על זאת במנוחי  $\text{cost}$ .

**לכן נרצה להביא למינימום את :** (למזה של טה)

$$\Lambda(\vec{\theta})=-\sum_{d=1}^m y^{(d)} \ln \left(\sigma\left(\theta, \vec{x}^{(d)}\right)\right)+\left(1-y^{(d)}\right) \ln \left(1-\sigma\left(\vec{\theta}, \vec{x}^{(d)}\right)\right)$$

נzieין כי כאן לא עובד לחשב גרדיאנט ולהשווות ל-0 (חישוב אספליסטי לא יעבד כאן).  
לכן נשתמש בגרדיינט דיסנט.

גובל להראות כי : זהו הגרדיינט (וקטור בגודל  $n+1$ )

$$\frac{\partial}{\partial \theta_j} \Lambda(\vec{\theta})=\sum_{d=1}^m\left(\sigma\left(\vec{\theta}, \vec{x}^{(d)}\right)-y^{(d)}\right) x_j^{(d)}$$

הנגזרת הcyונית לפיטה $\hat{z}$  המוחשבת בנקודה טה שהיא וקטור  $k+1$  ממדי.

מכאן, נוכל להתחיל להריץ גרדיאנט דיסנט מפני שכל מה שכתוב בגרדיינט נמצא בשונו (בנהוג שיש לי נירוש התחלמי לוקטור טה).

### אלגוריתם גרדיאנט דיסנט עבור Logistic Regression

- נאחל את טה להיות ערך התחלמי קטן / נירוש
- נזר על התחלך הבא עד אשר נגיע להתקנסות:

$$\theta_j^{New}=\theta_j^{Old}-\alpha \frac{\partial \Lambda}{\partial \theta_j}\left(\theta^{Old}, X, y\right)$$

מבצע עידכו של הווקטור טה באופן הבא :

$$\theta_j^{New}=\theta_j^{Old}-\alpha \cdot \sum_{d=1}^m\left(\sigma\left(\vec{\theta}, \vec{x}^{(d)}\right)-y^{(d)}\right) x_j^{(d)}$$

או באופן יותר אקספליסטי, אם נציב את הגרדיינט שהראינו לעיל:

כאשר לפחות הוא ה learning rate •

חישוב הגרדיינט של למדא (שכבר ראינו לעיל) – הוספת סיגמויד (שמאל) ולאחר מכן גזירה לפי פיטה $\hat{z}$  (ימין) :

$$\begin{aligned} \Lambda(\vec{\theta}) &=-\sum_{d=1}^m y^{(d)} \ln \left(\sigma\left(\theta, \vec{x}^{(d)}\right)\right)+\left(1-y^{(d)}\right) \ln \left(1-\sigma\left(\vec{\theta}, \vec{x}^{(d)}\right)\right) \\ \ln (\sigma(\vec{\theta}, \vec{x}^{(d)})) &=\ln \left(\frac{1}{1+e^{-\theta^T \vec{x}^{(d)}}}\right)=-\ln \left(1+e^{-\theta^T \vec{x}^{(d)}}\right) \\ \ln (1-\sigma(\vec{\theta}, \vec{x}^{(d)})) &=\ln \left(1-\frac{1}{1+e^{-\theta^T \vec{x}^{(d)}}}\right) \\ &=\ln \left(\frac{1+e^{-\theta^T \vec{x}^{(d)}}}{1+e^{-\theta^T \vec{x}^{(d)}}}-\frac{1}{1+e^{-\theta^T \vec{x}^{(d)}}}\right) \\ &=\ln \left(\frac{e^{-\theta^T \vec{x}^{(d)}}}{1+e^{-\theta^T \vec{x}^{(d)}}}\right) \\ &=-\theta^T \vec{x}^{(d)}-\ln \left(1+e^{-\theta^T \vec{x}^{(d)}}\right) \end{aligned}$$

→  $\frac{\partial}{\partial \theta_j} \Lambda(\vec{\theta})=\sum_{d=1}^m\left(\sigma\left(\vec{\theta}, \vec{x}^{(d)}\right)-y^{(d)}\right) x_j^{(d)}$

### חזרה על עקרונות ה-Logistic Regressions – שלמדו עד כה :

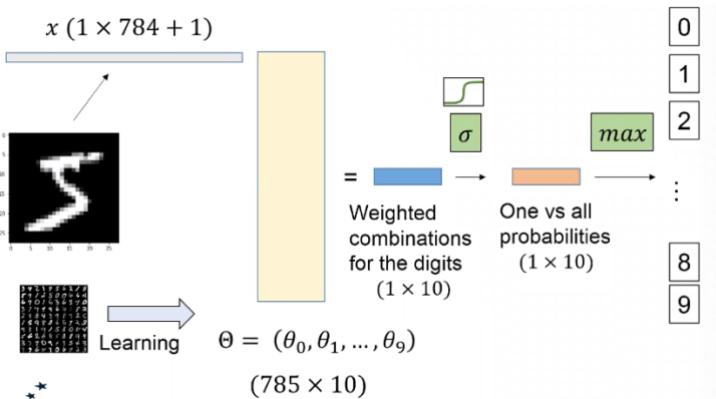
פונקציית ההיפוטזה או המודל, על פי הנחת LoR הינה :  $h_{\theta}(\vec{x})=P(y=1|\vec{x})=\sigma(\theta^T \vec{x})$  •

פונקציית המחיר / השגאה, שובעת מחישוב ההסתברות  $P(\text{D}|\theta)$  הינה :  $\Lambda(\theta)=\frac{1}{m} \sum_{d=1}^m-y^{(d)} \cdot \ln \left(h_{\theta}\left(x^{(d)}\right)\right)-\left(1-y^{(d)}\right) \cdot \ln \left(1-h_{\theta}\left(x^{(d)}\right)\right)$  •

$\underset{\theta}{\operatorname{argmin}} \Lambda(\theta)$  : נשתמש בגרדיינט דיסנט במטרה למצוא את :

**דוגמא:** דата שנקרא MNIST Digits ומשמעו ספורות שכותבים בני אדם. כל ספרה מוצגת בגודל 28x28 פיקסלים ונרצה למצוא מודל שモזה את הספרה הכתובה. אנחנו ניקח כל פיקסל ונשתח אותו לוקטור אחד, **אינפוט וקטור א'**,  $(28 \times 28 = 784)$ , שהוא שורה אחת ו-1 784+1=785. לכל ספרה למדנו מודל = טטה, ששואל את השאלה, האם זה 0 או לא? האם זה 1 או לא? ... כולם כל אחת מהטבות שואלת שאלה בינהו את הספרה שמויה בתמונה היא | לכל בין 0 ל-9 או שלא. لكن קיבלו 10 טבות. התוצאה היא **מטריצה (הצורה)** שגדלה 785 שורות ו-10 עמודות, 785 שורות מפני שכל טטה מקבלת אינפוט שהוא 785. נבע מכפלה פנימית בין הוקטור א' של התמונה ה"משוחחת" לבין המטריצה זו

**ונקבל וקטור בעל 10 ערכים (1x10)** (ונכניס אותו ל-**Tau**)



פונקציית סיגנום (המעבר בין הכהול לכתום) ונקל וקטור  
כתום שהוא גם כן בגודל 1x1 אבל הערכים בו הם בין 0 ל-1,  
אליה הן החסתברויות: כמו המודל חישב שהסיכויים שמדובר  
במספר 0, הסיכויים שמדובר ב-1, הסיכויים שמדובר ב-2 וכו'...  
ומתווקטור זהה ניקח את המיקום בו החסתברות היא  
מקסימלית ולפיכך נסוג את התמונה לקלאס 0 עד 9.  
הוקטור הכתום הוא למעשה מושג כבר ה-**posterior**, האלגוריתם  
 מגיע לשירות ל-**posterior**.

גישה זו נקראת **one vs all**. ומספר הטעות שנלמד הוא  
מספר הקלאסים (אם עליינו ללמידה אוניות אנגליות קטנות  
היו לנו לומדים 26 טבות)

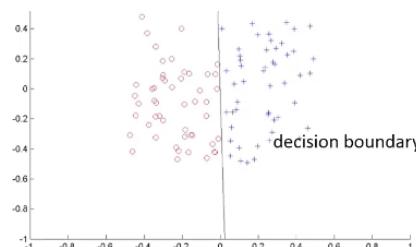
#### לסיכום,

- LoR היא גישה לקלסיפיקציה בעלת שם מיטה (כי ברגסיה לינארית לא ביצענו סיוע).
- האלגוריתם המבוצע (או המודל) מסובב בהתבסס על וקטור בעל פיצרים נומריים.
- $P(y=1|\vec{x}) = \sigma(\theta^T \vec{x})$  הגישה מבוססת על הנחת ה-LoR שהיא
- תחילה הלמידה משתמש בגרדיאנט דיסנט כדי למצוא טטה המניבה מקסימום ליקיליות / מינימום טעות בהינתן ה-data, observed data.
- תחת הנחתה LoR-ה.
- אין אפשרות להשתמש ב-**pseudo inverse** LoR-ב.
- ניתן להעתיק את אלגוריתם זה עבור קלאסים רבים (כפי שראינו בדוגמה לעיל one vs all).

#### – **קלסיפיקציה לינארית** – Linear Classifiers

הערה: מסובב בייסיאני יכול לסובב גם משתנים קטגוריאליים וגם רציפים, LoR

**מפרידים לינאריים** – נניח כי מרחב הדגימות שלנו הוא  $\mathbb{R}^2$  (כלומר 2 פיצרים). כל דוגימה היא נקודה. אם יש לנו המפריד את שני הקלאסים (או קו המפריד בין קונספטים) אז שני הקלאסים הם **נתונים להפרדה לינארית** (או: הkoncept ניון להפרדה לינארית).

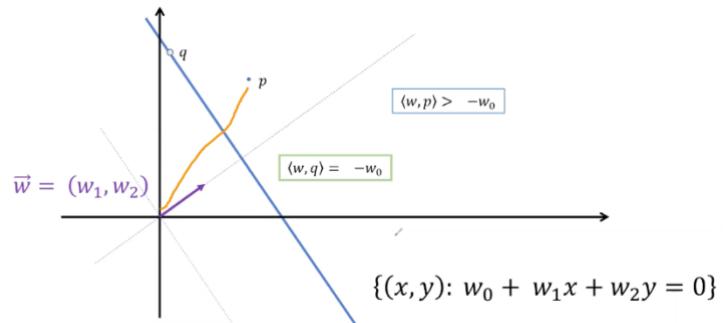
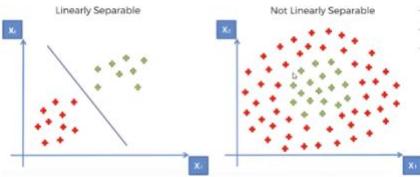


$$\text{ב-2D נציג קו על ידי המשוואת } w_0 + w_1x + w_2y = 0$$

(דיקוטומיה = DATA עם עיגולים ופלוסים כפי שוראים באירור)

$$\begin{aligned} \text{דיקוטומיה ב-2D} & \text{ ניתנת להפרדה ע"י קו אם:} \\ f(x,y) > 0 & \Rightarrow +1, \\ f(x,y) < 0 & \Rightarrow -1 \end{aligned}$$

המשמעות הגיאומטרית עבור המשווה שבה בחרנו לייצג קו ישר:



### Linear Discriminant Functions

$$g(x) = \mathbf{w}^T \mathbf{x} + w_0 = 0$$

ב הכללה כללית יותר, ב- $\mathbb{R}^n$ , כאשר  $(\mathbf{x})$  היא לינארית היא מגדיאה היפר-מיישור ע"י ההשווותה ל-0:  $\mathbf{w}^T \mathbf{x} + w_0 = 0$ .

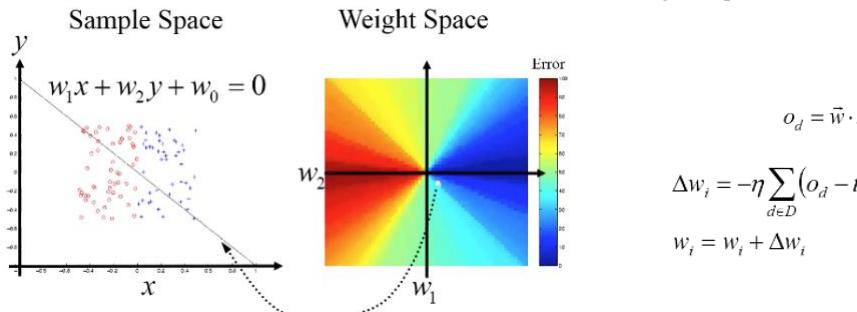
נשים לב כי נוכל לחשב על כך מכפלה פנימית של  $(\mathbf{w}, \mathbf{x})$  עם  $(\mathbf{w}, \mathbf{p})$ . והסיגוג יתבצע באופן הבא:  
בහינתן שקיים דאטה כזה שהוא ניתן להפרדה לינארית, נרצה ללמידה את שיקנה עבורנו מסווג.  
לכן עלינו למשער איזושי פונקציית טעות ולמצוא את ה- $\mathbf{w}$ -ים אשר ממזערים אותה.

$$E(\vec{\mathbf{w}}) = \frac{1}{2} \sum_{d \in D} (\vec{\mathbf{w}} \cdot \vec{\mathbf{x}}_d - t_d)^2$$

נסחה ללמידה את ה- $\mathbf{w}$ -ים אשר מביאים למינימום את השגיאה הריבועית על כל דוגמאות הדאטה ( $D$  הוא הדאטה סט):  
פונקציית הטעות מקבלת וקטור  $\mathbf{w}$  ומה שהיא עשויה להיות היא לסוכם את: המכפלה הפנימית של הוקטור  $\mathbf{w}$  עם וקטור הפיצרים  $\mathbf{x}_d$  פחות הליביל  $t_d$  של הנקודת דאטה זו (זה למשה  $y_d$  וסתם סומן באופן שונה) ביריבוע. זו פונקציה המזקירה סכום ריבועים פחותים. זהו למעשה בדיקת MSE  
לרגรสיה לינארית! ללחנו את שאלת הקלאסיפיקציה של ו ההפכתי אותה לשאלת רגרסיה לינארית. וכן גם השתמש כאן בגרדיינט דיסנט.  
.LMS Classification Algorithm – ומכאן קיבל את האלגוריתם

### LMS Classification Algorithm

סימונים: אטה (נראית כמו זו) תסמן את ה- learning rate שלנו (למשל 0.05), ו-  $D$  יסמן את הסט של דוגמאות ה- training. כל דוגמת אימון הינה זוג מהצורה  $(\mathbf{x}, t)$  כאשר  $\mathbf{x}$  הוא וקטור הפיצרים ו-  $t$  הוא ה- label .target output value



נתחול ערכים עבור וקטור המשקלות  $\mathbf{w}$

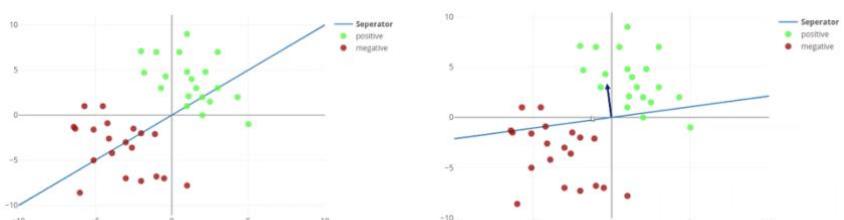
נחזיר עד התכנסות על הצעד הבא:

$$\mathbf{o}_d = \vec{\mathbf{w}} \cdot \vec{\mathbf{x}}_d$$

$$\Delta \mathbf{w}_i = -\eta \sum_{d \in D} (\mathbf{o}_d - t_d) \mathbf{x}_{id}$$

$$\mathbf{w}_i = \mathbf{w}_i + \Delta \mathbf{w}_i$$

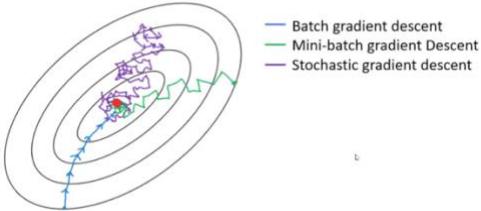
ועבור כל יחידת משקל לינארית  $w_i$ , נבצע:



**דוגמה לתחילת הלמידה:** משמאל הדאטה הייתה עליון

ללמידה (היוקם מול האדומים) וראינו כיצד תהליך הלמידה משנה את כיוון המפריד, כך שהחץ המאונך בדוגמה מיomin (שלב מסוים מהתהליך הלמידה) מיציג את וקטור המשקלים הلينאריים  $\mathbf{w}$ .

## Stochastic Vs. Standard Gradient Dsecent



```
Initialize each  $w_i$  to some small random number.
Until termination do
    For each  $\vec{x}_d$  in D compute
         $o_d = \vec{w} \cdot \vec{x}_d$ 
    For each linear unit weight  $w_i$ , Do
         $\Delta w_i = -\eta \sum_{d \in D} (o_d - t_d) x_{id}$ 
         $w_i = w_i + \Delta w_i$ 
```

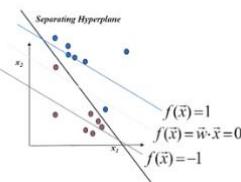
```
Initialize each  $w_i$  to some small random number.
Until termination do
    For each  $\vec{x}_d$  in D compute
         $o_d = (\vec{w} \cdot \vec{x}_d)$ 
    For each linear unit weight  $w_i$ , Do
         $\Delta w_i = -\eta (o_d - t_d) x_{id}$ 
         $w_i = w_i + \Delta w_i$ 
```

Batch

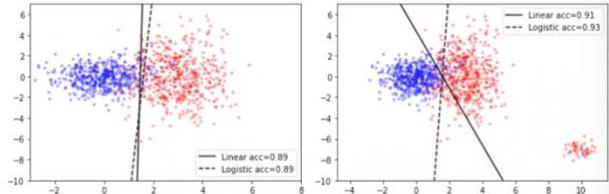
Stochastic

var Yakhini IDC

$$E[\vec{w}] = \frac{1}{2} \sum_{d \in D} (t_d - o_d)^2 = \frac{1}{2} \left[ \sum_{d \in D'} (\vec{w} \cdot \vec{x}_d - 1)^2 + \sum_{d \in D'} (\vec{w} \cdot \vec{x}_d + 1)^2 \right]$$



Given the hypersurface (derived from attributes of objects *linear* and *logistic*), we found the line  $h(x, y) = 0.5$   
(the threshold that separates the data, in the LoR case)



לכן, נרצה לשפר את LMS ולכפות עליו למזער טעויות: נרצה למצוא פונקציית טעות שאכן סופרת טעויות. הפונקציה מיינן:  $m$  הוא הגודל של  $D$ , ונרצה לספר את כמות הטעויות. הסכום על פונקציית היסמן אשר מבצעת מכפלה בין הליביל לבין הפרדיקציה שלו (כאשר אנו מסווים בין 1 ל-1), תסכם ל- $m$  אם אין טעויות מפני שמיינוס מינוס יניב פולס, ופולס יניב פולס, וכל מיס קלסייפיקציה תישפר כמינוס. לכן ברגע שנבעצט מחרות הסכום היליל, ככל שהחסכים יותר קרוב ל- $m$ , יש פחות טעויות, וככל שנתפרק מ- $m$ , האלגוריתם שלו יבצע יותר מיס-קלסייפיקציות. השאלה היא איך ניתן לעבוד ולמזור את הפונקציה הזו (שמהוירה ערכיהם בין 0 טעויות ל- $m$  טעויות) שכן לגזר את פונקציית היסמן לא תניב תוצאה יפה. מה שambil אוננו ל-Perceptron.

בגרדיינט דיסנט סטנדרטי טעויות נסכימות על כל הדגימות לפני העידכונים:  $E[\vec{w}] = \frac{1}{2} \sum_{d \in D} (o_d - t_d)^2$ .  
לפעמים ה- $t$  training set הוא מאוד גדול ואנחנו לא נרצה לבצע לעדכו עד שנרייך לולאה על כל הדגימות. בגרסת הסטטיסטיות של האלגוריתם, משקלים מתעדכנים על פי דוגמה אחת רנדומלית או על פי קבוצה קטנה של דוגימות mini-batch. המשמעות היא שדרושים פחות חישובים לפני כל עדכון, אבל גם דרושים קטנים יותר מכיוון שלא עושים שימוש בגרדיינט האמיטי. הגישה הסטטיסטיות יכולות להתמודד יותר טוב עם הבריחה ממיינמוס לקליל וכן מאפשרות מיקובל.  
מה שראינו עד עכשיו היא גישת-h Batch (חישוב הטעות על כל הדאטה).

$$\begin{pmatrix} \mathbf{x}_{10} & \mathbf{x}_{11} & \dots & \mathbf{x}_{1n} \\ \mathbf{x}_{20} & \mathbf{x}_{21} & \dots & \mathbf{x}_{2n} \\ \vdots & \vdots & & \vdots \\ \mathbf{x}_{m0} & \mathbf{x}_{m1} & \dots & \mathbf{x}_{mn} \end{pmatrix} \begin{pmatrix} \vec{w} \\ t_1 \\ t_2 \\ \vdots \\ t_m \end{pmatrix} = \begin{pmatrix} \mathbf{t} \\ w_0 \\ w_1 \\ \vdots \\ w_n \end{pmatrix}$$

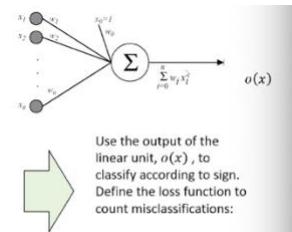
הערה נוספת: גישת-h-pseudo inverse (או pinve) גם כן עובדת עבור LMS

### מהן הבעיות באלגוריתם LMS? נסה למנות אותן באמצעות הדוגמה הבאה:

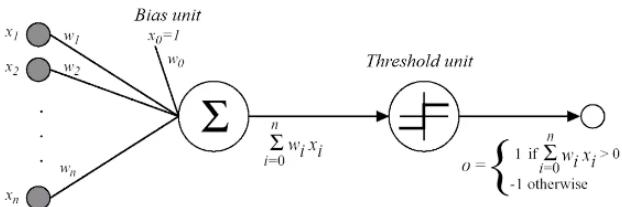
נפעיל את האלגוריתם על הדאטה הכלול-סגול שלנו ונקבל את הקו האפור בהира. ולמעשה הוא אינו מפדר מושלם! יש טעויות, ישנה נקודה סגולה בין כחולים ונקודה חולה בין סגולים! אבל הדאטה אכן ניתנת להפרדה ליניארית – להלן הקו השחור! אבל למה ההפרדה איננה מושלמת? זאת מכיוון שהגדנו פונקציית טעות שאינה סופרת מיס-קלסייפיקציות!  
אלא רק מגיעה לסוג של אייזון, גרסיה, מרתקים "טוביים" מספיק בין הנקודות להפרדה.

### שווה בין LMS ל-LR:

נפעיל את שני האלגוריתמים על שני דatasets, כך שהקלסייפיקציה עברו שני קבוצות הדאטה היא כחול-אדום, ונחשב את הדיוק של כל אחד מהאלגוריתמים (בכמה אchosים בלבד). בDATAה השמאלי שנוצר ממשני גאוסיאנים אחוז זהה 89% עבור שני האלגוריתמים. לעומת זאת בDATAה הימני אשר נוצר מוגaussian כחול אחד ושני גאוסיאנים אדומים, אחוז הדיוק של LoR גובר על אחוז הדיוק של המסוג הליניארי LMS. הפונקציה הליניארית רואה את הען האדום קטן כ-1 או -1, אך היא לא "מסתובבת" יותר מדי ולכן מトוספות אליה טעויות.



$$E[\vec{w}] = \frac{1}{2} \left( m - \sum_{d \in D} \text{sgn}(t_d o_d) \right)$$



$$o(x_1, \dots, x_n) = \begin{cases} 1 & \text{if } w_0 + w_1x_1 + \dots + w_nx_n > 0 \\ -1 & \text{otherwise} \end{cases}$$

Or in vector notations:

$$o(\vec{x}) = \begin{cases} 1 & \text{if } \vec{w} \cdot \vec{x} > 0 \\ -1 & \text{otherwise} \end{cases}$$

**The Perceptron**  
נרצה לבדוק את האינפוטים ו- $x_0=1$  ולבצע מכפלה פנימית עם וקטור המשקלים  $w$ , לבדוק אם הוא מניב תוצאה חיובית או שלילית ולסתוג בהתאם  
לקלאס 1/-1. עלינו ללמד  $w$ -ים שיביאו למינימום את הפונקציה הזו.  
להלן האלגוריתם הלומוד:

### The Perceptron Algorithm

אנו נניח כי הלייבל-target value עבורי הקלסיפיקציה היה 1 או -1.  
ואלגוריתם זה הינו דומה מאוד ל-LMS – אך מטרתו היא למצוא את המ퍼דר  
הlieneariy לא טעויות! וכן פרספטרון הוא תמיד סטוכסטי, אין פרספטון אשר  
משתמש בדגם אחד או קבוצה קטנה של דגימות (batch). האם הדבר  
תמיד מתאפשר?

- נתחל את ערכי וקטור המשקלים  $w$

- נבצע עד התכנסות:

$$o_d = \text{sgn}(\vec{w} \cdot \vec{x}_d)$$

$$\Delta w_i = -\eta(o_d - l_d)x_{id}$$

$$w_i = w_i + \Delta w_i$$

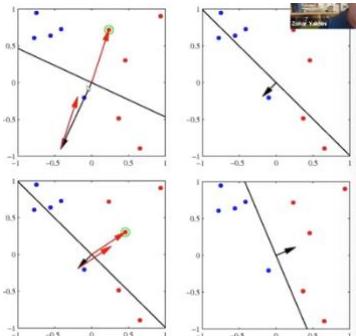
-- וכל וקטור פיצרים  $d$  ב- $D$  נחשב:

-- וכל יחידת משקל לינארית  $w$  נבצע:

For any misclassified (at the present iteration) training instance,  $x$ , with  $C(x) = +1$  we update the weights as:  
 $w = w + \eta x$

העבה: כאשר בחישוב הדלטה  $w$  נשים לב כי  $pd=0$  שווה ל-0 רק כאשר אין טעות! ולכן לא למה לבצע שניוי / לתיקן. למעשה העידכו הזה 0

כאשר אנחנו צודקים, כאשר אנחנו טועים ולמשל התקבל 2, אז  $w$  כולו כאשר אנחנו טועים אנחנו מוסיפים "חתייכה" של  $pd$ . אז איך בעצם פונקציית הטעות משתמשת בכל איטרציה?



Red dots are positive ( $t = +1$ )  
The marked one is initially mis-classified  
Recall that we then update by:  
 $w^{(j+1)} = w^{(j)} + \eta x$   
We therefore get, for a misclassified  $x$  with  $C(x) = +1$ :  
 $w^{(j+1)} \cdot x > w^{(j)} \cdot x$

נדגים כיצד פרספטון מושפר את פונקציית הטעות:

נשים לב שלבי השיפור הם: השלב הראשון הוא הגרף השמאלי העליון, השלב השני הוא הגרף הימני העליון, השלב השלישי הוא הגרף השמאלי התיכון והשלב הרביעי בו הגיעו למסוג מושלם הוא הימני התיכון. כאשר הווקטור והשינוי שלו מיוצג על ידי החץ השחור. והחצים האדומים מייצגים את הווקטורים  $pd$  אשר מצביעים לנקודות DATA, ו"חץ" מthem על פי הטעות מותוספים ל- $w$ .

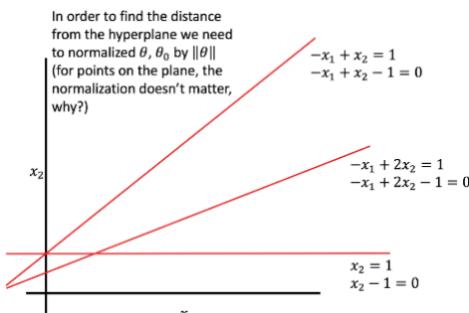
- Note: we also need to control  $\eta$  to really guarantee convergence  
(if its too big we may overshoot the perfect classifier)
- Some results on the rate of convergence were proven and can be useful in the context of ANNs
- The Perceptron itself is not a practical learning approach but is an important component of many modern learning approaches.

**משפט רזונבלאט לאלגוריתם הפרספטון:** אלגוריתם הלמידה פרספטון מתקנס למסוג מושלם (ללא טוויות על-הה) אם ורק אם  $pd$ , training data, מיתנת להפרדה לינארית.

## Linear Classifiers – תרגול 5

### מפרידים לינאריים בדימויים

- היפר-משיר במרחב: במרחב חד ממדי היפר משיר הוא נקודה, במרחב דו ממדי היפר משיר הוא קו ובמרחב תלת ממדי היפר משיר הוא משטח דו-ממדי. נשים לב שכיבוקל מפריד לינארי הוא מפריד במדוד אחד פחות (ביצוג היגיומי) מרחב הדאות בו אין מותעדים.
- הגדרת היפר-משיר במרחב: מרחב היפר-משיר הוא  $1-m$  (אם  $m$  הוא המרחב בו אנו עוסדים). כל הנקודות על היפר-משיר פותרות את המשוואה הבאה:  $\theta_0 + \theta_1x_1 + \dots + \theta_nx_n = b$  ( $= \theta_0$ ) כאשר  $x$  הוא וקטור הקואורדינטות של הדגימה. היפר-משיר מפריד את המרחב לשני מרחבים, כאשר כל נקודה שנמצא תוצאה במשווה שגדולה מ- $b$  ( $b$ -bias, טיטה<sub>0</sub>) נמצאת מעל המישור, וכל נקודה שנמצא תוצאה במשווה שקטנה מ- $b$ , נמצאת מתחת למישור.
- ה- $b$ , bias, טיטה<sub>0</sub> – מייצג את התזוזה של המישור מראשת הצירויות, והטיה (הכללית, המקדמים של  $x$ ) בכל מייצגת את התזוזה של המישור במרחב.
- נשים לב שהדוט-פרודקט = המכפלת הפנימית, בין  $x$  לטטה, משמעותה המרחק של הנקודה מהמשיר. על המישור יש נורמל שמאווק למישור, הדוט-פרודקט הוא הטלה של האיקס על הנורמל. מרחק חיובי משמעותו = מעל המישור, ו למרחק שלילי משמעותו = מתחת למישור. נשים לב שכדי לקבל את המרחק האימיטי על הווקטור טטה להיות מנורמל (חלק אותו בטורמל שלו עצמו).
- אם הטטה לא מנורמלת = המרחק לא מייצג את המרחק האוקלידי מהמשיר וכך יש משמעות רק לסימן של המרחק כפי שצוין לעיל.



- ה- $x_1$ ,  $x_2$ ,  $x_1$  AND  $x_2$  – מיצגים את התזוזה של המישור מראשת הצירויות, והטיה (הכללית, המקדמים של  $x$ ) בכל מייצגת את התזוזה של המישור במרחב.
- נשים לב שהדוט-פרודקט – המכפלת הפנימית, בין  $x$  לטטה, משמעותה המרחק של הנקודה מהמשיר. על המישור יש נורמל שמאווק למישור, הדוט-פרודקט הוא הטלה של האיקס על הנורמל. מרחק חיובי משמעותו = מעל המישור, ו למרחק שלילי משמעותו = מתחת למישור. נשים לב שכדי לקבל את המרחק האימיטי על הווקטור טטה להיות מנורמל (חלק אותו בטורמל שלו עצמו).
- אם הטטה לא מנורמלת = המרחק לא מייצג את המרחק האוקלידי מהמשיר וכך יש משמעות רק לסימן של המרחק כפי שצוין לעיל.

אנחנו מ Chapman מפריד לינארי כך שכל הנקודות המניבות תוצאה גדולה מ-0 יסווו לקלאס 1 (+). לעומת זאת נמנים מchapins טטה  $\theta \in R^{n+1}$  ( $n$  hyperplane weights & the bias  $\theta_0$ ), שהיא וקטור משקלות או מקדים ל- $x$ , שמיימת את המכפלת הפנימית המתווארת לעיל ומסוגת באופן הבא:

- $X_1$  AND  $X_2$
- $X_1$  OR  $X_2$
- Solution?
- If  $1 \times X_1 + 1 \times X_2 - 1.5 > 0$  predict 1
- Otherwise predict -1.
- i.e.  $\theta_0 = -1.5, \theta_1 = 1, \theta_2 = 1$
- Solution?
- $X_1 + X_2 - 0.5 > 0$  predict 1
- Otherwise -1

1 if  $\sum_{i=1}^n \theta_i x_i + \theta_0 > 0$  and -1 otherwise

להלן כמה דוגמאות:

### אלגוריתם הפרספטוריון

אלגוריתם שמחפש מפריד לינארי במרחב, שכן מchapins טטה באופן הבא: נתחיל עם טטה רנדומית (משיר) ובכל שלב נשפר אם יש צורך בשיפור (כלומר יש יש שגיאה = אופטימום פחות טרגט). השינוי בטטה הדרולונית היא learning rate כפול גודיאנט.

### כל העדכון של הפרספטוריון

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

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

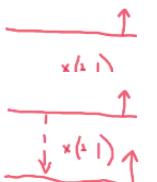
זהו הכלל לעדכו כך שאם  $o = Td - Od$ , אין טעות ולכן אין צורך בעדכו.

מעבר לשורה הראשונה: נשים לב שיש לנו נקודה שאינה מחקלאה החובי +1 = אבל המכפלת הפנימית עבורה שלילתית -1 =, על כן מבחן גאומטרית כך יראה המשיר (איוור). לכן ההפרש בין 0 ל- $t$  הוא שלילי, ונניח שפיציר ה- $-x$  הוא חיובי, אז הדוט-פרודקט שליהם שלילי וחדל תא

טטה חיובית (מן שאנחנו מכפלים במינוס את הדוט-פרודקט). לכן, המשיר "ירוד" לכיוון הנקודה, לכיוון השילוי, ונקודה שהעדכו גרים לכך שהנקודה תהיה כתע מעל המשיר – המרחק בין  $x$  למשיר גדול מפני שהdot-product בין  $x$  לטטה גדול.

מעבר לשורה השנייה: קורה דבר דומה, שוב יש להוריד את המשיר לכיוון נקודה ולכן המרחק, המכפלת הפנימית בין  $x$  לטטה, גדול.

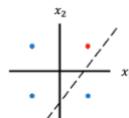
לכן באופן כללי אם המכפלת הפנימית בין  $x$  לטטה גבוהה, המשיר "ירוד" ביחס לנקודה שאנו מדברים עליו. ואם המכפלת קטנה, המשיר "עליה" ביחס לנקודה שאנו מדברים עליו.



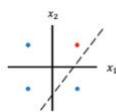
## Perceptron Algorithm

- The algorithm:

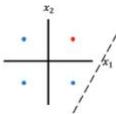
- Initialize weights to some small random number
- Repeat until convergence (no error = no weight update):
  - For each  $x^{(d)}$  in D compute: (\*  $x^{(d)} = \bar{x}_d$ ):
    - $o^{(d)} = \text{sgn}(\theta \cdot x^{(d)})$
  - For each  $\theta_i$  do:
    - $\Delta\theta_i = -\eta \sum_{d \in E}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



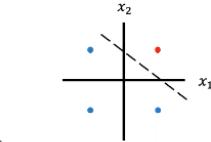
- Training data:
  - (-1,-1)>+1, (-1,+1)>+1, (+1,-1)>+1, (+1,+1)>-1
- Weight init:
  - $\theta_0 = 0.1, \theta_1 = -0.2, \theta_2 = 0.15, \eta = 0.05$



- Check  $(+1,-1)>+1$ 
  - $sgn(\theta \cdot x^{(d)}) = 0.1 - 0.2 * (+1) + 0.15 * (-1) = -0.25 < 0$
  - $o = -1$
- Since  $t!=o$  update required:
  - $\theta_0(\text{new}) = 0.1 - 0.05 * (-1 - 1) * 1 = 0.2$
  - $\theta_1(\text{new}) = -0.2 - 0.05 * (-1 - 1) * 1 = -0.1$
  - $\theta_2(\text{new}) = 0.15 - 0.05 * (-1 - 1) * (-1) = 0.05$



- Check  $(+1,+1)>-1$ 
  - $sgn(\theta \cdot x^{(d)}) = 0.2 - 0.1 * (+1) + 0.05 * (+1) = 0.15 > 0$
  - $o = +1$
- Since  $t!=o$  update required:
  - $\theta_0(\text{new}) = 0.2 - 0.05 * (+1 - (-1)) * 1 = 0.1$
  - $\theta_1(\text{new}) = -0.1 - 0.05 * (+1 - (-1)) * 1 = -0.2$
  - $\theta_2(\text{new}) = 0.05 - 0.05 * (+1 - (-1)) * 1 = -0.05$



We got the linear separator:

$$\theta \cdot \bar{x} = -0.1 * x_1 - 0.15 * x_2 + 0.2 = 0$$

## Stochastic Perceptron

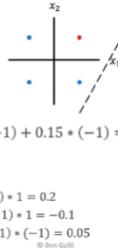
- The algorithm:

- Set weights randomly
- Repeat until convergence:
  - Choose  $d$  randomly (or in some order)
  - Calculate  $o^{(d)} = \text{sgn}(\theta \cdot x^{(d)})$
  - Calculate  $\Delta\theta_i = -\eta (o^{(d)} - t^{(d)}) x_i^{(d)}$  for each  $i$
  - Then update  $\theta_i = \theta_i + \Delta\theta_i$

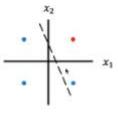
שלב ראשון: אתחול הטהה ולהלן המישור החתולי, כאשר "מעל" המישור מוגדר להיות שמאליה כלפי מעלה, שכן יש שתי טוויות האדומה שמעל המישור והכחולה שמתחתי למשור.

שלב שני: בדיקה עבור כל אחת מהנקודות, עבור 2 נקודות לא יתבצע עדכון ויתבצעו שני עדכנים עבור הנקודה הגדולה מהכחולה.

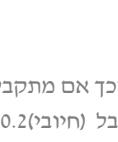
הכחולה שמתחתי למישור ועבור הנקודה האדומה שמתחתי למשור.



- Check  $(+1,-1)>+1$ 
  - $sgn(\theta \cdot x^{(d)}) = 0.1 - 0.2 * (+1) + 0.15 * (-1) = -0.25 < 0$
  - $o = -1$
- Since  $t!=o$  update required:
  - $\theta_0(\text{new}) = 0.1 - 0.05 * (-1 - 1) * 1 = 0.2$
  - $\theta_1(\text{new}) = -0.2 - 0.05 * (-1 - 1) * 1 = -0.1$
  - $\theta_2(\text{new}) = 0.15 - 0.05 * (-1 - 1) * (-1) = 0.05$



- Check  $(+1,+1)>-1$ 
  - $sgn(\theta \cdot x^{(d)}) = 0.2 - 0.1 * (+1) + 0.05 * (+1) = 0.15 > 0$
  - $o = +1$
- Since  $t!=o$  update required:
  - $\theta_0(\text{new}) = 0.2 - 0.05 * (+1 - (-1)) * 1 = 0.1$
  - $\theta_1(\text{new}) = -0.1 - 0.05 * (+1 - (-1)) * 1 = -0.2$
  - $\theta_2(\text{new}) = 0.05 - 0.05 * (+1 - (-1)) * 1 = -0.05$



בצע בדיקה עבור הנקודה החתולית, ומכוון שהבדיקה לא נכונה יתבצע עדכון – לפני העדכון מופיע מושם אל ולאחריו מופיע מימין. נשים לב כי הטעה גדל ולכן המרחק מראשי היצרים גדול, שזה מצוין עבוריו כי כך הנקודה החתולית אכן תתמקם מעל המישור. ונשים לב שטעה 1 וטעה 2 שמורות על הסימנים שלhn וועל כן ניתן להבין שהזווית לא משתנה מאוד.

בצע בדיקה עבור הנקודה האדומה, וגם כאן יתבצע עדכון. לפני העדכון מושם אל ולאחריו מימין. נשים לב כי הטעה 2 קטנה ולכן התקרבנו לראשיה, ובן השטנה הסימן של טעה 2 וכן שינוי בזווית של המישור. כתה הנקודה האדומה אכן מותחת למישור.

אבל נוצרה שגיאה עבור הנקודה החתולית  $(+1, -1)$  שאמורה להיות מעל המישור אך מופיעה מותחת למישור.

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

איך יודעים מה זה מעל ומה זה מותחת למישור? מציבים את ראשית היצרים וכך אם מתקבל ערך חיובי, אנחנו מעל המישור אחרת מותחת. אם נציב את ראשית היצריםכאן נקבל (חיובי) 0.2 ולכן לכיוון הכתולים נחשב מעל המישור.

### מה הבעה באlgorigthm הפרספטורוני ואיך נוכל לפתור אותה?

אלגוריתם אין תנאי עזרה למעט התכונות less-than-classification, perfect classification, מה שלא תמיד יתאפשר. נפתרו אותה על ידי הגבלת כמה ריצות (הבעיה כאן היא שאנו לא דע אם יכולים לשפר את מה שהגענו אליו עם "עוד קטת" ריצות). או על ידי הגבלת מספר הטעויות אבל גם כאן נוצרת אותה בעיה. ולבסוף הפתרון האלגנטי ביותר היא להפוך את בעיה זו לבעית אופטימיזציה – להשתמש בגרדיאנט-דיסנט ולמצוא מינימום טעות.

אלגוריתם זה נקרא LMS.

## להלן אלגוריתם הפרספטורוני

אשר מתיחס לכל השגיאות (ועל כל הסכום לפני האטה) אבל נוכל להשתמש בפרספטורוני טוכטוי, שהוא משתמש כל פעם בשגיאה של נקודת אחת.

### דוגמה עבור פרספטורון סטוכטוי:

שלב ראשון: אתחול הטהה ולהלן המישור החתולי, כאשר "מעל" המישור מוגדר להיות שמאליה כלפי מעלה,

לכן יש שתי טוויות האדומה שמעל המישור והכחולה שמתחתי למשור.

שלב שני: בדיקה עבור כל אחת מהנקודות, עבור 2 נקודות לא יתבצע עדכון ויתבצעו שני עדכנים עבור הנקודה הגדולה מהכחולה.

הכחולה שמתחתי למישור ועבור הנקודה האדומה שמתחתי למשור.

בצע בדיקה עבור הנקודה האדומה, וגם כאן יתבצע עדכון. לפני העדכון מושם אל ולאחריו מימין. נשים לב כי הטעה 1 וטעה 2 קטנה ולכן התקרבנו לראשיה, ובן השטנה הסימן של טעה 2 וכן שינוי בזווית של המישור. כתה הנקודה האדומה אכן מותחת למישור.

אבל נוצרה שגיאה עבור הנקודה החתולית  $(+1, -1)$  שאמורה להיות מעל המישור אך מופיעה מותחת למישור.

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

איך יודעים מה זה מעל ומה זה מותחת למישור? מציבים את ראשית היצרים וכך אם מתקבל ערך חיובי, אנחנו מעל המישור אחרת מותחת. אם נציב את ראשית היצריםכאן נקבל (חיובי) 0.2 ולכן לכיוון הכתולים נחשב מעל המישור.

## 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: (\*  $x^{(d)} = \tilde{x}_d$ ):
    - $\hat{o}^{(d)} = (\theta \cdot x^{(d)})$
  - For each  $\theta_i$  do:
    - $\Delta\theta_i = -\eta \sum_{d \in D} (\hat{o}^{(d)} - t^{(d)}) x_i^{(d)}$
    - Update  $\theta_i = \theta_i + \Delta\theta_i$

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

Minimize the distance between the positive instances and the  $+l$  iso-line of the function  
Minimize the distance between the negative instances and the  $-l$  iso-line of the function

אלגוריתם זה פועל בדיקת אופן בלבד הבודה שהוא משתמש בפונקציית טעות שונה = אין שימוש בפונקציית הסימן אלא ממשמש מושגים בדוט-פרודקט ומטרתו היא לא למצוא קלסיפיקציה מושלמת – מפריד מושלם, אלא לסייע את הטיעות, את המרחקים.

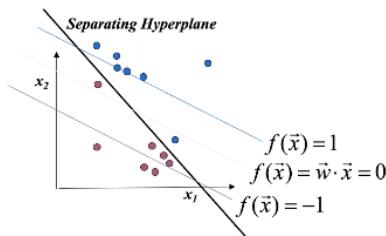
### הבדלים בין פרספטון ל-LMS הם:

- ה-label / target (לא קלאס) – במקרה זה  $+1, -1$ .
- חישוב האופטימוט – התוצאה היא מספר ולא קלאס (ערך האמיתי של  $\theta \cdot x$  ולא חישוב השימן של מכפלת פנימית זו).
- פונקציית האופטימיזציה – ב-LMS המרכיב המינימלי  $m-1, +1$  יוצר את המישור.
- ובפרשptron המישור יוצר 0 שגיאות קלסיפיקציה.
- LMS יתכנס, פרספטון לא בהכרח מתכנס אלא אם הדאטה ניתנת להפרדה ליניארית (אין שגיאה על ה-data).(training data).

שליך רגסיה ליניארית על קלסיפיקציה:  
במקרים לחזות ערכים רציפים ננסה להזוזם קלאס.

Predict

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



- What if instead of predicting a continuous value we try to predict a class?

- Hypothesis function :

$$h_\theta(x) = \theta^T x$$

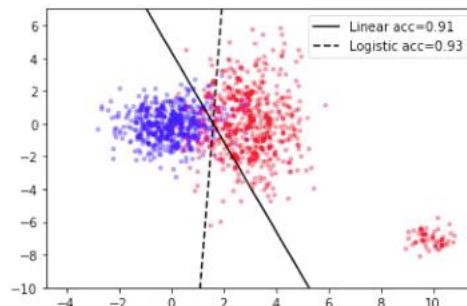
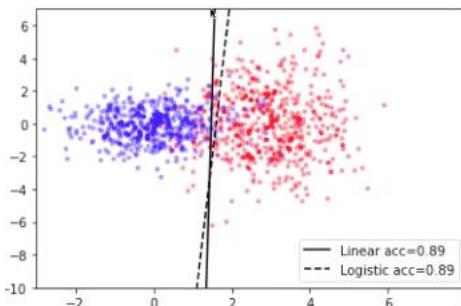
- Cost function (MSE):

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^m (h_\theta(x^{(i)}) - y^{(i)})^2$$

- Goal :

$$\min_{\theta} J(\theta)$$

$x_1$	$y$
1	1
1	1
0	0
1	1
0	0
1	1



ניסיונות לשילוטים,

פרשptron לא תמיד טוב  
מן שלא תמיד יתכנס

وكلاسيفيكيه على في رgression  
لينياريت = LMS לא תמיד  
תנייב מפריד אופטימום (כפי  
שניתן לראות מימין ובדוגמה  
לעיל).

ולכן נשאלת השאלה איך מוצאים מישור באlgorigitms שמתכנס גם אם הדאטה לא מופרד ליניארית? <<

## Logistic Regression

$$h_\theta(x) = P(1|x) \quad \text{פונקציית ההיפזזה היא הփוטרירוי.}$$

ננסה לחזות את ההסתברויות של דוגמיה להסתברות? כל שנקודה נמצאת בצד נכון ורחוקה יותר מהמישור, ההסתברות שלה גבוהה יותר. ככלומר

איך נעבור מරחק ממישור להסתברות? ככל שנקודה נמצאת מצד אחד ורחוקה יותר מהמישור, ההסתברות שלה גבוהה יותר. ככלומר

עלינו לחתך למרחקים ולהפוך אותן להסתברויות ביחס למישור. מתמטית נעביר סקללה של מרחקים לסקלה של 0-1. אחת

הפונקציות שמודעות לעשות זאת נקראת

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

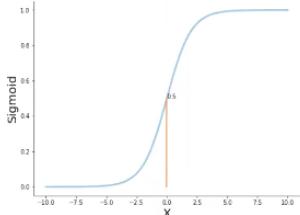
ונשתמש בה ב-LoR sigmoid : sigmoid

$x_1$	$y$
1	1
1	1
0	0
1	1
0	0
1	1

כאשר מכפלה פנימית של טבה עם איקס היא בדיקת המרחק של הנקודה מהמישור ולכן אנחנו לוקחים את המרחק הזה מכנים אותו לsigmoid ומקבלים תוצאה בין 0-1.

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

לכן, נציב את הסיגמוד (מסומן ב-S) בפונקציית ההיפותזה שלנו, ופונקציית ההיפותזה תהיה  $h_{\theta}(x) = S(\theta^T x)$ . והפרדייציה שלנו תהיה  $1 - 0.5 > 0.5$  ואחרת, 0.



נשתמש ב-maximum likelihood כדי להגדיר את פונקציית הולות. נרצה לחזות את הליביל ע' בהינתן הדאטה והמיישור, ולקבל עבורה את ההסתברות הגבוהה ביותר. ובאופן מתמטי נרצה למקסם את ההסתברות

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

הבא (הלייקלידוד): נזכור כי (א) הולות האינסטנס  $x$  להשתטייך לפחות 1.

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

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

ולכן נקבל:

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

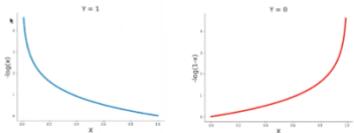
פיתוח מותमטני: לאחר שנניח אי תלות על האינסטנסים נרצה למצוא את המקסימום של מכפלת ההסתברויות המוקפת באדום, לכן נפעיל לו ונגזר ונקבל את הגרדיינט – פיתוח מלא ומפורט יותר מופיע בהרצאה

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

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

Cost Function Intuition

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



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

לאחר הצבת הסיגמוד זו הפונקציה שורצאה למשער:

نمזר את הפונקציה הזה על ידי גראדיאנט דיסנט באשר זו פונקציית הולות:

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

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

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

ולכל  $m$  נקודות הדאטה מתקיים:  
וכעת נפעיל גראדיאנט דיסנט.

Hypothesis function :

$$h_{\theta}(x) = S(\theta^T x)$$

Cost function:

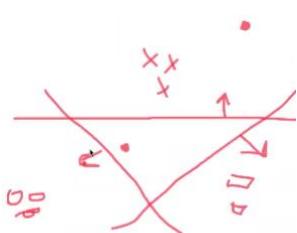
$$J(\theta) = \frac{1}{m} \sum_{d=1}^m -y^{(d)} \cdot \ln(h_{\theta}(x^{(d)})) - (1 - y^{(d)}) \cdot \ln(1 - h_{\theta}(x^{(d)}))$$

Goal :

$$\min_{\theta} J(\theta)$$

(כאן אין פתרון על ידי ידו).

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



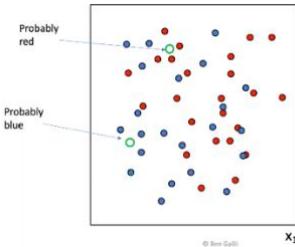
איך נוכל לפתור בעיה שיש בה יותר משני קלאסים עם מפריד ליניארי? גישת one vs all.

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 על הקלאס שביחסו לאחורי הולות נרצה שנקודה חדשה חדשה להשתティיך אליו. תסוג אליו – ההיפותזה של כל קלאס אומרת מה ההסתברות שהיא מושתתית לקלאס זה. עבור הנקודה העלומה בורר שההיפותזה של קלאס האיקסים תהיה גבוהה מההיפותזה של הריבועים והעיגולים ולכן היא תסוג כאיקס.

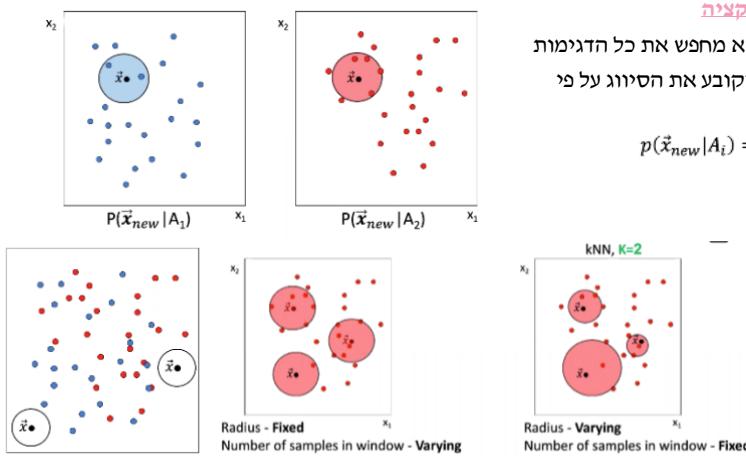
## KNN - K Nearest Neighbors – תרגול 6 – קבוצה nearest neighbor מוגדרת

האלגוריתם הוא משפחה שנקראת **instance based learning**. נניח כי נרצה להשכיר דירה, נתבונן בדירות דומות ונראה שהמחיר שלן הוא לפחות או יותר כמו הדירה אותה אנו רוצחים להשכיר. מבחינה יותר פרקטית, נניח שלפנינו הדאטה הכלול והאדום ונרצה לסוג את הנקודה החדשה הירוקה, מכיוון שסבירבה רק נקודות נרצה לסוג אותה ככחולה. זהו למעשה ייחודי של **אלגוריתמים שהם מבוססי נתונים**.

לעומת זאת באלגוריתמים **שמודליים**, כמו רוגריה למשל, מסתכלים על הדאטה ומנסים למודל אותה לפונקציה או מישור שמודלים היכי טוב את הדאטה הקיים ובוסף המידה **נותר** רק הדאטה הקיים.



- משפחת האלגוריתמים שהם **אינם בונים מודלים לדאטה** (כמו עץ החלטה), במקומם הם משווים אינטנסיבי חדש לאינטנסיבי הקיימים ב-data training.
- **הסבירויות** של אלגוריתמים מסווג: הלמידה היא מהירה מפני שבפועל אין באמות למידה, אבל קיים פוטנציאלי לסיווג/חיזוי/קלסיפיקציה איטיים (ב-(מ)O), מכיוון שיתקן מצב בו נדרש לעבור על כל הדאטה שלנו, וכך **space complexity** גדול, והוא עבר על כל הדגימות וכן הוא (מ)O.
- ניתן להשתמש באלגוריתמים מסווג זה הן לקלסיפיקציה והן לרוגריה.



**האלגוריתם הבסיסי של מבע קלסיפיקציה Parzen window** אשר מבע קלסיפיקציה

זהו אלגוריתם שמייצר ודיוס חיפוש קבוע, כאשר מגיעה דגימה חדשה הוא מחפש את כל הדגימות הקיימות שנכנסות לתוך "הכדור" הרוב ממדדי שנוצר על פי הרדיוס הקבוע וקובע את הסיווג על פי החסתברות. עושים זאת על פי הנוסחה הבאה:

$$p(\vec{x}_{new} | A_i) = \frac{1}{n_i} \sum_{\vec{x} \in A_i} \frac{1}{h^d} K\left(\frac{\vec{x}_{new} - \vec{x}}{h}\right)$$

אבל אלגוריתם זה אינו יודע לטפל במצבים בהם הרדיוס לא תופס אף דגימה הקיימת בדאטה. אלגוריתם דומה שעזר לנו לפתור היה תיקון פלט. אז נשנה את האלגוריתם – במקום שהרדיוס יהיה קבוע, נקבע את מספר הדגימות. אם קודם המעלים היו בגודל קבוע, אז כעת נשנה את גודל הרדיוס כך שיכנסו לתוכו המספר הרצוי של הדגימות. **זהו האלגוריתם של KNN. נשים לב שה-k – אינו תלוי בקלט.**

### אלגוריתם KNN

עכשו השאלה שעה, **בציד מגדילים את הרדיוס?** בפועל, אנחנו לא באמות "מגדילים רדיוס", אלא אנחנו עוברים על כל הדאטה ומחשבים את מרחק כל הדגימות מהדגימה החדשיה, ו-k הדגימות בעלות המרחק הכי קצר הן אלו ש"יכנסו לרדיוס" = קלומר על פיהם נחשב החסתברות ובוצע סיווג/קלסיפיקציה או נמצע וכך נבע פרדיקציה לרוגריה.

- אם אנחנו ברוגריה נעשה פרדיקציה על פי הממוצע  $\hat{f}(x) = \frac{1}{k} \sum_{i=1}^k f(x^{(i)})$
- ואם אנחנו בклסיפיקציה נעשה פרדיקציה על פי **majority vote**

### ה יתרונות והחסרונות של האלגוריתם

#### יתרונות

- זמן מהיר ללמידה / אין training
- אנחנו לא מאבדים מידע בתהילך המידול מפני שאנו שומרים את כל הדאטה
- ניתן ללמידה פונקציות קומפלקס / מטרה מאוד מורכבת

#### חסרונות

- זמן ריצה בפרדיקציה יכול להגיע ל-(n)O ובאופן כלל לא יהיה מהיר
- דריש אחסון זיכרון מרובה
- מושפע בקלות מדגימות / אטוריוביוטים פחות רלוונטיים

### השאלות שועלות מהאלגוריתם הן:

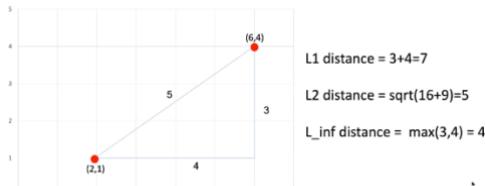
- (1) איך נגידר "קרוב"?
- (2) איך מתמודדים עם שאילתה איטית ועם מקום האחסון הגדל שעליו להחזיק?
- (3) ואיך נבחר את k?

$$L_p(x^{(i)}, x^{(j)}) = \sqrt[p]{\sum_{l=1}^d |x_l^{(i)} - x_l^{(j)}|^p}$$

- מרחק עבור פיצרים נומריים נחשב על ידי מרחק  $L$ : כאשר  $l$  הוא האינדקס של ממד הוקטור ו- $d$  הוא הממד

- **אם  $p=2$ :** פונקציית מרחק אוקלידי משתמשת בשורש ריבועי ומעלה ריבועית (כמו בעולם שלנו). מרחק אוויראי.

- **אם  $p=1$ :** פונקצייה זו נקראת מרחק מנהטן. סכום הצלעות.



### Distance For Numeric Features

- When  $p = 2$  the  $L_p$  distance is called the Euclidean distance
- When  $p = 1$  the  $L_p$  distance is called the Manhattan distance
- When  $p = \infty$  we define this function as follow:

$$L_\infty(x^{(i)}, x^{(j)}) = \max_l |x_l^{(i)} - x_l^{(j)}|$$

$$\bullet x^{(1)} = (1, 2, 4), x^{(2)} = (4, 0, 3)$$

$$\bullet \text{When } p = 2:$$

$$L_2(x^{(1)}, x^{(2)}) = \sqrt{\sum_{l=1}^3 (x_l^{(1)} - x_l^{(2)})^2} = \sqrt{(-3)^2 + (2)^2 + (1)^2} = \sqrt{14}$$

$$\bullet \text{When } p = 1:$$

$$L_1(x^{(1)}, x^{(2)}) = \sum_{l=1}^3 |x_l^{(1)} - x_l^{(2)}| = 3 + 2 + 1 = 6$$

$$\bullet \text{When } p = \infty:$$

$$L_\infty(x^{(1)}, x^{(2)}) = \max(|-3|, |2|, |1|) = 3$$

Example:

$$x^{(1)} = (7, 2), \quad x^{(2)} = (5, 5)$$

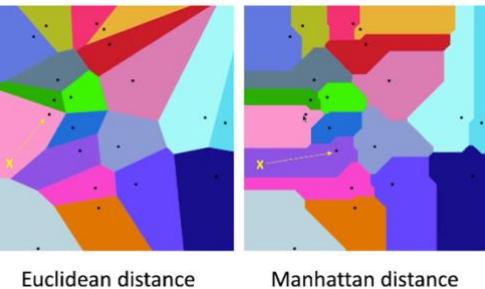
Which point is closer to origin?

Depend on the distance method:

Euclidean -  $x^{(2)}$

Manhattan -  $x^{(1)}$

### דוגמאות חישוב : 3



דוגמא נוספת מבחןת הסביבות והבדל בין מרחק אוקלידי ומרחיק מנהטן - **Diagrammat**:  
כל נקודה בתוך צבע מסוים, הנקודה השורה שקרובה אליה הנקודה שבתוך הצבע לפי שתי  
השיטות – תיאורית KNN יכול ללמד מכך פונקציית מאוד מורכבת.

**מה לגבי נתונים נומיים?** למשל כחול, ירוק, אדום... נחפוץ לDATA נומי ולאחר מכן נמדד  
מרחיקים (יש כאן קאזי) – אם נחפוץ אותם 1,2,3- זה יאמר למשל ירוק יותר לכחול וכדומה  
ולכן יש להיזהר עם זה). או שנשתמש בדרכים כמו Hamming, Value Difference Measure  
וכיו... שאלוHon שיטות שמחשובים מרחיקים על נתונים נומיים.

**מרחיק Hamming** = מרחיק ההמינג בין שני סטרינגים בעלי אורך זהה הוא מספר המוקומות שבהם "האותיות" בסטרינגים  
שונות. להלן דוגמאות חישוב.

אפשרויות נספת היא ליצור וקטור מקודד כך שעבור כל צבע ניצור וקטור שכל slot בו מייצג איזה צבע אתה ולבן זהו וקטור יחידה  
(1,0,0), (0,1,0), (0,0,1), אבל קידוד זה יכול ליצור בעיה כאשר יש לנו פיצרים ויש לבצע סוג של איזון.

**אלגוריתם KNN ממושקל** מה קורה כאשר יש לי שכן שהוא "מושך יותר קרוב" אליו? נרצה להעניק לו משקל גבוה יותר בהכרעה.  
לכן הנשכותות עבור KNN ממושקל עניינו לארוך יותר משקל גובה יותר:

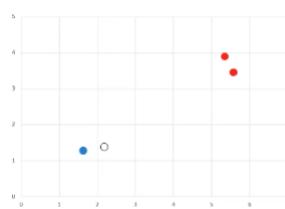
- עבור ורגישה/ערכים רציפים נשמש בנוסחה (ממוצע ממושקל):

$$\hat{f}(x) = \frac{\sum_{i=1}^k w_i f(x^{(i)})}{\sum_{i=1}^k w_i}$$

Where  $w_i = \frac{1}{\text{distance}(x^{(i)}, x)}$

עבור קלסיפיקציה נשמש ב- 1 חלקי המשקל. להלן דוגמה.

- K=3
- The 3 nearest neighbors:
  - X1 distance = 5, class = No
  - X2 distance = 2, class = Yes
  - X3 distance = 5, class = No
- Regular kNN will output 'No'
- The weighted:
  - $MAj \left( \frac{\text{No}}{5}, \frac{\text{Yes}}{2}, \frac{\text{No}}{5} \right) = \text{'Yes'}$



## 2- שיפור היעילות של האלגוריתם מבחןת זמן ריצה ו מבחנת מקום בזיכרון

אנו מחשבים זמן ריצה כרגע באופן הבא :  
 $T_{predict\ sample} = N_{samples} * T_{compute\ distance}$

נרצה למצמצם את זמן הריצה בעת שאילתה ואת המקום בזיכרון בו אנו משתמשים. לשם כך, למשל נמצמצם את  $N$  הדגימות על ידי סינון דגימות. מה שיעזר לנו למצמצם את זמן החיפוש בנוסף הוא שימוש במבנה נתונים מתאים למשתמש K-D Tree.

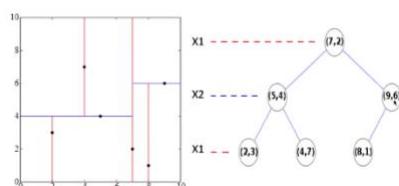
כדי למצמצם את זמן החישוב של המרחק –  $T$ , נבצע calculation interrupt, או שנצמצם את מספר הפיצרים.

### מילה בונושא Curse of Dimensionality (נזהר לההמשך)

**מדוע לא נרצה להתחשב בכל הפיצרים באלגוריתם שלנו :**

- מספר הדגימות הדרשיות גדול באופן אקספוננציאלי עם מספר המשתנים.
  - המידע הרלוונטי שמור רק במס' מועט של פיצרים מתוך כל הפיצרים הנתונים לנו.
  - במקרים רבים כל הדגימות רוחקות אחת מהשנייה – והדבר לא טוב עבור NN.
- בפועל, מעבר לנקודת מסוימת, הוספה של פיצרים נוספים מובילה לביצועים לא טובים.

Points set: (2,3), (5,4), (9,6), (4,7), (8,1), (7,2)



### שיפור היעילות על ידי שימוש ב-K-D Tree

במוקום לחפש את השכן הקרוב ביותר על כל ה-training data נבנה מבנה נתונים שייעיל לחיפוש. נחלק את הדadataה ל-partitions, בכל פעם בממד שונה. נחפש אחר שכנים, ראשית נמצאת את החלוקה הרלוונטית ואז בצע חיפוש על החלוקה זו בלבד.

הרעינו הכללי נכון אבל יש ניואנסים קטנים שאנו מוביילים לאופטימיזציה, לא ניכנס אליהם.

דוגמא :

### ביצד למצמצם דגימות מה-data למען שיפור היעילות

**המטרה שלגנטיאת:** להוריד נקודות שלא משפיעות על גבולות ההכרעה – ככל שהנקודה רחוקה יותר מהגבול הסיכון שהיא תשפיע על ההכרעה קטן.

**ישנן שתי שיטות גריידיות לעשות זאת – מפני שהן תלויות סדר.**

- **השיטה הירושית:** נכנס ל-set נקודות אחת אחרי השניה אבל נשמר רק את אלו שאינם מסוכנות נכון.

- **השיטה ההפוכה:** קיבל את כל הנקודות בתוך ה-set וגעבור על הנקודות ונסיר את אלו שמסוכנות נכון על פי שכני ה-KNN שלהם.

• Backward KNN( $S$ )  
 $T = S$   
For each instance  $x$  in  $T$   
if  $x$  is classified correctly by  $T - \{x\}$   
remove  $x$  from  $T$   
Return  $T$

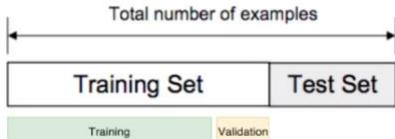
• Forward KNN( $S$ )  
 $T = \emptyset$   
For each instance  $x$  in  $S$   
if  $x$  is **not** classified correctly by  $T$   
add  $x$  to  $T$   
Return  $T$

### מתכבל ב-k-NN עבור k-ים קטנים מדי : Overfitting

לכן יהיה علينا להגדיל את ה- $k$  עד לנקודת מסוימת. מפני שככל נקודה יוצרת מסביבה מרחב של כל הנקודות שקרובות אליה, לכן כאשר ה- $k$  קטן מאוד ייווצרו איים כמו שיתין לראות בדוגמה עבור  $k=1$ , הדבר משפייע מאוד על הגבולות וזה מצב של overfit. בנוסף, עבור training data אחר יתקבלו "נקודות רעש" שונות ולכן גם שם עשוי להתפרק overfitting.

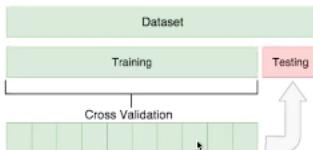


### 3- פיזול הדאטה ו-Cross Validation

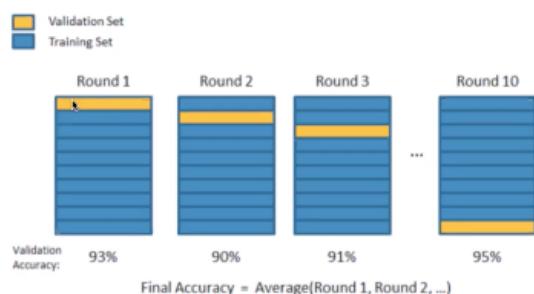


כאשר אנו משתמשים במודל סטטיסטי (כמו גורסיה לינארית, למשל), אנחנו מותאים את המודל על ה-training set במטרה לבצע פרדיקציה על דאטה חדשה. במטרה לעשות זאת אנו מותאים את המודל על הדאטה לקבוצות-training ו-test. נשים לב שאנו יכולים להשתמש ב-test set במטרה לבחור את ההיפר-פרמטרים שלנו, ואנו יכולים בונסף test set, ליצור קבוצה נוספתvalidation.

אבל, לפיזול הדאטה פעם אחת בלבד יש מספר חסרונות: אם הדאטה שלנו אינו מופיע מופיע אז אנחנו נהיה חייבים להימנע מפיזול כדי לא לשביד מידע, ואם נקבל פיזול שאינו מייצג אז שיטתה זו לא תעבור (נוכל לצמצם זאת על ידי הפעלת "שאפל" על הדאטה).



נשתמש ב-Cross Validation כאשר הדאטה שלנו אינו מופיע מופיע גדול כדי פיזול k- קבוצות אלה או כאשר אנחנו מעדיפים לא לחלקה ספציפית של 3 קבוצות. למעשה קרוס-וילידישן עובד באופן דומה לפיזול val/split,k-1,k. מלבד העובדה שהוא מופעל על יותר תתי-קבוצות. כמובן, אנחנו מפצלים את הדאטה ל-k קבוצות, מותאמים על k-1 קבוצות מתוך ה-k שיחסקנו, ואת הקבוצה האחורונה נשמר ל-test. אנחנו יכולים לעשות זאת עבור כל אחת מתתי הקבוצות.



ב-k-folds cross validation k-1 תתי קבוצות כדי לאמן את הדאטה ומשאירים את (folds).

אנחנו משתמשים ב-1-k תתי קבוצות כדי לאמן את הדאטה ומשאירים את הפולד האחרון להיות ה-test set.

לאחר מכן אנחנו עושים ממוצע על המודל לנגד כל אחד מה-folds ולאחר מכן test set נגד הדאטה.

אנחנו מסיימים למודל ובודקים נגד ה-test set.

### מניבת תוכאה סטטיסטית יותר חזקה מאשר חלוקה רנדומית Cross Validation

ל-3 קבוצות ו גם מתאפשר להפעלה על גבי datasets קטנים.

### Leave One Out Cross Validation (LOOCV)

בסוג זה של ה-CV, מספר ה-folds (תתי הקבוצות) שווה למספר הדוגימות בדאטה-סט ובכל פעם אנחנו משתמשים בכל הדוגימות בלבד. דגימה אחת שמשמשת טסט.

לאחר מכן אנחנו ממצאים את כל ה-folds הללו.

מכיוון שנקבל מספר גדול מאוד של training sets (שווה למספר הדוגימות), שיטה זו מאוד יקרה מבחינה חישובית ועדיין להשתמש בה עבור datasets קטנים.

אם ה-dataset היו גדול, סביר כי יהיה עדיף להשתמש בשיטה אחרת, כגון k-folds.

היתרון הגדול בשיטה זו היא רמת הדיוק. החיסרונו הוא חוסר הפרקטיות עבור datasets גדולים.

### از מתי משתמש באיזו שיטה?

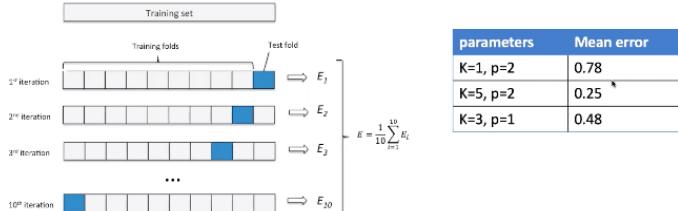
- ככל שיש לנו יותר folds : נקטין את השגיאה על-h-bias אבל נגדיל את השגיאה על השונות, המחיר החישובי יגדל, וכן הסטם – ככל שיש לנו יותר folds ייקח יותר זמן לחשב עבור כולם וכן ידרש שימוש גדול יותר בזיכרון.

- ככל שיש לנו פחות folds : אנחנו מקטינים את הטעות על השונות, אבל הטעות על-h-bias תהיה גדולה יותר. מבחינה חישובית, יותר.

- For kNN – need to choose

- K=?
- P=?

- Cross validation – a method for hyper parameter optimization



parameters	Mean error
K=1, p=2	0.78
K=5, p=2	0.25
K=3, p=1	0.48

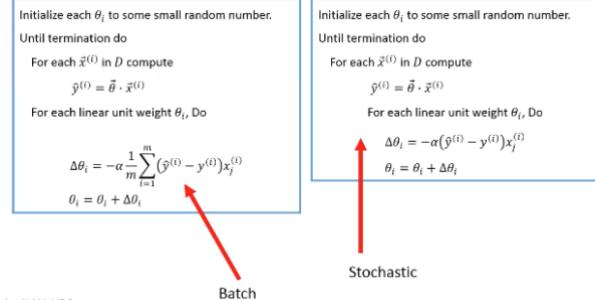
לכן, ב-datasets גדולים מומלץ לקובע k=10.

ב-3 datasets, כפי שכבר צוין, עדיף להשתמש ב-LOOCV.

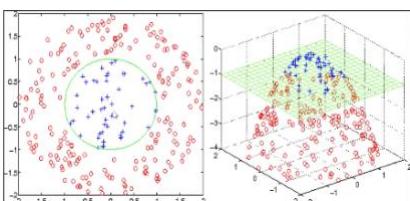
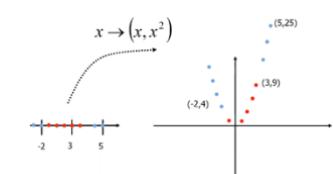
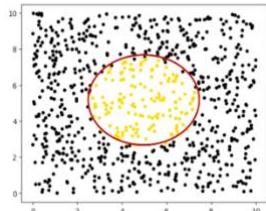
## סידרת חישובית ממען: רצאה 7

## המש

- ניתנה כאן תזכורת על גרדיאנט דיסנט בצורה של באז'י סטוכסטי (כל מה שנאמר מופיע בהרצאות על LoR ורגסיה לינארית).
- נשים לב כי בשקף זה בכל מקום בו מופיעה טהה' מדובר בטטה'ן.
- התזכורת נאמרה כאן מפני האלגוריתם הפרספטורון שלמדנו בשיעור שuber מתנהג כמו גרדיאנט דיסנט סטוכסטי, אין גרסה אחרת לפרספטורון.
- סטוכסטי = מקרי, רנדומי. מבחינת שינוי סדר הדוגמאות שיכול להיות רנדומי.



```
Initialize each  $w_i$  to some small random number.
Until termination do
    For each  $\vec{x}_d$  in D compute
         $o_d = \text{sgn}(\vec{w} \cdot \vec{x}_d)$ 
        For each linear unit weight  $w_i$ , Do
             $\Delta w_i = -\eta(o_d - t_d)x_{id}$ 
             $w_i = w_i + \Delta w_i$ 
```



$$(x, y) \mapsto (x, y, x^2 + y^2)$$

$$x^2 + y^2 - 1 = 0$$

$$w = (-1, 0, 0, 1)$$

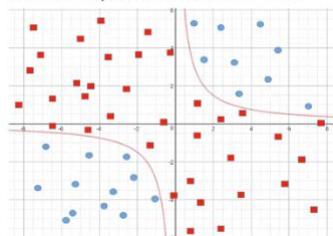
הפרספטורון ומפרידים לינאריים – מיפוי לממד גובה יותר

לפנינו דאטה בממד 1 שאין לנו להפרדה לינארית. נפה את הדאטה הניל לממד 2 כך שהנקודה 2- במד 1 הפכה לנקודה (-2,4) בממד 2 על פי התייאור  $(x, x^2)$ : x.

בממד הגובה יותר יש מפריד לינארי!

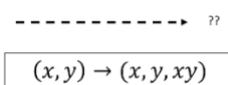
לכן לאחר מיפוי לממד 2 ולפי משפט רוזנבלט, **הפעלה של פרספטורון על הדאטה** **במימד החדש אכן תניב מפריד לינארי**.

Linear separation in 2D? 3D?



דוגמאות נוספות:

נדיר את ההפרדה בממד הגובה יותר על ידי וקטור המשקלות, כך למשל, בדוגמאות אלו וקטור המשקלות א' עבור הממד הגובה יותר – ממד 3, יכול 4 ערכים.



:phi-separability גדרה

$\varphi$ -separability

חלוקה (בינהריה), או זיכיון, training set  $C_0, C_1$  ותקרא

$w^T \varphi(x) > 0 \quad x \in C_0$  אם קיימים וקטור w מימד N כך ש: ניתן לחלק ב-phi, :

$$w^T \varphi(x) < 0 \quad x \in C_1$$

$$\varphi(x) = \langle \varphi_1(x), \dots, \varphi_N(x) \rangle$$

$\varphi_i$  Mapping functions

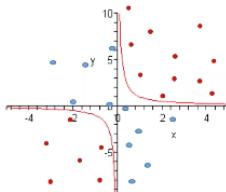
$\{\varphi_i(x)\}_{i=1}^N$  Mapped instance

Engineering the map:

$$(x_1, x_2) \mapsto (1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1 \cdot x_2, x_1^2, x_2^2)$$

And we will then get

$$w = \left( -1, 0, 0, \frac{1}{\sqrt{2}}, 0, 0 \right)$$



הרענון מאחוריו Non-linear Mapping : נפה דاطה ממוחב אינפוט

בעל ממד נמוך למרחב אימבינטי (ambient space) גובה ונקודות

שהՃאטה תהיה ניתנת להפרדה לינארית במרחב החדש. דוגמה למיפוי

מממד 2 לממד 4 :

$$(x_1, x_2) \mapsto (1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1 \cdot x_2, x_1^2, x_2^2)$$

ופונקציית הדיסקרימיננטה תהיה :

$$F(x) = \sum_{i=0}^5 w_i \varphi_i(x) = \\ w_0 + \sqrt{2} w_1 x_1 + \sqrt{2} w_2 x_2 + \sqrt{2} w_3 x_1 x_2 + w_4 x_1^2 + w_5 x_2^2$$

הרענון הוא שוכן להשתמש במרחבים יותר ויותר גבויים מבחינת ממד עד אשר נגיע להפרדה הילינארית הרצויה – שזו המטרה שלנו.

### משפט קובר סופר את הדיכוטומיות שניתנתה להפרדה לינארית Cover's Function Counting Theorem Counting Dichotomies

- דיכוטומיה של קבוצה S היא חלוקה של S לשתי תת-קבוצות-זרות של S.
- נניח כי יש לנו K דגימות בקבוצות דגימות S.

אז יש לנו  $2^{K-1}$  אפשרויות לדיכוטומיות שונות מעל הדגימות הללו

כל דיכוטומיה מדירה משנית סיווג/קלסיפיקציה (הפרידה בין קלאסים)

משפט קובר Cover's Counting Theorem : במרחב N-ממדי מספר הדיכוטומיות של K דגימות הניתנתה להפרדה לינארית הוא :

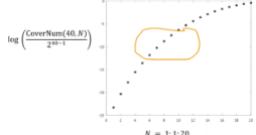
$$P(K, N) = \frac{\sum_{i=0}^N \binom{K-1}{i}}{2^{K-2}}$$

$$2 \sum_{i=0}^N \binom{K-1}{i}$$

:

כל ש-N גדול, מספר הדיכוטומיות גדול וכן גם החסתברות גדולה (בנחתה כי K הוא מספר קבוע, שבדרכ כל גובה הרבה יותר מ-N).

\*\* אין משמעות להערכת ההסתברות, נשאר מהשקל רקודם שוחר הראה.



### Full Rational Varieties

גדרה: full rational variety מסדר r במרחב אינפוט מממד n מתואר על ידי כל המונומיאלים (monomials) מדרגה r של משתני האינפוט ב-x

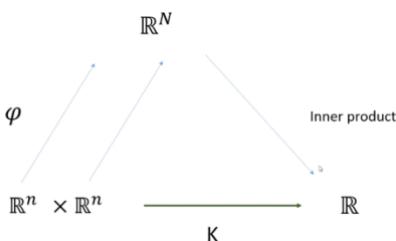
$$\sum_{j=0}^n r_j = r \quad \varphi_i(\vec{x}) = 1^{r_0} x_1^{r_1} x_2^{r_2} \dots x_n^{r_n}$$

$$\frac{(n+r)!}{n! \cdot r!} = \binom{n+r}{n}$$

מספר גורמי המונומיאל השונים בביטוי זה הוא :

אם  $n=10$  ו- $r=5$  נקבל 3000 ממדים (מממד 10 ומעלה). פרקטית אנחנו עלולים לממדים מאוד גבויים. נרצה להיות יותר חסכוניים מפני שבמצב זה

הפרשפטון יצרך לבצע 3000 מכפלות פנימיות! מה שמאפשר לנו לחסוך, הם קרנלים.



### Kernels – קרנלים

פונקציה K ממרחב הזוגות הסדורים (R^n x R^n) ל-R נקראת קרנל (kernel) אם קיים

מייפוי (מסומן באות היוונית פי) מ- $R^n \times R^n$  ל- $R$  כך שמתקיים :

$$K(x, y) = \varphi(x) \cdot \varphi(y) \quad \text{für } \varphi: R^n \rightarrow R$$

קרナル הוא הכללה של מכפלה פנימית.

קרנלים עוזרים לנו להימנע מחיפוי אקספליסיטי אחר מרחב אימבינטי (ambient space) ואחר פונקציות מייפוי, פי, לממד גבוה יותר.

הקרנלים מעבירים את הלמידה לפעולות ישרות בממד האינפוט (שהוא מממד נמוך יותר).

למעשה, קרנלים יכולים גם לתמוך בלמידה של מייפוי מרחבים בממד אינסופי (general Hilbert spaces).

## More Kernel Examples

Corresponds to  $\varphi$  being a full rational variety of order  $d$

Homogenous Polynomial kernel:  $k(x, y) = (x \cdot y)^d$

Inhomogenous Polynomial kernel:  $k(x, y) = (x \cdot y + 1)^d$

Radial Basis function kernel:  $k(x, y) = e^{-\frac{\|x-y\|^2}{2\sigma^2}}$

Sums and products of kernels are kernels as well!

**דוגמאות לKERNEL:**

Given  $x = (x_1, x_2)$  let  $\varphi(x) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)$

Given  $x = (x_1, x_2)$   $y = (y_1, y_2)$  we then get

$$\begin{aligned}\varphi(x)\varphi(y) &= (x_1^2, \sqrt{2}x_1x_2, x_2^2) \begin{pmatrix} y_1^2 \\ \sqrt{2}y_1y_2 \\ y_2^2 \end{pmatrix} \\ &= x_1^2y_1^2 + 2x_1x_2y_1y_2 + x_2^2y_2^2 \\ &= \left( (x_1, x_2) \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \right)^2 = (x \cdot y)^2 = k(x, y)\end{aligned}$$

Hence,  $k(x, y) = (x \cdot y)^2$  is a kernel for this  $\varphi$

**Learning:**

- Initialize each  $w_i$  to some small random number.
- Until termination do
  - For each  $\vec{x}_d$  in D compute  $o_d = \text{sgn}(\vec{w} \cdot \vec{x}_d)$
  - For each linear unit weight  $w_i$ , Do
 
$$\Delta w_i = -\eta(o_d - t_d)x_{id}$$

$$w_i = w_i + \Delta w_i$$

**Classification, use:**

$$f(x) = \text{sgn}\left(\sum_i w_i x_i\right)$$

## חזרה לפרספטון – צעד עדכון המשקלים

- רַק לְדָגִינוֹת שָׁהֵן מְסוּגָּות בָּאוּמָן שָׁגַּי יִשְׁשַׁפְעָה
- אִם אֱאוֹטֶפֶט שְׁלִילִי וְהַטְּרוֹגֶט (הַלִּיבֶּל) חִוּבִּי, אֲנַחֲנוּ מְעַלְּמִים אֶת הַמְשֻׁקּוֹלָות עַל יְדֵי הַוּסְפָּת חָלֵק/שְׁבָר חִיוּבִּי של הדגימה
- אִם אֱאוֹטֶפֶט חִוּבִּי וְהַטְּרוֹגֶט (הַלִּיבֶּל) שְׁלִילִי, אֲנַחֲנוּ מוֹרַידִים אֶת הַמְשֻׁקּוֹלָות עַל יְדֵי הַוּסְפָּת חָלֵק/שְׁבָר שְׁלִילִי של הדגימה
- לֹכֶן, בְּעַדְכָּן שֶׁל  $W$ , אֲנַחֲנוּ תִּמְדִיד מִזְרִיפִּים חָלֵק/שְׁבָר חִוּבִּי של  $Xd^* \cdot Td$  (מַכְפָּלָה שֶׁל הַטְּרוֹגֶט/הַלִּיבֶּל (סְקָלָאָר) עַמְּ הַוקְטוֹר  $A$ , אִם יִשְׁתַּחַזְקֵה קָלְסִיפִּיקָצִיה)

גייע באlgorithm המבצע המסומן ע"י פונקציה (א) מסווגה

לכן למעשה יש איזושחו מקדם שמסומן ad שהוא זה שכופל את  $Xd^* \cdot Td$  לכל d דגימות

דאטה, לכל נקודה.

ומכאן נובע השווון השני של לאחר המכפלה הפנימית שבין וקטור המשקלות לקטור

א. השווון השלישי נובע מלינאריות של מכפלה פנימית.

הנקודה שאנחנו באים לסתוג הוא הוקטור  $A$  באlgorithm המבצע. אנחנו מכפילים

.instance based learning – אותו עם כל דגימות הדאטה על פי השווון האחרון

$$f(x) = \vec{w} \cdot \vec{x} = (\sum_d a_d t_d \vec{x}_d) \cdot \vec{x} = \sum_d a_d t_d (\vec{x}_d \cdot \vec{x})$$

$\vec{w}$  is a sum of  $\eta t_d \vec{x}_d$  fractions, added in the learning cycles

There exist (non negative) numbers  $a_d, d \in D$  that can be used to recast the decision function

- מהיות ש-  $A$  הוא קומבינציה של חלקים/שברים של הדגימות כפי
- שחשבנו לעיל, כתבו את פונקציית ההחלטה כז':
- נשים לב כי במתורה לשימוש בפונרציית ההחלטה זו  $f(x) = \sum_d a_d t_d (\vec{x}_d \cdot \vec{x})$  אנחנו צרכים רק לאחסן את כל נקודות הזאתה  $Xd$
- שעבורן  $ad$  אינו 0.
- נקודות דאטה אלו נקראות support vectors

Initialize each  $w_i$  to some small random number.

Until termination do

For each  $\vec{x}_d$  in D compute

$$o_d = \text{sgn}(\vec{w} \cdot \vec{x}_d)$$

For each linear unit weight  $w_i$ , Do

$$\Delta w_i = -\eta(o_d - t_d)x_{id}$$

$$w_i = w_i + \Delta w_i$$



Updates the feature weights

Init the coefficients  $a_i$

Until termination do

For each  $\vec{x}_d \in D$ , do

$$o_d = \sum_{i=1}^m a_i t_i(\vec{x}_i, \vec{x}_d)$$

if the classification is wrong,  
that is  $o_d t_d < 0$  ,  
then

$$a_d = a_d + \eta$$

## הפרספטון הדואלי / The Dual Perceptron

הפרספטון הדואלי (צד ימין) לומד את

המקדמים ad של הדגימות הקיימות בדאטה.

תמיד נסיף חתיכה חיובית של  $Xd^* \cdot Td$ .



Updates the instance coefficients

```

Init the coefficients  $a_i$ 
Until termination do
    For each  $\varphi(\vec{x}_d)$ , where  $\vec{x}_d \in D$ , do
         $o_d = \sum_{i=1}^m a_i t_i (\varphi(\vec{x}_i), \varphi(\vec{x}_d))$ 
    if the classification is wrong, that is  $o_d t_d < 0$  ,
    then
         $a_d = a_d + \eta$ 

```

**מעבר לממדים גבוהים יותר**  
באופן שקול לאלגוריתם הפרספטרון הדואלי, אנחנו יכולים לכתוב את המעבר לממד גבוה יותר  
על ידי שימוש בפונקציה הממפה ( $\varphi$ ) על  $x$  והדגם  $xd$ . ההבדל היחיד הוא השימוש ב $\varphi$ .  
מכאן, **מיון שקיים מיפוי זה נ楹 להשתמש בקורסיל מהדרה**.  
כעת נמיר את הפרספטרון הדואלי לממדים גבוהים יותר על ידי **שימוש בקורסיל במקומם** במכפלה  
פנימית של  $\varphi(x)$  עם  $\varphi(x_d)$ .

### פרספטרון קורסל / The Kernel Perceptron

```

Init the coefficients  $a_i$ 
Until termination do
    For each  $\varphi(\vec{x}_d)$ , where  $\vec{x}_d \in D$ , do
         $o_d = \sum_{i=1}^m a_i t_i K(\vec{x}_i, \vec{x}_d)$ 
    if the classification is wrong, that is  $o_d t_d < 0$  ,
    then
         $a_d = a_d + \eta$ 

```



### SVM = Support Vector Machine האלגוריתם המבצע

- We need to specify three components of an SVM
  - 1. A kernel  $K(*, *)$  – a hyperparameter, provided to the learning algorithm
  - 2. The support vectors  $\vec{x}_d$
  - 3. The weights  $a_d$  for the support vectors
- Choosing the kernel is often done by experience, by trial & error and by using validation data
- The identity of the support vectors will be a by-product of learning the weights  $a_d$  – they will be those instances that have non zero weights
- Note: we will later also add a slack parameter to allow more flexibility (these will be learned based on a control hyperparameter)

- תהי תת-קובוצה של דוגימות training-set שנקראות support vectors
- תהי קובוצת של משקלות עבור הדוגימות הללו

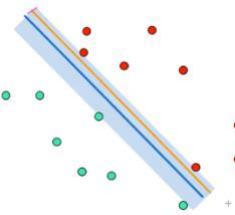
תהי פונקציית קורסל שהיא (כتوزאה, באפקט שלח) לא-лиニアרית ממפה למרחב מממד גבוה יותר.

$$class(\vec{x}) = \text{sgn}\left(\sum_{d \in SV} a_d t_d K(\vec{x}_d, \vec{x})\right)$$

נסווג +/- לפי כל "UMBOSUS דוגימות" פשוט:

מה אנחנו עושים בכלל המסוגן? סוכם על כל הדוגימות שהן support vectors – נפעיל את הקורסל על ה- support vector ועדי  $x$  הנקודה החדשה, נכפיל בליקיל ונשקל במקדם  $ad$ , אותו עליינו ללמידה.

אשר הקורסל הוא היפר-פרמטר ולפננו לבצע עלייו ולידציה, לבחור את הקורסל הטוב מבין הקורסלים.



### נריצה לעשוות אופטימיזציה לשולאים

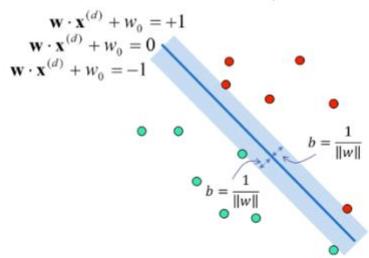
**גזריר:** השולאים (margin) של מרחב מפריד (ליניארי למשל)  $(x, y)$  מוגדרת כמרחק המינימלי בין דוגימה למורחב מעל כל דוגימות

$$\text{Margin} \equiv \min_{d \in D} dist(x_d, f(x) = 0)$$

המטרה שלו במשמעות SVM ולמידת SVM היא **לקסם את המרחק המינימלי**, מה שיחזק את האלגוריתם המבצע  
ליותר יציב, עמיד יותר בפני "רעשים", פחות רגש ובעל הכללה טובה יותר. בדוגמה נרצה למצוא את הקו הכהול.

### SVM Optimization

$$\begin{aligned} \text{Minimize } & \frac{1}{2} \|\mathbf{w}\|^2 \text{ subject to :} \\ & t_d (\mathbf{w} \cdot \mathbf{x}^{(d)} + w_0) - 1 \geq 0 \quad \forall d \end{aligned}$$



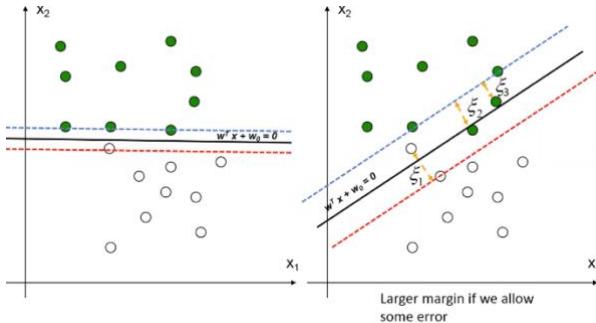
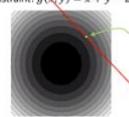
אנחנו מחפשים את המישור המפריד עם השולאים המקסימליים: את המישור המפריד  
מייצג  $w$ , ונרצה להביא למינימום את הפונקציה שモפעה ממשマル, שהיא חצי מהנורמה  
של הווקטור  $w$  (אשר מייצג את המישור הפריד), ביריבוע, **תחת אילוצים מסוימים**:  
כל נקודה בדאטה של  $i$ ,  $d$ , עליינו שהליקיל של  $i$  כפול ( $t_d$ ) המכפלת הפנימית של נקודה זו  
 $xd$  עם  $w$  ועוד 0 ( $w$ ) פחות אחד = אי שלילי. ככלומר נרצה את  $w$  הקטן ביותר שעדיין  
מקיימים את זה. המשמעות = כל הנקודות הן בклאס הנכוון. וה-  $w$  הקטן ביותר מגדר את  
ה-margin הגדול ביותר.

כדי לעשות זאת לא נשתמש בגרדיאנט דיסנט אלא ב קופלי לגרני.

### הרענון של בופלי לגרנוויז:

הכטבים מייצגים את המרחק מהראשית, הנורמה = המרחק האקלידי מוהראשית. כאשר  $g$  מייצג ישר, הישר האדום. ואנחנו נחפש את הנקודה הצחובה. (נרחיב על כך בהמשך)

- Minimize  $f(x, y) = x^2 + y^2$
- Subject to the constraint:  $g(x, y) = x + y - 2 = 0$



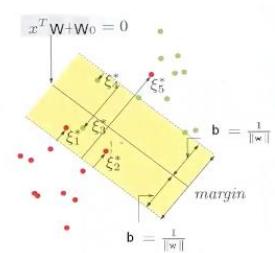
ונכל לאפשר שגיאה מסוימת על הנקודות, כך שלא כלן יסכמו יחד  
-1, אלא  $-1 \leq$  חחות משתנה סלאק כלשהו לכל נקודה  $d$  שהוא אי<sup>b</sup>  
שלילי, וכך שסכום כל-ה-slacks של כל הנקודות יהיה קטן  
מאיזשהו קבוע (המגבלה על הסלאק).  
כך שלבסטוף, נקבל מסוג שעושה שימוש על חלק מהנקודות מה  
שעוזר למנוע overfitting של המודל.

כיצד נמצא שולטים גודלים יותר אם נאפשר מעט שגיאות?  
Slack משתמש במשתנים "גמיישיס" = משתני

- הרענון: אנחנו מוכנים "לסבול" מספר מסוים (לא יותר מדי!) של נקודות training שיסווגו באופן שגוי.
- משתני "Slack" (הסימון הנחשוי, קס) יכולים לעזור לנו:

Minimize  $\|w\|$  subject to:

$$\begin{cases} \forall d, t_d (w \cdot x^{(d)} + w_0) \geq 1 - \xi_d \\ \xi_d \geq 0 \\ \sum \xi_d \leq \text{Const} \end{cases}$$



### סיכום עד כה,

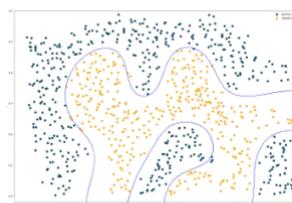
האלגוריתם הוא קבוצת דגימות והליילים שלhn.

היפר-פרמטרים: הקרן, שמאפשר לנו לעבור לממדים גבוהים ושולט בממד אליו אנחנו מופיעים (שהחסכוי למצוא בו מפheid גובה יותר). וכן, הקבוע

שולט במשתני slack.

האוטופוט הוא תחת קבוצה של דגימות training שם ה-  
support vectors וכן קבוצה של משקלות עבור דגימות אלו.  
בעורטם נזכיר את המסוג.

## Support Vector Machines – learning, in practice



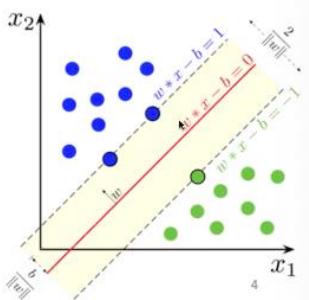
- Input:**  
Data instances and labels
- Hyperparameters:**  
A kernel  
Control of the slack variables
- Output:**  
A subset of training instances - the “support vectors”  
A set of weights for these data points

- Map instance space non linearly into a higher dimensional feature space (mapping space) using a mapping that has a kernel.
- Optimize for linear separability in the higher dimensional space using dual formulation.  
Similar to the kernel Perceptron, but also optimizes margin and converges even if there is no perfect separation.
- This process finds the optimal margin linear separation in the mapping space (using quadratic programming).
- We obtain a non-linear decision boundary in the original instance space.
- Can accommodate some mis-classified training data points, to an extent controlled by a process hyperparameter.

### למידה חישובית ממדים | תרגול 7 – SVM

המטרה שלנו: למצוא מפריד ביןארי שיכל להפריד את הדאטה (היום נתמקד בהפרדה ל-2 קלאסים).  
אלגוריתם SVM מבוסס על 3 רעיונות:

- הernal Trick / The Kernel Trick – מפנה את הדאטה למרחב גבוה שבו יותר קל לסוג עם משטחים מוחלטים שם ביןארים.
- Shallow Margins / Max Margins – עבור בעיית ההפרדה הליניארית, היפר-מישור בעל השולטים המקסימליים הוא המשווה הליניארי האופטימלי.
- גולריזציה ו-Soft Margins – מרחיב את ההגדרה לעיל עבור בעיות הפרדה שאין ביןאריות, לאפשר טוויות.



הגדרות:	
-	משווה ביןארי – פונקציה ביןארית (היפר-מישור במרחב הפיזרים) שיכל להפריד דאטה-סט p-ממדי.
-	$f(\vec{x}, \vec{w}, b) = sign(\vec{w} \cdot \vec{x} + b)$
-	שולטים (xi) = Margin(xi) – המרחק בין גבול החלטה ו-xi.
-	$Margin(\vec{w}, b) = \min Margin(x_i)$
-	$\vec{w}, b = argmax Margin(\vec{w}, b)$ – Maximal margin classifier

הערך הנואם לחת את כל הדאטה שלנו ולעשות עליו איזושה מיפוי למרחב אחר שנקרה שבו יהיה מפריד ביןארי טוב יותר.  
פונקציית פי היא פונקציית המיפוי שלנו.

הבעיה במיפוי הדאטה היא: המיפוי עצמו היה פעולה יקרה ( מבחינת עילוות ) וכן העבודה במרחב גובה היא מאוד יקרה ( מבחינת סיבוכיות זמן ).  
לכן עלינו למצוא דרך לעבוד בגובה מבלית למפותה למדד זהה. עלינו למצוא את הפונקציה שגדולה בעודה במרחב הגובה = קרנל.

### תקרנול טריק / The Kernel Trick

נניח כי אנחנו צריכים רק את המכפלת הפנימית במרחב המיפוי (נראה בהמשך למה הנה זה והוא נכון)	-
ככלומר, נרצה לקחת שני אינסטנסים x ו-y, נפנה את שניהם על ידי פי למרחב הגובה ולאחר מכן נבצע עליהם במרחב הגובה מכפלה פנימית: $(\varphi(x) \cdot \varphi(y))$	-
אם נוכל למצוא פונקציה שמניבה את אותה התוצאה "בליי" למפות, נוכל לצמצם את סיבוכיות זמן הריצה פונקציה זו נקראת Kernel, והקרנול-טריק נועד למנוע את המיפוי	-
-	-

דוגמא לקרナル: ניקח את הוקטור הדו-ממדי הבא  $x = (x_1, x_2)$  ומבצע עליו מיפוי למרחב תלת ממדי:  $\varphi(x) = (x_1^2, \sqrt{2} \cdot x_1 x_2, x_2^2)$ .  
ונחשב את המכפלת הפנימית של פי(x) ופי(y):

$$\begin{aligned} &x_1^2 y_1^2 + 2x_1 x_2 y_1 y_2 + x_2^2 y_2^2 \\ &= (x_1 y_1 + x_2 y_2)^2 \\ &= (x \cdot y)^2 = K(x, y) \end{aligned} \quad \begin{aligned} &x_1^2 y_1^2 + 2x_1 x_2 y_1 y_2 + x_2^2 y_2^2 \\ &= (x_1 y_1 + x_2 y_2)^2 \\ &= (x \cdot y)^2 \end{aligned} \quad \text{כאשר } y \text{ הוא גם וקטור דו ממדי. התקבל (כפל מקוצר):}$$

$\varphi(x) = (1, \sqrt{2} \cdot x_1, \sqrt{2} \cdot x_2, \sqrt{2} \cdot x_3, \sqrt{2} \cdot x_4)$  והמיפוי הבא המכפלת ל-15-ממדים  $x = (x_1, x_2, x_3, x_4)$ .  
דוגמא גנטיפת: עבור 4-ממדים נרצה לחשב את המכפלת הפנימית בין שני וקטורים 4-ממדדים:

$$\varphi(x) \cdot \varphi(y) = 1 + \sum_{i=1}^4 2x_i y_i + \sum_{i=1}^4 x_i^2 y_i^2 + \sum_{i=1}^3 \sum_{j=i+1}^4 2x_i x_j y_i y_j$$

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

לכן  $(x \cdot y + 1)^2$  היא פונקציית הernal.

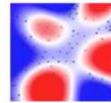
## פונקציות קרNEL

- יש מספר פונקציות קרNEL ידועות

**אנחנו לא צריכים לדעת את המרחב שהקרNEL ממפה אליו**

$$\text{קרNEL פולינומיאלי מדרגה } d: K(x, y) = (\alpha x^T y + \beta)^d \text{ בערך 3 פרמטרים}$$

$$K(x, y) = \exp\left(\frac{-\|x - y\|^2}{2\sigma^2}\right) \text{ : Radial Basis Function (RBF)}$$



מעניין לדעת כי לאחר פיתיחה של RBF קרNEL עם טורי טילול וסיגמה שהיא (1 חלקי שורש 2) קיבל **ממד אינסופי**.

כל שהסיגמה קטנה יותר, הקרNEL מתנהג יותר כמו instance (כמו בNN kNN עבור  $k=1$ ) based

- Where  $\sigma$  is a parameter
- We can replace  $\frac{1}{2\sigma^2}$  with  $\gamma \rightarrow \exp(-\gamma\|x - y\|^2)$

The radius of the "balls" is determined by the parameter  $\gamma = \frac{1}{2\sigma^2}$

- A smaller  $\gamma$  means a larger radius, a lower "model complexity"
- A larger  $\gamma$  means a smaller radius, a finer grain coverage but may lead to an overfit

**קרNEL הוא היפר-פרמטר** – כך שהקרNELים המצוינים לעיל הם מייצגים משפחאות של קרNELים, כאשר אנחנו מפעילים קרNEL אנחנו חייבים להגדיר את הפרמטרים שלו (אלפא, בטא, או ב-RBF סיגמה) לאחרת למעשה בחרנו קרNEL ספציפי.

## Kernel Perceptron

נשים לב (כפי שצווין בהרצאה) כי בכל שלב של הפרספטורו, אם יש צורך בעדכון (כלומר הדוגימה  $d$  סוגה באופן לא נכון = שגיאה) אנחנו מוסיפים חלק/שער קטן של  $Xw$ . המשמעות היא שיש רק אינסטנסים מסוימים שמשנים את  $w$ , רק האינסטנסים שאנו נזקקים בהם. ככלمر תמיד יהיה לנו אינסטנסים מעוניינים יותר ומעוניינים פחות, כאשר המעוניינים יותר הם אלה שאנו נזקקים עליהם ואלו שייעזרו לנו לשפר את המודל ולבנות את המפריד.

כלומר, אם בסופו של דבר  $w$  מקבל כל פעם חלק אחר מ- $Xw$  (אם דוגמה  $d$  היא קיבלה קלסיפיקציה שגויה), נקבל שהמשקלים הם

$$w = \sum_d \alpha_d t_d x_d$$

קומבינציה לינארית של חלק מדוגמאות הדאטה:

כאשר  $\alpha$  אי שלילי (גודול שהוא מ-0).

אינסטנסים שלא משפיעים על תהליכי הלמידה, הם אינסטנסים שעבורם  $\alpha = 0$ .

$$f(x) = \vec{w} \cdot \vec{x} = \left( \sum_d \alpha_d t_d x_d \right) \cdot \vec{x} = \sum_d \alpha_d t_d (\vec{x}_d \cdot \vec{x})$$

מכאן שנשנה את פונקציית החלטה באופן הבא:

האינסטנסים  $Xw$  שעבורם  $\alpha$  שונה מ-0 הם האינסטנסים שאנו צריכים והם נקראים "support vectors"

$$\text{if } \left( t_i \sum_d \alpha_d t_d (\vec{x}_d \cdot \vec{x}_i) \right) < 0: \\ \alpha_i = \alpha_i + \eta$$

כדי להשתמש בצורה הזו של פונקציית ההחלטה علينا לעדכן את שלב העדכון של הפרספטורו:

כך שבמקום לעדכון את  $w$ , נעדכו את אלפא.ad.

כאשר אלפא היה המשקל של האינסטנס.

**בפרספטורו הדואלי**, שהוא כמו הפרספטורו מלבד ההחלפה בין המשקלים  $w$ , למשקלים אלפא, יש לנו את המכפלה הפנימית של דוגמה  $d$  ( $xw$ ) עם הדוגימה החדשה  $x$  מה שיסיע לנו לעבור על ידי פי לממד גבוה יותר, ומעודד אותנו למצוא פונקציית קרNEL כדי למנוע מיפוי.

מה שmobail אותו **לקרNEL פרספטורו – השלב הראשון לפיוון אלגוריתם SVM**.

The Dual Perceptron algorithm:

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

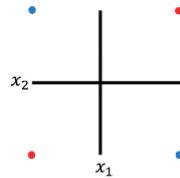
דוגמה הרצה של קרגל פרספטורו:

יש לנו 4 אינסטנסים, ו-2 קלאסים, لكن המרחב הוא דו-ממדי, זה הוא הליבל. זהה בעיית XOR.

$$K(x_i, x_j) = \varphi(x_i) \cdot \varphi(x_j)^2$$

להלן התוצאות המוצגות בטבלה = הדגימות החדשות במרחב 4.

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



נתחול (אתה :

$$\alpha = [\alpha^1, \alpha^2, \alpha^3, \alpha^4] = [0, 0, 0, 0]$$

מבצע בדיקה עבור האינסטנס הראשון  $i=1$ :

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

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

מבצע בדיקה עבור האינסטנס השני  $i=2$ :

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

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

מבצע בדיקה עבור האינסטנס השלישי  $i=3$ :

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

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

מבצע בדיקה עבור האינסטנס הרביעי  $i=4$ :

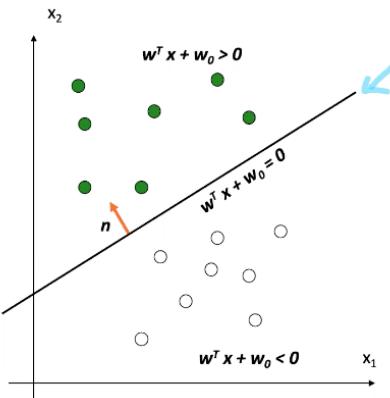
$$\sum_{d \in D} \alpha_d t_d K(\vec{x}_d, \vec{x}_i) = \\ 1 * 0 - 0 * 4 + 0 * 0 - 0 * 4 = 0 \\ sgn(0) = -1$$

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

הדבר האחרון שעשינו לעשות הוא לבצע בדיקה על האינסטנס הראשון שובי, שבאייטרציה הקוזמת האלפה הניבה עבורי שנייה.

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

ואכן מתקבלת תוצאה חיובית כנדרש ובכך מסתיימים האלגוריתם עם לפחות  $[1, 0, 0, 0]$  שמניבת פתרון עבור בעיית XOR!



### מיקסום השולטים – Max Margin

$$f(x) = w^T x + w_0$$

בהינתן פונקציה ליניארית ( $f(x)$ ):

ובהינתן ההיפר-מיישור של מරחב הפיצרים

הנורמל (וקטור ייחידה) של הווקטור המייצג את ההיפר מיישור הינו (הווקטור

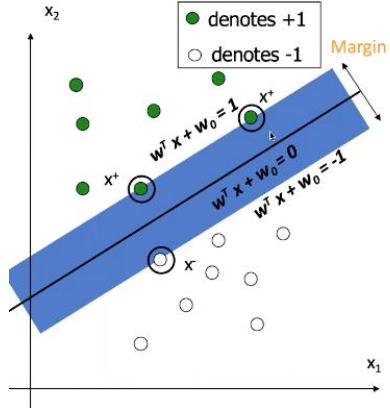
$$n = \frac{w}{\|w\|}$$

לחילק לנורמה שלו):

השאלה שאנו שואלים – איך מישור מפריד בין נקודות לבנות לירוקות והוא

המיישור בו הכי טוב לבחור, שימזע את הטיעות? היוריסטיקה (הנחה הלא

邏輯ית) אומרת שככל ששולוי המישור דוחבים יותר, כך המישור יותר טוב  
ב הכללה.



כדי להבטיח מישור בעל שולטים מקסימליים נדרש :

- **שכל נקודות הירוקות יקיימו את המשוואה הזו:**

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

כלומר יהיו רוחקות לפחות ב-1 מהמישור.

- **וככל הלבנות יקיימו את המשוואה הזו:**

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

כלומר יהיו רוחקות ממהמישור לפחות ב-1 (ב-1 מתחת למישור).

$$M = (x^+ - x^-) \cdot n$$

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

$$w^T x^+ + w_0 = +1$$

$$w^T x^- + w_0 = -1$$

**רוחב השולטים הוא :**

$$= \frac{2}{\|w\|}$$

$$\text{Maximize}_{\|w\|} \frac{2}{\|w\|}$$

כעת נרצה למקסם את הביטוי שקיבלנו עבור רוחב השולטים, ולמכן מטרתנו היא :

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

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

אבל, המיקסום הוא תחת האילוצים הבאים :

שימושותם – לא יהיו נקודות בתחום השטח של השולטים (safe zone) של המישור וכן, המישור יפריד בין הקלאסים.  
ניתן לפתור בעיית אופטימיזציה זו ע"י quadratic programming, שבשל הקורונה לא נתעמק בפתרונו עבורה.

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

- 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:

$$\alpha_d \geq 0$$

כעת, נוכל להחליף את ש באילוקט שלנו – ע"י קופלי לגרנץ'.

כך נוכל למצוא את הגרדיינט עבור ש ו-0-0 (לא נכנס לכל התנהליק בדרך...) ... וב證פוק של דבר נגיע ל :

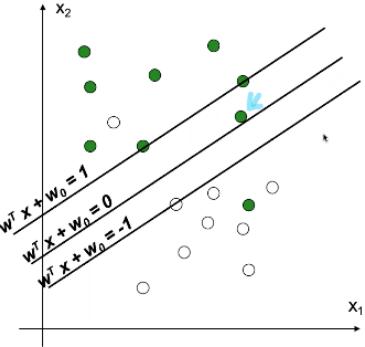
### המשוואת הדואלית של ה-SVM בה נרצה למקסם את השולטים בהינתן האילוצים

$$\begin{aligned} & \sum_d \alpha_d - 1/2 \sum_d \sum_e \alpha_d \alpha_e t_d t_e x_d^T x_e \\ & \sum_d \alpha_d t_d = 0, \alpha_d \geq 0 \end{aligned}$$

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

מה שמאפשר לנו להחליף את המיפוי ל"קרגול-טוריך":

## גnewlineיזציה ו Soft Margin



רוב הדאטה שנתקל בו הוא למעשה מעשה לא ניתן להפרדה ליניארית (דאטה "רעה", outliers, וכו')...  
משתני Slack יכולים לאפשר חירגה מהשוליים (לא בהכרח מיס-קלסיפיקציה, אלא חדרה ל-  
safe zone מרוח ביחסו) של דאטה מורכב או "רוועש"  
נסמן כל חירגה בקסי (הסימן של הנחש) – המרחק מחישול הרלוונטי לគודזה הספציפית.

$$\frac{1}{2} \|w\|^2 + \gamma \sum_d \xi_d$$

ונשנה את בעיית האופטימיזציה למיצועו של:  
כל שהשוליים רחבים יותר, סכום הקשי יגדל, יהיו יותר חירגות. על החירגות יש לנו היפר-  
פרמטר גמא שמכפיל את סכום החירגות. ככל שהגמא (סקלר) גבוהה יותר אנחנו נשלם יותר על  
כל חירגה, וכך הגמא מאנון בין מיקסום השולאים לבין תשלים החירגות.

**כך האילוצים שלנו לבעיית האופטימיזציה ישתנו:**

- 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:

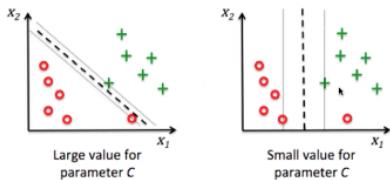
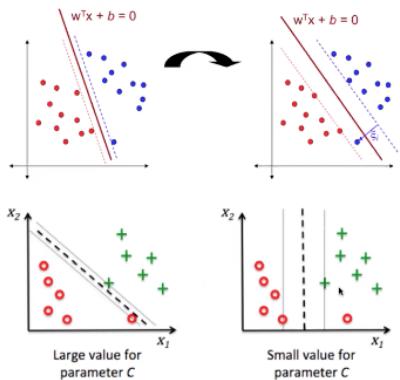
$$\alpha_d \geq 0 \quad \mu_d \geq 0$$



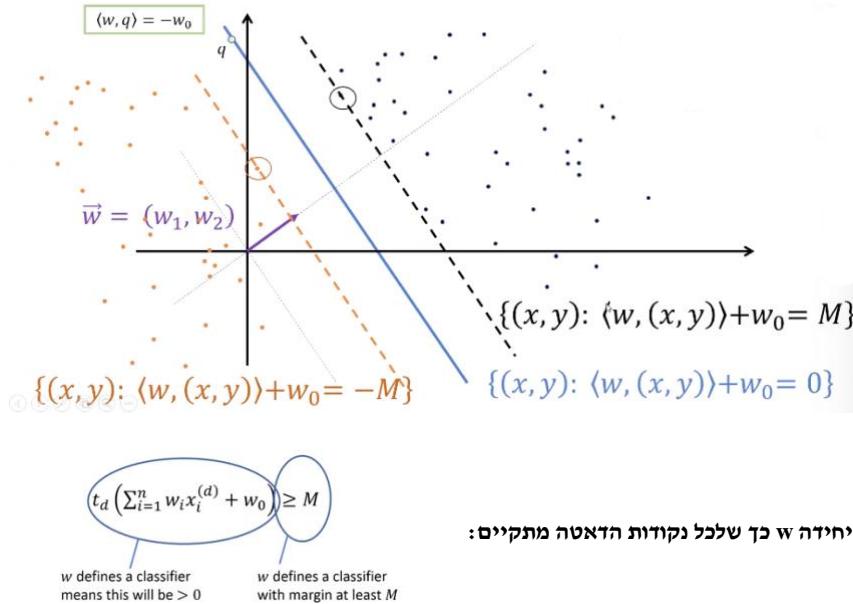
- Dual - maximize

$$\sum_d \alpha_d - 1/2 \sum_d \sum_e \alpha_d \alpha_e t_d t_e x_d^T x_e \\ \sum_d \alpha_d t_d = 0 \quad 0 \leq \alpha_d \leq \gamma$$

נעבור מהמצב הפרימרי למצב הדואלי בו נוכל להשתמש בקרנל טרייך וברגע שנפתרו את המשוואה שנמצאה לעיל נמצא פתרון עבורו  
בעיית האופטימיזציה החדשה שלנו שכוללת את החירגות.



## המשך – Maximal Margins, Optimization and Slack variables – SVM



- הגיאומטריה של השולטים
- אופטימיזציה תחת אילוצים
- משתני Slack

### פירוש גאומטרי עבור השולדים

בגדרה הרשנית של שולדים – margin, התייחסנו למרחק מינימלי בין הנקודה  $X_d$  למישור  $w \cdot x + w_0 = 0$ , כדי למצוא את המרחק הזה אנחנו לוקחים את הנקודה במשור הקרובה ביותר ל- $X_d$  ונחשב עבורה את המרחק (האנך).

נניח כי יש לנו דאטת הדעתה להפרדה ביןארית וחפש  $w$  שהוא מفرد ליניארי שמעורר את השולדים. אם  $M$  זה האורך בו השולדים יישגים (בריהי השגה, achievable) אווי קיים וקטור יחידה (אוורך 1) כך שמתקיים האיוו.

יהי  $M$  השולדים הישגים עבור הדעתה שלנו. אז קיים וקטור יחידה  $w$  כך שלכל נקודות הדעתה מתקיים:

### שולדים מקסימליים / Maximum margin

- נסמן את השולדים עבור מישור מועמד בוקטור יחידה  $w$ , מהיות  $0 < M < \|w\|$  אנו יודעים שעבור כל הדגימות מתקיים הא שוויון לעיל.
- אנחנו מחפשים מסוגם של שולדים מקסימליים ולפניהם, אנחנו מחפשים  $w$  ו- $M$  שיכולים לפחות את בעיית האופטימיזציה הבאה:

ש מגדיר את הכיוון,  $0 < M < \|w\|$  מגדיר את המרחק מהכבש, כאשר  $w$  הוא וקטור והוא ת-ממדי,  $w_0$  הוא סקלר,  $M$  הוא סקלר.  $X$  הוא וקטור ת-ממדי.

הדרינו את כל אוסף נקודות המגדירים את המגבילות לכל נקודות הדעתה, ונרצה למצוא את  $M$  המקסימלי עבורו.

$$\begin{aligned} & \max_{M, w, w_0} M \\ & \text{subject to} \\ & \forall d \quad t_d \left( \frac{\langle w, x^{(d)} \rangle}{\|w\|} + w_0 \right) \geq M \end{aligned}$$

בעיית אופטימיזציה זו שקופה לביעור האופטימיזציה הבאה:  
 $\text{Minimize } \frac{1}{2} \|w\|^2 \text{ subject to :}$   
 $t_d \left( \langle w \cdot x^{(d)}, w_0 \rangle + 1 \right) \geq 0 \quad \forall d$  (הראינו בהרצאה כיצד הגיעו לשקלילות)

### אופטימיזציה תחת אילוצים = כופלי לגרנו'

התובנות על נקודות הקיצון תחת domain מסוים.

השיטה של כופלי לגרנו' (Lagrange multipliers) היא אסטרטגיה למציאת מקסימום/מינימום של פונקציה תחת אילוצי

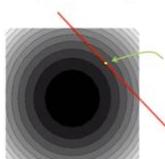
שוויון. למשל:  $\text{Maximize } f(x, y) \text{ subject to } g(x, y) = 0$

אנו מניחים כי גם  $f$  וגם  $g$  בולטות נזרת ראלקית רציפה

$L(x, y) = f(x, y) - \lambda g(x, y)$  השיטה מציגה משתנה חדש (למגדן  $\lambda$ ) שנקרא כופלי לגרנו' ולומדת את פונקציית לגרנו' המוגדרת ע"י:

תנאי הכרחי עבור נקודה  $(x^*, y^*)$  להיות הפתרון של בעיית האופטימיזציה המקורית הוא שכל הנזרות החלקיות של פונקציית

- Minimize  $f(x, y) = x^2 + y^2$
- Subject to the constraint:  $g(x, y) = x + y - 2 = 0$



הקו האדום מייצג את  $g$ . העיגולים הם קווי הגובה של הפונקציה  $f$ . ככל שהעיגולים שחורים יותר כך הערך של  $f$  יותר קטן. נרצה למצוא את הערך המינימלי של  $f$  מתחת למגבילה של הקו האדום. לכן נלק לאורך הקו האדום נגד הגרדיאנט עד שנתכנס למינימום.

$$\nabla L(x, y) = 0$$

דוגמא:

### דוגמה נוספת:

Find the min and max values of  $f(x, y) = x^2 + 2y^2 - 4y$  subject to  $x^2 + y^2 = 9$ .

#### Solution

Set three equations as follows

$$\nabla f = \lambda \nabla g \Rightarrow 2x = \lambda 2x, \quad 4y - 4 = \lambda 2y$$

and the constraint implies  $x^2 + y^2 = 9$ .

$$\begin{aligned} x &= 0 \\ y &= \pm 3 \\ \lambda &= 1 \\ y &= 2 \\ x &= \pm\sqrt{5} \end{aligned}$$



Plugging these 4 points into the function we get:

$$\begin{aligned} f(0,3) &= 6 \\ f(0,-3) &= 30 \\ f(\sqrt{5}, 2) &= f(-\sqrt{5}, 2) = 5 \end{aligned}$$

### בחזרה ל-SVM

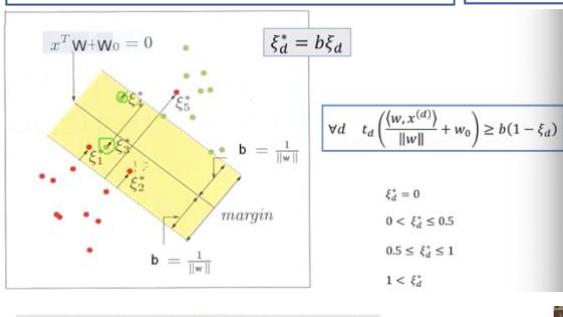
זכור את המטריה שלנו למצער את  $t_d(\mathbf{w} \cdot \mathbf{x}^{(d)} + w_0) - 1 \geq 0 \quad \forall d$ . נשים לב כי לגרנו' לא אומר לנו שום דבר לגבי אי-שוויון. בנוסף, אין התייחסות לקיים התנאי עבור כל האילוצים לכל הנקודות. אז קיימת הרחבה אשר אומرت שמשפט לגרנו' מתקיים גם עבור אי-שוויונים וגם עבור אילוצים על כל נקודות הדאטה  $d$ .

$$\begin{aligned} \max_{M, \mathbf{w}, w_0} \quad & M \\ \text{subject to} \quad & \forall d \quad t_d \left( \frac{\langle \mathbf{w}, \mathbf{x}^{(d)} \rangle}{\|\mathbf{w}\|} + w_0 \right) \geq M(1 - \xi_d) \\ & \xi_d \geq 0, \quad \sum_{d \in D} \xi_d \leq C \end{aligned}$$

$$\begin{aligned} \min_{\mathbf{w}, w_0} \quad & \|\mathbf{w}\|^2 \\ \text{subject to} \quad & \forall d \quad t_d \left( \langle \mathbf{w}, \mathbf{x}^{(d)} \rangle + w_0 \right) \geq (1 - \xi_d) \\ & \xi_d \geq 0, \quad \sum_{d \in D} \xi_d \leq C \end{aligned}$$

### משתני Slack

ונניק לכל נקודות דאטה  $d$ , משתנה Slack שמוסמן באוט קסי היונוגי (נראית כמו נחש), שהוא אי שלילי. ונוצרת לנו בעיית אופטימיזציה דומה, מממד שונה – מממד אחד יותר, יש לנו קסי לכל דגימה.



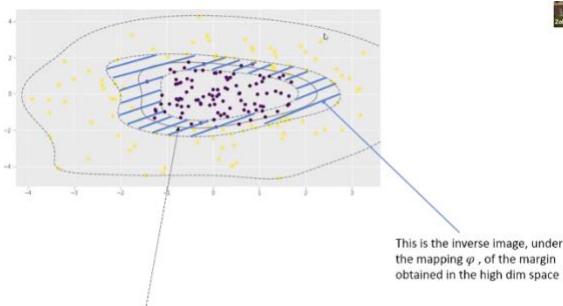
### חשיבותו של משתני Slack:

כל שהנתיכון קטן, הקסי קטן וככל שהנתיכון גדול, הקסי גדול.

### SVMs סיכום

האלגוריתם המבוצע שלנו דורש support vectors, מקדמים וקרנל:

$$class(\vec{x}) = \operatorname{sgn} \left( \sum_{d \in SV} a_d t_d K(\vec{x}_d, \vec{x}) \right)$$



- הלמידה דומה לקרנל פרספטורון אבל משתמשת ב- convex optimization ובירשת לגרנו' / KKT (הרחבה של לגרנו').
- בלמידה אנחנו מוצאים מפריד ליניארי בעל שלולים וחבים במרחב מממד גבוה.
- Kernel מיציג את המכפלת הפנימית בממד הגובה.
- SVMs מאפשרים מיס-קלסיפיקציות במידה מסוימת שנשליטה על ידי היפר-פרמטר.

## מבוא לתיאורית הלמידה / Introduction to Learning Theory

### במידה חישובית מידי-ע

אנחנו מודדים היפוטזה  $H$  מරחב היפוטזות  $H$  למידע שאנו רואים (training set). לכן, זהה בעית שיעורך – אנחנו נרצה שהטעות של ההיפוטזה שלנו תהיה קטנה ככל שאפשר על ה- $h$ . training set-error. לרוב זה נקרא "in-sample-error". אבל, במידה אנחנו לא באמת מתעניינים ב- $"in"$  (generalization error) אלא בטעות שאנו לא רואים! זה נקרא "out-of-sample-error" או טעות ההכללה (out-of-sample-error).

### שיעורך מול הכללה / Approximation vs. Generalization

- **שיעורך / Approximation** : מודד כמה טוב ההיפוטזה מודדת את ה-training data.
  - **הכללה / Generalization** : מודדת כמה טוב ההיפוטזה צפואה למודל דעתה חדש.
- במידה אנחנו מעוניינים בהכללה וכן תחילה הלמידה הוא קשה. אנחנו נרצה דרך דריך את הביצועים של ההכללה מתוך ה-data.sample
- סיבוכיות הדוגמיה – כמה training data נחוצה עבור רמה מסוימת של ביצועים.

**דוגמה:** למידת פונקציה בוליאנית

$f(x_1, x_2, x_3, x_4) = t \in \{0,1\}$

נרצה ללמד פונקציה בוליאנית (קלסיפיקציה בינה-רית) מעל 4 משתני אינפוט .label.

ונתנו לנו 7 דוגמאות – training examples, עליהם אנחנו יודעים את ה-label. במרחב שלנו יש 16 נקודות, ומכיון שננתנו לנו 7, יש לנו 9 נקודות שאנו לא יודעים עליהם כלום – יש לנו  $2^8 - 7 = 255$  דיכוטומיות (פונקציות בוליאניות) אפשרויות שהן קוניסטנטיות עם ה-data.

Example	$x_1$	$x_2$	$x_3$	$x_4$	$y$
1	0	0	1	0	0
2	0	1	0	0	0
3	0	0	1	1	1
4	1	0	0	1	1
5	0	1	1	0	0
6	1	1	0	0	0
7	0	1	0	1	0

Note:  
 $X = \{0,1\}^4$  and therefore  $|X| = 16$ .  
If all Boolean functions are in our hypotheses space  $H$  then  $H = 2^{16}$

### אין ארוחות חינם / "No Free Lunch"

- טעות ההכללה (שמסומנת  $Err_{GEN}(h)$ ) של היפוטזה היא ממד הטעות של  $h$  עבור כל ה-non-training examples.
- יהי  $F$  מרחב כל הקונספטיים האפשריים עבור  $(x, f) = y$  שהן קוניסטנטיות עם training dataset מסויים.

**משפט:** עבור כל היפוטזה  $h$ , טעות ההכללה הממוצעת(\*) מעל כל הקונספטיים ב- $F$  היא  $0.5$  • כאשר ממוצעת = בהנחה שכל היפוטזות הקוניסטנטיות  $h$  הן סבירות במידה שווה.

*	$(f_3)^*$	$(f_{257})^*$
$f_1$	$f_2$	$f_3$
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1
$f_{257}$	$f_{258}$	$f_{259}$
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

**No free lunch - example**

**הסבר עבור המשפט אין ארוחות חינם:** יש 512 הרחבות אפשריות ל-training data. נתבונן בהרחבת מספר 3 שמוסמנת ב- $f_3$  ונדיר את  $(f_3)^*$  להיות – הנגטיב של  $f_3$ . כמה טיעויות  $f_3$  יכולות לעמוד?

במקסימום 9 ובמינימום 10. לעומת זאת הטיעויות של  $(f_3)^*$  משלימות את הטיעויות של  $f_3$ , כך שאמם  $f_3$  עשתה 4 טיעויות,  $(f_3)^*$  עשתה 5 טיעויות. לכן, מסגר הטעויות הממוצע של  $f_3$  ו- $(f_3)^*$  הוא בדוק 4.5 מכיוון שלכל היפוטזה יש את "ההיפוטזה-תג" שלה, ההנחה של אין ארוחות חינם היא להציג לכל היפוטזה את ההיפוטזה הפוכה לה, וכך הסתברות השגיאה על כל הרחבות היא חצי.

- משפט ה-NFL מוכיח כי כל הקונספטיים שהם קוניסטנטיים עם ה-data training הם בעלי סבירות שווה בהינתן ה-data.
- במצבות לא כל הקונספטיים סבירים באותה המידה.

**תופעות ריאליות:** אין דגימות שמתפלגות יוניפורמיות בכל הרחבות האפשריות של הדאטה. קונספטיים אמתיים (טבעיים, פרי- אדם, סוציאולוגיים) הם בעלי רגולריות, חוקים, מבנה.

**דגימות training עם ערכי-אטריביטות נתונות/notations:** דגימות training עם ערכי-אטריביטות נתונות/notations לא אינדיקטיבית לבני הקלאס האמתי של דגימות non-training – אטריביטות דומיים. כפי שראינו – אנחנו מנצלים תוכנות אלה.

**ה- $k$ -up-set הכללי שלנו:** נוסה ללמידה קונספט  $C$  (פונקציה, קלסיפיקציה – דיכוטומיה: כאשר  $C$  מוכל במרחב הדגימות). יש לנו  $k$  דגימות ממרחב האינסטנסים  $X$ . נרצה לחציע היפוטזה (אלגוריתם מבצע) בעל צורה מאפיינית בהתאם למשימה. עבור מסווגים אנחנו משתמשים במרחב ההיפוטזות  $H$  – אשר משרה ות-קבוצה של קבוצת החזקה של  $X$ .

Rule	Counterexample
$\Rightarrow y$	1
$x_1 \Rightarrow y$	2
$x_2 \Rightarrow y$	3
$x_3 \Rightarrow y$	1
$x_4 \Rightarrow y$	7
$x_1 \wedge x_2 \Rightarrow y$	3
$x_1 \wedge x_3 \Rightarrow y$	3
$x_1 \wedge x_4 \Rightarrow y$	3
$x_2 \wedge x_3 \Rightarrow y$	3
$x_2 \wedge x_4 \Rightarrow y$	3
$x_3 \wedge x_4 \Rightarrow y$	4
$x_1 \wedge x_2 \wedge x_3 \Rightarrow y$	3
$x_1 \wedge x_2 \wedge x_4 \Rightarrow y$	3
$x_1 \wedge x_3 \wedge x_4 \Rightarrow y$	3
$x_2 \wedge x_3 \wedge x_4 \Rightarrow y$	3
$x_1 \wedge x_2 \wedge x_3 \wedge x_4 \Rightarrow y$	3

Example	$x_1$	$x_2$	$x_3$	$x_4$	$y$
1	0	0	1	0	0
2	0	1	0	0	0
3	0	0	1	1	1
4	1	0	0	1	1
5	0	1	1	0	0
6	1	1	0	0	0
7	0	1	0	1	0

variables	Counterexample			
	1-of	2-of	3-of	4-of
$\{x_1\}$	3	—	—	—
$\{x_2\}$	2	—	—	—
$\{x_3\}$	1	—	—	—
$\{x_4\}$	7	—	—	—
$\{x_1, x_2\}$	3	3	—	—
$\{x_1, x_3\}$	4	3	—	—
$\{x_1, x_4\}$	6	3	—	—
$\{x_2, x_3\}$	2	3	—	—
$\{x_2, x_4\}$	2	3	—	—
$\{x_3, x_4\}$	4	4	—	—
$\{x_1, x_2, x_3\}$	1	3	3	—
$\{x_1, x_2, x_4\}$	2	3	3	—
$\{x_1, x_3, x_4\}$	1	***	3	—
$\{x_2, x_3, x_4\}$	1	5	3	—
$\{x_1, x_2, x_3, x_4\}$	1	5	3	3

Example	$x_1$	$x_2$	$x_3$	$x_4$	$y$
1	0	0	1	0	0
2	0	1	0	0	0
3	0	0	1	1	1
4	1	0	0	1	1
5	0	1	1	0	0
6	1	1	0	0	0
7	0	1	0	1	0

בחזורה לדוגמה שלנו של הפונקציהabolianitis: נגביל את מרחב ההיפותזות שלנו, כך שלא יהיה לנו יותר 512 הרחבות אפשריות של-h-training data. יש רק היפותזות מסווג מסוים הן כשרות, נניח כי גימום conjuctions (משפט וגם) מגדיר את הקונספט שלנו ונציג את כל משפטיו "וגם" האפשריים עם 4 משתנים.

"וגם" הוריך לא מייצג את הדעתה שלו מפני שיש עבורו דוגמאות נגדיות. ממשיך ונשלול את כל משפטי "וגם" האפשרים הנ"ל, מפני שעבור כלם יש לנו ב-training הדוגמאות נגדיות, ונגיע לכך שאנו מודל מושך מרחב ההיפותזות שגדלו שהוא קונסיטנטי (邏輯) עם הדעתה (the-training).

לכן, ננסה להגדיר מרחב ההיפותזות אחר,  $\text{out-of}$ -set, ונבדוק עבורי. להלן הדוגמאות נגדיות עבור מרחב ההיפותזות הנ"ל –

ונראה כי ככל אחת מהאפשרויות נקבל דוגמאות נגדיות.

עבור ה-\*\*\* אנחנו כן מסכימים עם הדעתה! רק אחד מהם הוא קונסיטנטי עם הדעתה – למדנו את הקונספט האמתי בהנחה שאין טעויות ובהינתן מרחב ההיפותזות מוגבל!

אבל הגענו לקונספט האמתי בכך שהגבילנו את מרחב ההיפותזות שלנו. ישנו עוד מודלים שהם קונסיטנטיים עבור הדעתה שלנו.

### הגבלת מרחב ההיפותזות:

עיי בחירת סוג הפונקציה אנחנו מגבילים את מרחב ההיפותזות. להלן חסרונות ויתרונות:

- חסרון: הפונקציה/קונספט האמתיים עשויים לא להיות שייכים בכלל למרחב ההיפותזות שבחרנו, וכן היפותזות מורכבות יותר אין תמיד טובות יותר.

- יתרון: היפותזות פשוטות הן טובות יותר עבר הכללה. בכך אנו מנעמים מ-overfitting, למרות שהදעתה שלנו יכול ל לכלול טעויות. הן יכולות "لتפוצס" מבנה חבויה ועל כמה אפשרות למידה, והן יותר קלות יותר למידה ( מבחינה חשיבות).

### מרחבי היפותזות באლגוריתמי למידה שלמדנו:

הטעות האמתית (במסונגים)

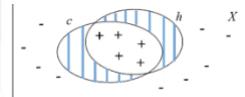
היא ההסתברות לMISS-קלסיפיקציה.

הטעות האמתית של היפותזה  $h$  ביחס ל-target concept  $c$ , היא ההסתברות ש- $h$  יטוגן באופן שגוי דוגימה שהוגרלה רנדומלית מתוך התפלגות הדעתה  $F$ :

$$\text{error}_F(h) = \Pr_{x \sim F}[c(x) \neq h(x)]$$

הטעות תלויה ביותר בתפלגות הדעתה  $F$ !

Algorithm/Problem	Hypothesis Space
Regression	Linear functions, polynomial functions etc
Decision Trees	Trees/axis aligned boundaries
Perceptron	Linear decision boundary (+ we can map to higher dimensions)
KNN	Complex separating boundaries – depends on K
SVM	Separating hyperplane in $\varphi$ -space (depends on Kernel choice)
Logistic Regression	A linear decision boundary (+ we can map to higher dimensions)



### שיעורון טעויות

- שאלת המפתח שעולה כאן היא: האם טעות ה- $\text{out-of-sample}$ ? יכולה להגיד לנו מושdot טעות ה- $\text{out-of-sample}$ ? ובאופן יותר כללי:

האם נוכל להגדיר ולומר ממהו על הטעות האמתית או על הטעות המוצפפת / expected error של ההיפותזה שלנו?

כאר נפתח את תיאורית סיבוכיות הדגימה נבדיל בין שני המקרים הבאים:

1. הקונספט האמתי נמצא במרחב ההיפותזות.
2. הקונספט האמתי אינו במרחב ההיפותזות.

- נשתמש ב-test set כדי לשער את הטעות האמתית של היפוטזה מועמדת. אם ה-test set הוא יתרת  $X$  (מרחב הדגימות) אז נדע את הטעות האמתית! כמו זה מקרה לא ריאלי – אנחנו חיקיבים להסתמך על הדגימות ונגידר את טעות הדגימה, עבור קבוצת דגימות

באופן הבא:  $\text{error}_S(h) = \text{the ratio of misclassified samples in } S$

כל ש-S גודלה יותר כך ההערכה טובה יותר.

### **תהליך שיעור סטטיסטי / Statistical Estimation Procedure**

נשתמש ב-test set בגודל  $|S|$  ונניח כי ראיינו  $k$  שגיאות. נוכל להראות כי משערך עבור טעות ההכללה יהיה  $|S|/k$ . כאשר אנו תלויים בגודל של ה-test set שלו, נוכל לקבל הבטחה סטטיסטית כמו: בודאות של 95% אנו יודעים שהשגיאה האמיתית היא קטנה מ-  $\text{eps} + |S|/k$ . זהה וובע מחייב אינטראול הودאות.

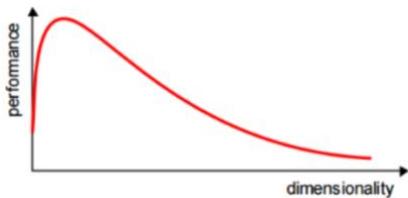
#### **לסיום הרצאה:**

- NFL Thm
- Hypothesis spaces and how they allow for learning
- “out-of-sample” error and “in-sample” error
- Statistically assessing errors

## הממדית חישובית מיפוי – תרגול 8 – הורדת ממד

כאשר אנחנו מדברים על ממד, אנחנו מתחווים לממד של מרחב הדגימות שלנו – כמובן, כמות הפיצ'רים שיש לכל דוגימה היא הממד בו אנחנו עוסקים. הכוונה בהורדת ממד היא להצליח להביא את הדגימות שלנו ממספר גובה לממד נמוך, ולבצע את הלמידה בממד הנמוך.

**למה שורצחה להוריד ממד?**



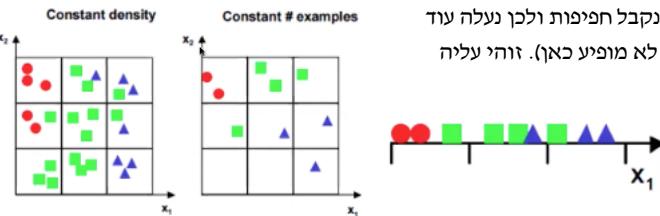
- סיבה מותבקשת אחת היא שזמני הריצה והחישוב יתקצרו – כתוצאה ישירה של הקטנת הוקטורים שאוטם אנו לומדים.
- מבחרת מודלייזציה – המידע שאנו שומרים במרחב מממד נמוך יותר הוא פחות "משמעותי" כך שחישוב המרחקים, למשל, בין נקודות הדאטה נעשה פשוטה. בנוסף צמצום ממדים מפחית את סיבוכיות הדגימה (מספר הדגימות שנוצרך כדי להבטיח רמת ביצועים מסימית על DATA חדש). וכן, תמיד נרצה להציג פשטוט יותר של הדאטה כדי להמנע מ-overfitting.

סיבה נוספת שורצחה לעשות זאת היא שרוב המידע המשמעותי עבורנו כדי ללמידה, טמון ביחסית מעט פיצ'רים. הוספת הפיצ'רים טובת עד לנוקודה מסוימת – שלאחריה הדיק והביצועים של המודל שלו מתחילה לדעך – עקרון זה נקרא **the curse of dimensionality**. הסיבה לכך נובעת מהעובדה שהדאטה הופך להיות "דיליל" יותר ככל שעולים במדדים. לכן, כדי לשמור על אותה צפיפות גם כשנוסיף פיצ'רים (= ובכך נגדיל את הממד) נדרש להעלות את כמות הדגימות (סיבוכיות הדגימה עולה). העניין הבסיסי הוא שהיחס הזה אקספוננציאלי – על כל פיצ'ר נוסף נדרש להעלות ממשמעותית את כמות הדגימות. לכן אפשר להבין שורצחה לעבוד במדדים נמוכים יותר (וגם זה, עד גבול מסוים).

x1	x2	x3	x4	y
10	14	24	1	1
50	2	5	5	1
5	16	30	0.5	2
30	10	2	3	3
10	23	21	1	2

בדוגמה זו ניתן לראות כי פיצ'רים 1 ו-4 הן קומבינציה לינארית אחד של השני ועל כן ניתן לוותר על אחד מהם ובכך להוריד ממד מפני שהם לא תורמים לנו, הם מכילים את אותם הערכים במכפלה סקלרית.

בדוגמה השנייה, נניח שורצחה להוסיף פיצ'ר לדאטה בו ישנו 3 אטרוביוטים, מפני שיש כמעט כמעט וחיפה ואינו ממש הפרדה. נחליט להעלות לממד בעל 9 אטומים. התקבל שם כל דليلים שיחיה עליינו להוסיף דגימות כדי לשמר על אותה צפיפות – 27 דגימות. וכך במודל זה מקבל חיפפות ולכן נעה עוד ממד ואף במודל זה כדי למנוע דילילות נוצרך להגיע ל-81 דגימות (היאור לא מופיע כאן). זהה עלייה אקספוננציאלית כפי שנאמר לעיל.



נראה שני תהליכי להורדת ממד – כמובן לדילול מספר הפיצ'רים, בחירת פיצ'רים / Feature Selection ו-**חילוץ פיצ'רים** / Feature Extraction. הבדל המהותי בין שתי השיטות בהינתן דוגמה חדשה הוא מבחינת הסיבוכיות – ולרוב בחירת פיצ'רים ייצח מבחינת יעילות.

### בחירה פיצ'רים – Feature Selection

נניח שאנו מתחווים עם  $n$  פיצ'רים. שיטה זו בוחרת תת-קובוצת, בוגד קבוצה מ- $n$  (למשל  $m$ ), של פיצ'רים מבין הפיצ'רים המקוריים – אלו הפיצ'רים שיהיו הכי משמעותיים עבורנו כדי ללמידה. **בדיעבד זאת קיימות 3 גישות עיקריות:**

#### Wrapper •

בדרכ זו, נגייע לתת קבוצה של פיצ'רים עיי' בחינה של כל תת-קובוצות האפשרות. זה לא באמת אפשרי, כי אם כך האלגוריתם יהיה אקספוננציאלי (יש  $\binom{n}{m}$  תת-קובוצות אפשריות של פיצ'רים). לכן, האלגוריתם יבודד בצורה גרידית. **נתחיל עם קבוצה ריקה, וכל איטרציה נבעצט.**

1. נסיף כל פיצ'ר בנפרד לקבוצה, ונitin לאלגוריתם הלמידה ללמידה את הדאטה באמצעות קבוצת הפיצ'רים.
2. לבסוף נסיף באמות את הפיצ'ר שנותנו לנו את התוספת הכי משמעותית לדעך.

ניתן לבצע את האלגוריתם הזה עיי' התחלה בקבוצה ריקה והוספה פיצ'רים (**forward**) או להפוך, עיי' התחלה מקבוצה כל הפיצ'רים והסרת פיצ'רים שהיעדרותם תפגע היכי מעת בדיק (backward). נעצור את האלגוריתם כאשר נגיע לממד מסוים שהגרנו, או לרמת דיקוק מסוימת שרצינו להגיעה אליה. יש כאן חסרון ויתרונו בכך שעליינו לבחור אלגוריתם למידה שיחשב לנו דיקוק / שגיאה. היתרונו הוא שנבחר פיצ'רים היכי טובים לביצוע שלו. החיסרונו הוא שצורך לבחור אלגוריתם – מה שיוצר מצב של "ביצה ותרנגולת".

אסטרטגיית חיפוש (פיצ'רים טובגים) ווסף עבור גישה זו יהיה: חיפוש **.bidirectional** רק על חלק מהמטרה וחיפוש אקספוננציאלי על החלק

- **backward** בין forward ו- **backward** לשלב בין forward ו- **backward** (קצת מזה קצר מזה ושוב קצר מזה), להתחילה את שתי השיטות (forward ו- **backward**) ולהיפגש באמצע, ולהשתמש ב- **random** בתהליך הדבר עשוי להוביל מפני שכל האסטרטגיית הללו בפועל אכן אופטימלית ולמן יתכן שימוש ברנדום ווביל אותו לפתרונות טובות יותר.

## Filter

הדרך השנייה לא מותיחסת לאלגוריתם למידה ספציפי, אלא בוחרת נתן קבוצה באופן גנרי. זה גם חיסרונו, מכיוון שהוא יכול להיות שמודל מסוים יעדיף פיצ'רים מסוימים, ו- **filter** לא מתחשב בכך. לכן, צריך להיות מודעים לפיצ'רים שאחנו מורידים באמצעות שיטה זו. כדי לבחור את נתן הקבוצה, נחפש מטא (קורלציה) בין הפיצ'רים לפונקציית המטריה (קלאסים, מחיר הבית וכדומה) – כמובן, אם נמצא פיצ'רים שיש ביניהםween בין פונקציית המטריה מטא, זה אומר שלמידה שלهما תלמד אותנו הרבה על הדאטה – לכן נרצה לכלול אותם בתנתן הקבוצה שלנו, ולהציגו ממנה פיצ'רים עם מטא לפונקציית המטריה. את המכדידה הזאת טبعי לבצע באמצעות מקדים

$$-1 \leq \rho \leq 1 : \text{(Pearson correlation)}$$

### – בודק את המתאים הילינרי בין שני משתנים מקרים / וקטוריים – Pearson Correlation

דווגמה :

$$\rho = \frac{\sum_{d=1}^m (x_k^{(d)} - \mu_k)(y^{(d)} - \mu_y)}{\sqrt{\sum_{d=1}^m (x_k^{(d)} - \mu_k)^2 \sum_{d=1}^m (y^{(d)} - \mu_y)^2}} = \frac{\sigma_{x_k y}}{\sigma_{x_k} \sigma_y}$$

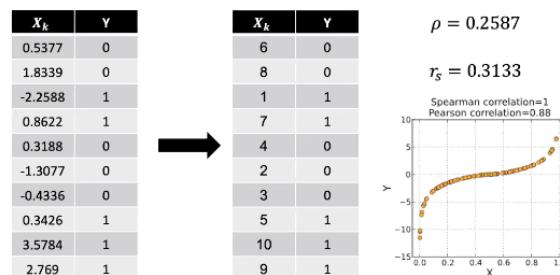
$x_k$	$y$	$x_k - \mu_k$	$y - \mu_y$	$\mu_k = 0.6243$	$\mu_y = 0.5$	$\rho = 0.2587$
0.5377	0	-0.0866	-0.5			
1.8339	0	1.2096	-0.5			
-2.2588	1	-2.8831	0.5			
0.8622	1	0.2379	0.5			
0.3188	0	-0.3055	-0.5			
-1.3077	0	-1.932	-0.5			
-0.4336	0	-1.0579	-0.5			
0.3426	1	-0.2817	0.5			
3.5784	1	2.9541	0.5			
2.769	1	2.1447	0.5			

ככל שניה קרובים לאחד הקצאות נקבע מתאם חזק יותר (חיובי או שלילי), כאשר  $\rho = 0$  מעיד על חוסר מתאים מוחלט.

הчисIRON בפירסן הוא שיש חישוב רבה מאו על ערכיהם עצם של הדאטה ופונקציית המטריה, ולכן הוא רגישות outliers – רגישות לרעיש. כתוצאה לכך, יכול להתקיים מטא בין פיצ'רים מסוימים לפונקציית המטריה אבל לא נראה זאת זה במקדים המתאים. לכן, קודם כל נמיין את הדאטה (motion), נalive כל ערך באינדקס שהוא קיבל אחריו המינו ('הדרגה שלוי'), ועבור הערכים האלה נמדד מטא לפונקציית המטריה באמצעות פירסן.

– Spearman rank correlation – ממיין את הדאטה, ומבעץ פירסן על האינדקס שקיבלו לאחר המינו = הדרגה, לכן אנחנו עדין נמצאים בטוחה שבין  $-1$  ל-  $1$ . אבל, הפעם המשמעות של כך הינה מונוטוני עולה, כלומר ספירמן בודק מונוטוניות בדאטה.

דווגמה :



## Embededeb •

**תוקן כדי תחליך הלמידה האלגוריתם ייחס "ציון חשיבות" / importance score עבור כל פיצ'ר.** האלגוריתמים שראינו שיודעים לחשב חשיבות של פיצ'ר הם: **עץ החלטה** – שמחשב את split-goodness, **ורוגסיה** – כאשר הפיצ'רים הם בעלי אותה הסקללה ככל שהמשקל שנותנו לפיצ'ר גובה יותר כך הוא יותר חשוב עבור תחליך הלמידה, יותר משמעותי.

### חילוץ פיצ'רים – Feature Extraction

גם כאן, נניח שאנו מתחילה עם **n** פיצ'רים. נרצה להוריד את **n** מפיצ'רים (להוריד מימד) ועם זאת לשמר כמה שיותר אינפורמציה. על מנת לעשות זאת אנחנו צריכים להגיד איך אנחנו מוחדים את **n** מפיצ'רים ששלו – דבר שמדד אחד עברו יהיה השוואות של הדאטה: ככל שהשוואות יובחו יותר (הדאטה מפוזר יותר) יש לנו יותר אינפורמציה, לעומת השוואות נמוכה והדגימות ידומות זו לזו. בuit, נרצה להטיל את הדאטה למימד חדש, נמוך יותר, **ודעין לשמר על השוואות** שלו. תחליך זה נקרא **projection** = תחליך העברת ושינוי נקודות הדאטה מערכות צירים אחרות והוא פועל **בשני שלבים**:

1. נביא את הדאטה במימד המקורי שלו למוציע/תוחלת  $\mu$ , ככלمر למרכזו הראשית הציגים – נעשה זאת ע"י חישור הממוצע/התוחלת של כל בעמודה/פיצ'ר (יקטור התוחלות/הממוצעים).

דווגמה עבור **3** דוגמאות בעלות **2** פיצ'רים:

$$\begin{bmatrix} x_{1,1} & \dots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \dots & x_{m,n} \end{bmatrix} \xrightarrow{\text{substracting the mean}} \begin{bmatrix} x_{1,1} - \mu_1 & \dots & x_{1,n} - \mu_n \\ \vdots & \ddots & \vdots \\ x_{m,1} - \mu_1 & \dots & x_{m,n} - \mu_n \end{bmatrix}$$

2. הטלחה עצמה על המימד החדש – באמצעות מכפלה פנימית של הוקטור החדש, עם מטריצת הטלחה A (שהיא בעצם הציגים של המרחב החדש)

PCA – algorithm

- Build the covariance matrix  $S$
  - Find Eigenvectors and Eigenvalues of  $S$  by solving  $(S - \lambda_i I)a_i = 0$
  - Sort Eigenvectors by Eigenvalues
  - Build the matrix  $A_j$  using  $j$  eigenvectors with the  $j$  greatest eigenvalues
  - Transform  $x' = A_j(x - \mu)$
- cut נראה מהי מטריצת הטלחה A :  
(בתרגול שעה ורבע / שעה ועשרים כזה)

### Analysis Component Project - PCA

$X$  – הדאטה לאחר הזזת הממוצע ל- $0$ ,  $n$  – המימד החדש שאליו אנחנו רוצים להוריד את הדאטה (לכן נשאף לכך ש-  $n < k$  כמה שאפשר). נגיד את מטריצת הטלחה הייתה:

$$A = [a_1, a_2, \dots, a_k] \in \mathbb{M}_{n \times k}$$

ובנוסף נדרוש שהמטריצה תהיה **אורותונורמלטיב** (כל הוקטוריים מנורמליים ומאונכים זה לזה) – מכפלה פנימית של כל זוג וקטורים היא  $0$ , ומכפלה פנימית של כל וקטור עם עצמו היא  $1$ . מכך נבע ש-  
cut עלינו למצוא את הוקטוריים  $a_1, \dots, a_k$  ש碼ריכים את המטריצה. כפי שאמרנו בהטלחה, על מנת לשמור את האינפורמציה בצורה מיטבית, אנחנו רוצים למקסם את השוואות של הדאטה במימד החדש. ניקח וקטור דואטא כללי במימד החדש,  $x'$ . עברו כל רכיב  $x'_i$  ( $i \in \{1, \dots, k\}$ , מתקיים:

$$Var(x'_i) \underset{\text{def}}{=} E[(x'_i - \mu_i)^2] \underset{\forall i, \mu_i=0}{=} E[x'^2_i] \underset{\text{proj.}}{=} E[(a_i^T X^T)^2] \underset{E \text{ is linear}}{=} a_i^T E[X^T X] a_i = a_i^T S a_i$$

כאשר **מכוון שהמוצע של כל פיצ'ר הוא 0**, מתקיים ש-  $X^T X = S$  – מטריצת השוואות המשותפת (covariance matrix) של הדאטה (עד כדי כפל בסקלר).

עשינו אנחנו רוצים למצוא את הוקטור  $a_i$  שמקסם את השוואות של  $x'$  תחת האילוץ שנובע מהאורותונורמלויות –  $a_i^T a = 1$ . זהה בעיית אופטימיזציה שנטוור בעזרת כופלי לגראנג' (פירוט בתרגול), שבסופה נגלה שהוקטוריים שאנו מקסמים את השוואות של כל רכיב הם הוקטוריים העצמיים של  $S$ . הוקטוריים ששמוראים על האינפורמציה המירבית, הם הוקטוריים עם הערכים העצמיים הגבוהים ביותר – לכן נבחר כמה לחתת (תלויה כמה נרצה לרדמת במימד), נבנה מהם את מטריצת הטלחה, ונוכל לבצע את הטלחה למימד החדש.

## למידה חישובית מכייען – הרצאה 9 – תיאוריה

### General Setting

- דוגמאות מגוונות מתחום  $\Omega = (X, Y, P)$  כאשר  $X$  – הוא מרחב הוקטוריים במינימ פיזרים ( $m, Y$  הם התכפיות,  $h$ -labels,  $P$ -value).
- והינה התיפלגות של הוקטוריים והמטרות כך שההתפלגות המשותפת הינה ממינימ פיזרים  $m + 1$  (כלומר  $+1$ ). אנחנו נ שאף שההתפלגות זו תהיה תלואה! אחרת אין לנו איך למדוד. אבל הדוגמאות עצמן ב-training data-הן אכן בלתי תלויות.

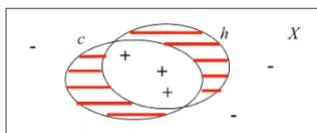
אלגוריתם הלמידה  $L$  לוקח training data, מתחום  $D \in \Omega^m$ .

האלגוריתם למידה עובד עם סט היפותזות  $H$  מרחב ההיפותזות.

האלגוריתם מחיר היפותזה/מודול  $L(D) = h \in H$  (האינפוט הוא  $h$ -data)training data והאינפוט הוא היפותזה)

תאוריות "איו אורות חינט" תקפה כאשר אין לנו מרחב היפותזות ואז אי אפשר למדוד.

**טעות האמיתית / The True Error of  $h$**  – נניח כי קיימת התפלגות מעלה  $X$ . נגיד:  $c(X) \neq h(X)$  (TrueError( $h$ ) = error<sub>D</sub>( $h$ ) = Prob<sub>X~D</sub>( $c(X) \neq h(X)$ )).  
כאשר  $D$  בנוסחה זו מייצג את התפלגות מעלה  $X$ , מסומן לעיל ב-(P)



### שיעורון סטטיסטי של טעות הסיווג / Statistical Estimation of the Classification Error

נשתמש ב-set test בגודל  $SI = n$ , נניח כי ספרנו  $r$  טויות. נ שערק את טעות ההכללה על ידי:

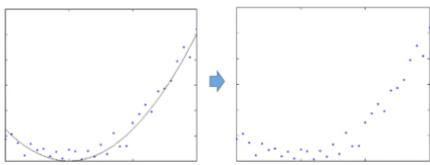
מתאorigiy סטטיסטיקת הדוגמאות נוכח להניח  $95\%$  רוח סמך (Confidence Interval) CI – עבור טעות ההכללה:

$$se = \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$$

כאשר  $se$  הוא standard error-h-ה-errror.

כל ש- $n$  יותר גודל הטעות הסטנדרטית קטנה. כאשר הספרה 2 באינטראול נובעת מהעובדת כי החלטנו על וודאות של  $95\%$  (אם היינו רוצים וודאות של  $99\%$  כנראה ספרה זו הייתה מגעה ל- $3\sim$ ). וודאות זו משמעותה הסבירות שנקבל טעות מוחץ לטובה. חישוב טעות זו על training data, יקטן ככל שנרחיב את המורכבות של מודול.

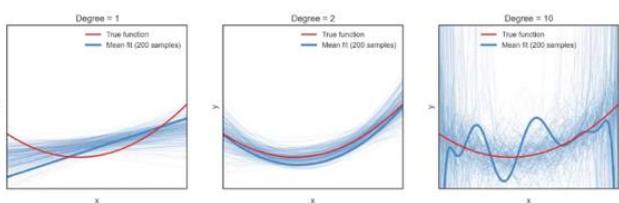
### Bias Variance Decomposition



דוגמה ברגression: הפונקציה (הקוונספט C) היא סוג של פונקציה ריבועית וניקח דוגמאות עברו ה- $h$  training set-ה-errror עם רוש. הלמידה גם תניב פונקציה –  $h$  היפותזה. אנחנו חיברים להניח "עולם" על מרחב ההיפותזות מכיוון שאחרות יהיה ב-NFL ולא יוכל למדוד את המודול. אם נניח "עולם לינארי" על מרחב ההיפותזות, לא משנה כמה DATA יהיה לי, תמיד תהיה טעות. ו אם נעבד על "עולם ריבועי" גם לא בהכרח נמצא את  $h$  perfect fit מפני שם הדאטה שלנו לא בהכרח מייצג את העולם, אבל ככל שנגדיל את הדאטה, נתקרב יותר.

**נתחל את ניתוח / Analysis set-up**: בהינתן המודל האמתי,  $c$ , נגריל ממנו דוגמאות עברו ה- $h$ ,  $D$ , training data, ונכנסו אותו לאלגוריתם למידה  $L$ , על מרחב היפותזה  $H$ . האלגוריתם יניב מודול  $h(x|D) = L(D)$  כך שניתן להגדיר אותה באופן הבא –  $h(x|D)$  (ההדגשה כאן ש- $h$  היא פונקציה אשר תליה ב- $D$ ). כדי להבין את כוח גישת הלמידה שלנו נרצה למצוא (לממצא את התוחלת), עבור כל אינפוט וקטור קבוע של פיזרים  $x$ ,

$$\begin{aligned} y &= h(x|D) && (\text{thin blue, individual } 200 \text{ samples } D) \\ y &= E(h(x|D)) && (\text{thick blue, average of } 200 \text{ samples } D) \\ y &= c(x) && (\text{red}) \end{aligned}$$



ולחשב את  $E_D([h(x|D) - c(x)]^2)$ . כאשר עברו כל  $D$  נקבל מספר אחר, ולכן שיש בתוכו הריבוע הוא למעשה משתנה מקרי (משתנה מקרי פחות מספר, מהיות ש- $c$  אינו תלוי בדאטה, וזה הקונספט האמתי). לכן נרצה לחשב את התוחלת שלו, להעלות אותה בריבוע ועליה לחשב את הטעות. למעשה, זה הינו התוחלת של ההפרש בין מה שייצר האלגוריתם הלומד  $L$  עברו הדאטה, לבין הממציאות.

**בחזרה לדוגמה**: הגרף האדום מייצג את הקונספט האמתי. אנחנו מתבוננים ב-200 קווים דקים כחולים מייצגים את היפותזות שהאלגוריתם הלומד  $L$  הניב עבורה. הקווים כחולים העבירה מייצג את תוחלת (המייצוע של training datas 200).

$H$  = linear functions  
 $m = 10$

$H$  = quadratics  
 $m = 10$

$H$  = polynomials of deg 10  
 $m = 10$

- חן training datasets D בגודל m, שהגנו לפि התפלגות בלתי תלואה מעל אוכלוסיות הדאטה כדי לשערק את הביצועים של אלגוריתם הלמידה L שמפיק את ( $D | x$ ) h כאוטופוט עבור אינפוט dataset D, רצחה לשערק את:
- $E_D([h(x|D) - c(x)]^2)$  זהו ה-**expected squared error**
  - מה-sh-**training datasets** מתחום  $\Omega = (X, P)^m$ . (נשים לב ש-ED מסמן תוחלת)

#### מושגים חשובים עד כה

- = הקו הכהול העבה  $E_D[h(x|D)]$  = **the expected (or mean) prediction value at  $x$**
- = ההפרש בין הקו הכהול העבה לקו האדום בנקודה  $x$  = **Bias** =  $E_D[h(x|D)] - c(x)$  = **the expected prediction vs. true value, at  $x$**
- = התוחלת של (משתנה מקרי פחות התוחלת שלו) בריבוע, היא **השונות** של הבדיקה ב- $x$ . השונות מחושבת בנקודה מסוימת והיא מחשבת את המרחק בין קו כחול דק לקו הכהול העבה בנקודה  $x$ .
- = **משפט**: תוחלת הטעות בריבוע ב- $x$  נתון מרכיבת משני רכיבים : הביאס בריבוע והשונות.

### Decomposition for Squared Loss

For a given instance  $x \in X$  we use the shorthand  $h(x) = h(x|D)$ .  
All expectations are with respect to  $\Omega = (X, P)^m$

$$\begin{aligned} (c(x) - h(x))^2 &= (c(x) - E(h(x)) + E(h(x)) - h(x))^2 \\ &= (c(x) - E(h(x)))^2 + (E(h(x)) - h(x))^2 \\ &\quad + 2(c(x) - E(h(x)))(E(h(x)) - h(x)) \end{aligned}$$

$$E[(c(x) - h(x))]^2 = (c(x) - E(h(x)))^2 + E[(E(h(x)) - h(x))^2]$$

↑ Bias                      ↑ Variance

© Zohar Yakhini and Ariel Shamir IDC

אם ניקח את ה-data-training ונגביל אותו הרכיב שיפסיק להשתנות הוא ה-**bias** כאשר המודל שלנו הוא **perfect fit** לא-**shallow** (a+b $x^2$ )  
שורה שנייה = שורה שלילית = פתרחת סוגרים עבור  $(x, h(x))$  הוא משתנה מקרי  
שורה רביעית = הסוגרים השמאליים הם מספר ובסוגרים הימניים ( $x$ )  
שתייה ב-D, נזכיר שהתוחלת עליו היא מספר. מכיוון שאנו מפעילים תוחלת ומילינאריות  
התוחלת (התוחלת תכנס לתוך הסוגרים ונקבל תוחלת על תוחלת, שזה פשוט  $E(h(x))$ )  
התוחלת של ( $x$  וזה מניב 0, כפול סקלאר) השורה שמתחלת ב-2 תהיה 0.

הוכחנו כי הטעות שאנו ורצים לשערך:

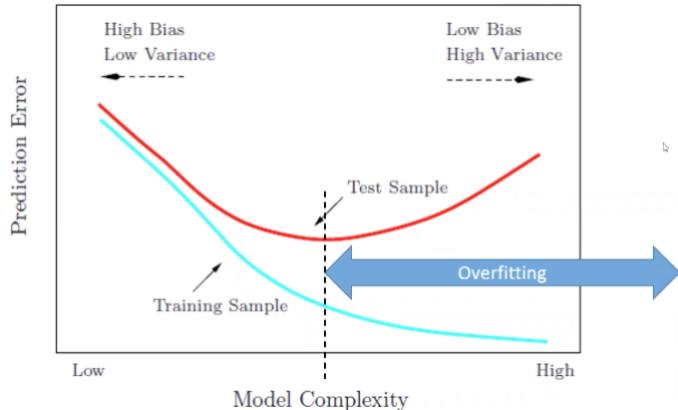
ניתנת לפרק לשני רכיבים ה-**bias** וה-**variance**!

ניתן לראות שה-**bias** תלוי באוסף הhipotheses  $h$  וה-**variance** תלוי בדוגמאות.

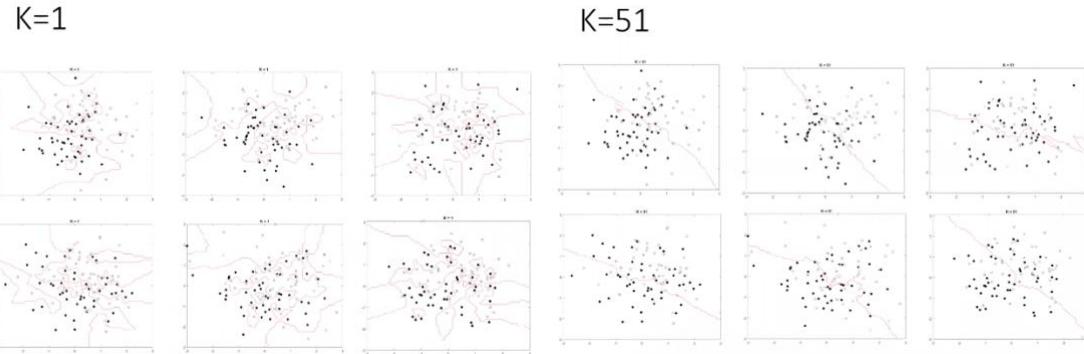
**הבייאס / Bias**: הטעות הבוטה באלגוריתם (ההגבלה על מרחב הhipotheses)

**השונות / Variance**: הטעות הבוטה בDATA (אם ראיינו מספיק דוגמאות DATA?)

**מורכבות המודל**: למשל עבור למידה ליניארית - אם אנחנו במודל פחות מורכב וה-**bais** גבוהה, אם נוסיף עוד דוגמאות DATA וה-**bais** לא ישתנה, נדע שנוכל להעלות את מורכבותו לגובהה יותר – רצחה להקטין את ה-**bais** ונצטרך להזהר מושנות גבולה מידי. פרקטית אנחנו לומדים מתוך ה-**test** את הטעות ומנסים להעריך את התוחלת על כל דוגמה בו שהוא יודעים.



דוגמה של מספר הדגימות (הגובה ביתר) נקבע את המודל הפשוט ביותר (MLC). לכן  $k$  גובה מושפעו training sets נמוך, ה-**bias** שלו יהיה גדול – הוא יטעה יותר וה-**variance** יהיה קטן (ניתן לראות ש"גבול החלטה" עבור  $k=51$  דומה עבור  $k=1$  – מפני שהוא מוקטן מוגנות השגיאות מוגנות עבור training sets שווים).



### PAC Learning – Probably Approximately Correct Learning

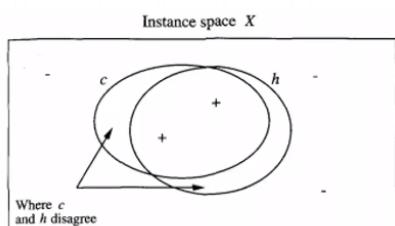
- 1. – מובטחת וודאות עם ביטחון של  $1-\delta$  certainty (Probability .1).
  - 2. – הגבול רצוי, **epsilon**, על הטעות יהיה נקוב. האינטרול.
  - 3. **נשער את השימוש במשאים**: גודל ה-**h**-set (sample complexity) training set – ו-**זמן/מקום** של למידה (הלמידה מתאפשרת בזמן).
- פולינומייאלי** עבור דוגמאות (training)

מרחבי היפותזות קונסיסטנטיים ביחס למרחב הקונסיסטנטיים: מרחב היפותזות  $H$  הוא קונסיסטנטי ביחס למרחב הקונסיסטנטיים  $C$  אם  $C$  מוכל-ב- $H$ . היפותזה  $h$  תקרא  $D$ -קונסיסטנטית ביחס לקונסיסטנטיים  $C$  ול-**D training data** אם לכל נקודה  $d$  ב-**D מתקיים**  $h(d) = c(d)$ .

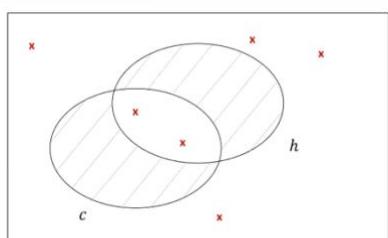
**(טיורתיות אין טעויות על ה-**training!

אלגוריתם למידה קונסיסטנטי: אלגוריתם למידה  $L$ , שפועל על training data שמופקת על ידי קונסיסטנטיים ממוחב הקונסיסטנטיים  $C$ , ומשתמש במרחב היפותזות קונסיסטנטי  $H$  יקרא אלגוריתם למידה קונסיסטנטית אם לכל קונסיסטנטיים  $D$ , ולכל קונסיסטנט  $c$  ממוחב הקונסיסטנטיים  $C$  האוטופוט  $L(D)$  (מניב היפותזה) הוא  $D$ -קונסיסטנטי ביחס ל- $c$ , כלומר אם נסמן  $h = L(D)$  מתקיים  $h(d) = c(d)$  לכל נקודה  $d$ .

**גדדרה יותר כללית**: אלגוריתם למידה  $L$  משתמש במרחב היפותזות  $H$  ופועל על DATA אימון שモפקת ע"י הקונסיסטנט  $C$  יקרא קונסיסטנט, אם קיימת היפותזה שהיא  $D$ -קונסיסטנטית, אז  $L(D)$  הוא אלגוריתם  $D$ -קונסיסטנטי.



**סיבוכיות הדגימה**: מרחב היפותזות סופי / **Finite Hypothesis Space** – אם נניח כי קונסיסטנט המורה מוכל-ב- $H$ , ההיפותזה  $h$  תקרא epsilon-bad אם  $\text{error}(h) > \text{epsilon}$ . אזי אלגוריתם למידה קונסיסטנט חייב להניב היפותזה  $h$  קונסיסטנטית כלשהי, עבור כל  $m$  דוגמאות. השאלה היא מהי הסתברות שזו היפותזה  $h$  תהיה epsilon-bad? מהו P של הדעתה אימון שמוביל ל- $h$  כזו?



החסם על  $P$  (הסתברות ש- $h$  הינה epsilon-bad): נתבונן בהיפותזה  $h$  מרחב היפותזות  $H$ , שהיא epsilon-bad. מחיות כל  $m$  הדוגמאות בלתי תלויות, הסתברות  $(\text{bad} | \Omega) = (X, P)^m$  של היפותזה  $h$  שהיא epsilon-bad להיות קונסיסטנטית (שלא תהיה לה טעות על ה-**training**) עם כל  $m$  הדוגמאות היא קטנה שווה  $m^{-m} (1 - \text{epsilon})^m \leq (1 - \epsilon)^m$ . זה מתקיים עבור היפותזה אחת... המשך בעמוד הבא

If  $h$  is epsilon bad then  $P(\text{err}) \geq \epsilon$   
To be the output of a consistent learning algorithm, all  $m$  training data points had to have avoided the blue region

נתבונן ב- $m$  נקודות DATA,  $D \in X^m$ , ההסתברות שקיים היפוטזה שהיא  $\text{epsilon}-\text{bad}$  ועדיין קונסיסטנטית ביחס ל- $D$ :

הערכת sample complexity במרחב היפוטזות סופי: מה הסיכוי שקיים היפוטזה  $\text{epsilon}-\text{bad}$  שהיא קונסיסטנטית – קטן ממהicho של ההסתברויות שכל היפוטזות שהן  $\text{eps}-\text{bad}$  כנ"ז קונסיסטנטיות עם הדאנו אימון בעל  $m$  הדגימות – קטן ממספר היפוטזות שהן  $\text{eps}-\text{bad}$  כפוף להסתברות שהיפוטזה היא  $\text{eps}-\text{bad}$ . קטן מגודל מרחב היפוטזות כפוף להסתברות ואילו השווון האחרון נובע מטור טילור.

$$\Pr(\exists h \text{ which is } \varepsilon\text{-Bad and consistent}) \leq \sum_{h \in \varepsilon\text{-Bad}} \Pr(h \text{ is consistent with } D_m) \leq |\{h \text{ is } \varepsilon\text{-Bad}\}|(1 - \varepsilon)^m \leq |H|(1 - \varepsilon)^m \leq |H|e^{-\varepsilon m}$$

אם נdag שהביטוי שהתקבל יהיה קטן מדلتא, נהייה "טסודרים".

$$|H|e^{-\varepsilon m} \leq \delta \text{ or } m \geq \frac{1}{\varepsilon} \ln \frac{|H|}{\delta} = \frac{1}{\varepsilon} (\ln |H| + \ln \frac{1}{\delta})$$

- נשים לב שעלייה לינארית במספר הדגימות מקטין את הסיכוי לטעות באופן אקסוננציאלי
- במטרה לצמצם את ההסתברות לכישלון להיות תחת רמת דلتא מסוימת נדרש לדרש:
- זהו איינו חסם הדוק! (בעיקר מכיוון שהחלפנו את המספר של מספר היפוטזות שהן  $\text{eps}-\text{bad}$  בגודל של מרחב היפוטזות  $H$ ).

דוגמה: מרחב האינסטנסים  $X$  הוא וקטוריים בוליאניים  $m$ -ממדים. גם מרחב היפוטזות  $H$  וגם מרחב הקונסיסטנטיים  $C$  מכילים צירופים של  $x$

ליטרליים (או המשתנה או שלו) בוליאניים מהצורה:  $x_1 \vee \bar{x}_5 \vee \bar{x}_{22}$

אין מוגבלה נספת על מרחב היפוטזות: לכל משתנה בוליאני  $x_i$  היפוטזה שלו יכולה להכיל את  $x_i$  או את המשלים שלו  $\neg x_i$  או אף אחד מהם. אף אחד מהם אומר את הקונסיסטנטי False עבור  $x$  (הוא לא עוזר לנו).

- Start:  $x_1 \vee \bar{x}_1 \vee x_2 \vee \bar{x}_2 \vee x_3 \vee \bar{x}_3 \vee x_4 \vee \bar{x}_4$
- Instance 1:  $x_1 \vee x_2 \vee \bar{x}_3 \vee x_4$
- Instance 2:  $x_1 \vee x_4$

Example	$x_1$	$x_2$	$x_3$	$x_4$	$y$
1	0	0	1	0	0
2	0	1	0	0	0
3	0	0	1	1	1
4	1	0	0	1	1

- Consistent with Instances 3 & 4

נראה אם נוכל לבנות עבור מרחב זה לומד קונסיסטנטי. להלן הדאטה אימון:начחיל עם הכל (start), למרות שהוא לא במרחב שלנו, לאחר מכן מכוון גנייע לאינסטנס 1 וניקח את כל הליטרלים שמספריהם לו להלן מה שニアר לאחר התיקון. וכך, נעבור לאינסטנס 2, איקס 2 לא יכול להופיע בצורתו הלא שלילית ולכן נוציא אותו וגם את not-איקס 3. סיימנו בכך מפני שכבר נהיה קונסיסטנטיים עבור אינסטנסים 3 ו-4.

מכאן נוכל להסביר אלגוריתם כללי לחומרה שלנו עבור מציאת אלגוריתם לומד קונסיסטנטי (=מסכים עם הדאטה אימון):

נתחיל עם:  $h = x_1 \vee \bar{x}_1 \vee x_2 \vee \bar{x}_2 \vee \dots \vee x_n \vee \bar{x}_n$  (נשים לב ש- $X = h$  אומר הכיוון כללי) לכל דגימה DATA עם ערך שלילי

(0) בעמודות ה- $x,y,z = \text{false}$  (x) נסיר את כל הליטרלים המקוריים:  $\bar{x}_i$  מההיפוטזה החתולית  $h$ . לכן ישאר לנו:  $h = l_{i_1} \vee l_{i_2} \vee l_{i_3} \vee \dots \vee l_{i_k}$  ממה היפוטזה החתולית  $h$ . כלומר  $l_{i_1}, l_{i_2}, l_{i_3}, \dots, l_{i_k}$  מוכיון של הליטרלים שלהם (גם ליטרל ריק הוא תקין, כפי שנשים לב עבור זה). ומהו קונסיסטנטי גם עם האינסטנסים החשובים (1 בעמודות ה- $x,y$ ) מוכיון של הליטרלים שלהם (גם ליטרל ריק הוא תקין, כפי שנשים לב עבור הדוגמה לעיל שההיפוטזה החשופית אינה מכילה את המסתנה  $X$  ובכל זאת היפוטזה מסכימה עם הדאטה אימון באינסטנס הריבועי) הם בהכרח ב- $\neg h$ . וכן נשים עם  $h$  שהוא גם קונסיסטנטי עם דגימות הדאטה החשובות בהינתן  $m$  אינסטנסים יי'קח לנו  $m^m$  (גודל הממד, מס' הפיצרים) איטרציות ואם נניח שהממד הוא בגודל קבוע הסיבוכיות תהיה  $O(m^m)$ . אז אם נוכיח שגם sample complexity הוא לינארי, אנחנו נהיה עדין באלגוריתם לינארי.

### כמה אינסטנסים נדרשים לנו?

- נניח כי יש לנו 10 אטרוביוטים.
- בדוגמה שלנו, השתמש בחיבור (משפט "ווגם") של פיצרים (או חיבור/גימום ריק עבור כמה מה- $\neg$ -ים) נקבל שגודל מרחב היפוטזות הוא:  $|H| = 3^{10} = 59,049$ .

נרצה להבטיח בודאות של 95% שההיפוטזה שלנו תניב טעות שketuna מ-10%. (הטעות על דגימות שלא ראיינו)

נצריך לכך:  $m > \frac{1}{0.1}(\ln 59049 + \ln \frac{1}{0.05}) = 10(11+3) = 140$  instances

נשים לב שהגודל של מרחב הדגימות  $X$  הוא  $2^{10} = 1024$ .

- נניחCut כי יש לנו 20 אטריבואוטים.
  - נקבל שגורל מרחב ההיפותזות Cut הינו:
  - במקרה זה, כדי לקבל ודאות של 95% שההיפותזה שלנו תنبא טעות קטנה מ-10%, נצטרך:
- $$m > \frac{1}{0.1}(\ln 3.5 \cdot 10^9 + \ln \frac{1}{0.05}) = 10(22+3) = 250 \text{ instances}$$
- ויותר מ-250 דוגימות.
- ובמקרה זה יש לנו בערך 10<sup>8</sup> (בערך מיליון) אינסנסים אפשריים (גודל מרחב הדגימות).

### **PAC Learnable formula של קונספט שהוא**

עבור מרחב קונספטים  $C$ , ומרחב דוגימות  $X$  (כל דוגמה היא ממימוד  $n$ ), ובור אלגוריתם למידה  $L$  על מרחב היפותזות  $H$ , נאמר **הו  $C$ -sh PAC Learnable** אם  $L$  ע"י  $C$  הוא PAC Learnable  $\forall \epsilon, \delta, c \in C$  מתקיים:

1. מתקיים  $\exists h \in H$  כך  $\text{error}_D(h) \leq \epsilon$  (היפותזה  $h$  מתקינה ב- $D$  עם הסתברות שגדולה מ- $(1 - \delta)$ ).
2. אוסף הדוגימות של  $L$  היא פולינומיאלית ב- $n$ :  $\text{poly}(n)$ .
3. מוגן הוכחה שמרחב קונספטים  $C$  הוא PAC Learnable ביעור מרחב היפותזות  $H$  קוניסיטנטי ( $C \subseteq H$ ):

  1. נפער או נפתח אלגוריתם למידה קוניסיטנטי.
  2. נבדוק שהחסם על מספר הדוגימות ( $m$ ) הוא פולינומיאלי ב- $\frac{1}{\epsilon}, \frac{1}{\delta}$ .
  3. נודע שכל צעד באלגוריתם שלנו הוא פולינומיאלי - כלומר האלגוריתם כולם פולינומיאלי ב- $n$ .

- Consider a class  $C$  of possible target concepts defined over a set of instances  $X$  of length (dimension)  $n$ , and a learning algorithm  $L$  using hypothesis space  $H$ .

#### Definition

**C is PAC-learnable by L using H**  
**if**  $\text{error}_{\pi}(h) \leq \epsilon$  for all  $0 < \epsilon < \frac{1}{2}$ ,  $0 < \delta < \frac{1}{2}$ , and for all  $c \in C$  and distributions  $\pi$  over  $X$ , the following holds:

with data drawn independently according to  $\pi$ ,  $L$  will output, with probability at least  $(1 - \delta)$ , a hypothesis  $h \in H$  such that  $\text{error}_{\pi}(h) \leq \epsilon$ ,  
 $L$  operates in time and sample complexity that is polynomial in  $1/\epsilon, 1/\delta, n$ .

© Amit Chawla, Trister Zadeh, Michael Mitzenmacher

### **האם $C = H = \text{Disjunctions of Boolean Literals}$**

- יש לנו אלגוריתם לומד (האלגוריתם שמוצג לעיל).
- סיבוכיות הדוגימה היא פולינומיאלית בכל הפרמטרים:

$$m \geq \frac{1}{\epsilon} \ln \frac{|H|}{\delta} = \frac{1}{\epsilon} (\ln |H| + \ln \frac{1}{\delta}) = \frac{1}{\epsilon} (\ln 3^n + \ln \frac{1}{\delta}) = \frac{n}{\epsilon} \ln 3 + \frac{1}{\epsilon} \ln \frac{1}{\delta}$$

- ונקבל חסם על  $m$  להיות מספיק כדי להבטיח טעות עם הסתברות גודלה מ- $\epsilon$ .  
 3. כל שלב בתהליך הלמידה הוא פולינומיאלי (בודקים דוגימה אחת על ידי התיחסות ב-(n) ליטרליים כדי לשמר על ביטוי קוניסיטנטי) **לכן,  $n$  ליטרליים הוא disjunction!**

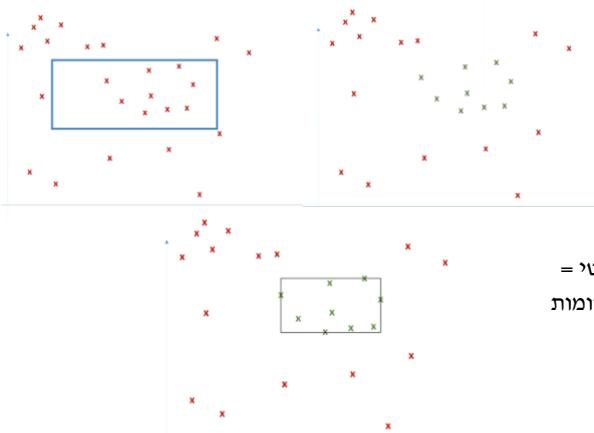
- We have  $n$  Boolean features.
- Each instance in  $X$  is defined by any  $n$  Boolean values. Hence,  $|X| = 2^n$
- A complete hypothesis space contains  $|C| = |H| = 2^{|X|} = 2^{2^n}$  concepts.
- Assume that we have a consistent learner.
- If we now try to apply our simple bound we get:

$$m \geq \frac{1}{\epsilon} (\ln |H| + \ln \frac{1}{\delta}) = 2^n \frac{1}{\epsilon} \ln 2 + \frac{1}{\epsilon} \ln \frac{1}{\delta}$$

**דרך כלל מרחבי היפותזות שלנו לא יהיו סופיים!**

הנה לדוגמה שתווי בברצאה 10: כל המלבנים ב- $\mathbb{R}^2$  שהם מקבילים לצירים יהיה מרחב היפותזות שלנו. להלן קונספט מסוים (המלבן הכהול), כאשר ה-Aים הם ה-training data שלנו. כאשר האירור שלא מכיל את ה-Aים מתאר את ה-training האמתי שבו לא רואים קונספט ספציפי אלא איזושהי קלסיפיקציה (הימני).

נרצה למצוא אלגוריתם למידה קוניסיטנטי: סביר להניח מלבד כך שהצלעות יקחו את האיקס ה- $x$  קטן והאיקס ה- $y$  גדול, וימקמו 2 צלעות מקבילות לציר ה- $x$  (ואת ה- $y$  ה- $y$ ) קטן והאיקס ה- $x$  גדול שימקמו 2 צלעות מקבילות לאורך ציר ה- $x$ . ונטען שהזיהו אלגוריתם קוניסיטנטי = מותאים עם הקונספט על ה-training- $x$  בודדות מכיוון שבתוך המלבן לא יהיו נקודות אדומות מכיוון שמדובר הנחנו שהנקודות הירוקות הן בתוך מלבן.



## קניטריה חישובית מתייען: תרגול 9 – תיאוריה

אין ארוחות חינם – No Free Lunch

הגדירות:

- הדיק הומוכל של היפוטזה ( $Acc_G(L, c)$ ) על קונספט  $c$  = הדיק של  $L$  על דוגמאות שאינן מה-data-training, מבלי להתחשב בדיק המוכל אליו מעניין – בקהלות נוכל להגיע אליו ל-100%.

$$C = \text{קבוצת כל הקונספטים האפשריים על מרחב האיסטנסים, } (x=c) \text{ (קונספט הוא מיפוי מדאות ללייבל)}$$

$$\text{משפט: לכל } L, \text{learner, מתקיים: } \frac{1}{|C|} \sum_{c \in C} Acc_G(L, c) = \frac{1}{2}$$

כלומר: ממוצע הדיק המוכל מעל כל הקונספטים ב- $C$  הוא 0.5 עבור כל התפלגות נתונה  $D$  על מרחב הדוגמאות  $X$  ובגודל  $n$ .

**Proof:** Given any training set  $S$ :

$$\text{For every concept } c \text{ where } Acc_G(L, c) = \frac{1}{2} + \delta,$$

חכמים שודם מאוורי העירון הזה בא להראות לנו שעיל מנת להשכיל למד את המודול שלו הצלח בצורה בעלת ממשמעות, אנחנו חיבים את הנבайл וההיפותזות שלנו איכשהו. כמובן, אם כל ההיפותזות האפשריות הן סבירות זה להיות הקונספט, אז  $\delta$  מועה אך נסימן, הדיק המוצע שלנו בהצלחה על כל הקונספטים היה  $\frac{1}{2}$ . השיבה לכך היא שעובדו של מושגים  $L$ , אם הדיק שלו על קונספט (דוגמאות שלא ראיינו) הוא  $\delta + \frac{1}{2}$ , קיים קונספט אחר (כח שיחיל את הדוגמאות בזורה 'הופכה') שעבורו הדיק של  $L$  היה  $\delta - \frac{1}{2}$ . לכן הדיק המוצע עברו כל מודול  $L$ :

$$\forall x \in S, c'(x) = c(x) = y \quad \forall x \notin S, c'(x) = \neg c(x)$$

$$\frac{1}{|C|} \sum_{c \in C} Acc(L, c) = \frac{1}{2}$$

No Free Lunch Theorem

**הכללת NFL:** לכל שני לומדים  $L_1, L_2$

אם קיים קונספט  $c$  כך שהדיק המוכל של  $L_1$  על  $c$  גדול מהדיק המוכל של  $L_2$  על  $c$ , אז  
 קיים קונספט  $c'$  כך שהדיק המוכל של  $L_2$  על  $c'$  גדול מהדיק המוכל של  $L_1$  על  $c'$ .

For any two learner  $L_1, L_2$

$\exists$  learning problem  $c$  s.t  $Acc_G(L_1, c) > Acc_G(L_2, c)$

$\exists$  learning problem  $c'$  s.t  $Acc_G(L_2, c') > Acc_G(L_1, c')$

L1=	x1	x2	x3	ŷ	L2=	x1	x2	x3	ŷ
	0	0	0	0		0	0	0	0
	0	0	1	0		0	0	1	0
	1	1	0	1		1	1	0	1
	0	1	0	1		0	1	0	1
	1	1	1	0		1	1	1	1
	0	1	1	1		0	1	1	0
	1	0	0	0		1	0	0	1
	1	0	1	1		1	0	1	0

If the concept is  $(0,0,1,1,0,1,0,0)$   
 then  $L_1$  is more accurate with 75% and  $L_2$  has 25%

If the concept is  $(0,0,1,1,1,0,1,1)$   
 then  $L_2$  is more accurate with 75% and  $L_1$  has 25%

דוגמה פשוטה עבור NFL: עם שני קונספטים המשקנה מ-NFL:

לא לצפות שהאלגוריתם הלומד עליינו תמיד יהיה הכי טוב

אלגוריתם פשוט יכול להיות טוב יותר לפעמים (המורכבים יותר יובילו ל-overfit)

ומומלץ לנסה גישות שונות

כאמור, על מנת להיות מסוגלים להצלח בצורה משמעותית, יהיה חיבים למרחב החיפוש שלנו – מרחב היפותזות. על ידי החרות מסוימות.

מעבר לסביר המקורית לכך, יתרכז נסיך הוא שלחיפוש יהיה מהיר ופשוט יותר, אם אנחנו מצמצמים את האפשרויות, חסרנו לכך יהיה העבודה שאם אנחנו מוגבלים את מרחב היפותזות באופן כזה שלא משקף באמצעות את המיציאות, יתכן שלא נמצא היפותזה שיתיהן לנו אפס טוויות על הקונספט.

דוגמא נוספת להצלחה של מרחב היפותזות ראיינו במלל הקורס – חישפנו רק פונקציות לינאריות ברגression, בנו עצי החלטה

שבכל קודקוד ששאלה רק לגבי פיצ'ר בודד (זו הסבה שהמפרדים' שלו בעצי החלטה מתקבלים לציירים, וכו').

## Learning Complexity

### חוורה להערכת הטעות

עד כה כדי לקבל מושג לגבי הטעות האמיתית שלנו, השתרמשנו ב-test set – שאליו התייחסנו כדוגמאות שהמודול שלנו לא ראה, ולכן לא למד, ומשałקפות את המיציאות (הקונספט). אם ה-test set שלנו היה כל יתר הדוגמאות מ- $X$ , אז הטעות שלנו על ה-test set היא בדיק הטעות האמיתית שלנו. כמובן שהוא לא מצב ריאלי, ולכן ככל שנגדיל את גודל ה-test set שלנו, ונקרב להערכת טובה יותר של הטעות האמיתית על הקונספט.

עבור test set בגודל  $|S|$ , ומספר טוויות  $r$ , שיעור הטעות הכללית היא (הדוגמאות בלתי תלויות):

$$p = \frac{r}{|S|}$$

בנוסף, נגדיר את שגיאת התקן להויה:

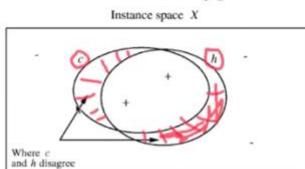
$$se = \sqrt{\frac{p(1-p)}{n}}$$

כתלות ב- $|S|$ , נוכל להתחייב למרווח טוויות מסוימים: "בבטוחן של  $x\%$ , הטעות האמיתית קטנה מ-

Margin של הטעות זהה נקרא רוח סמך – CI :Confidence Interval)

$$CI = p \pm 2(se)$$

## True Error of a Hypothesis



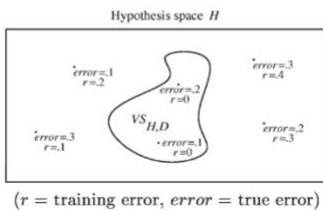
**Definition:** The **true error** (denoted  $\text{error}_D(h)$ ) of hypothesis  $h$  with respect to target concept  $c$  and distribution  $D$  is the probability that  $h$  will misclassify an instance drawn at random according to  $D$ .

$$\text{error}_D(h) \equiv \Pr_{x \in D}[c(x) \neq h(x)]$$

## Version Spaces

### Version Space $VS_{H,D}$ :

Subset of hypotheses in  $H$  consistent with training data  $D$



( $r$  = training error,  $\text{error}$  = true error)

נוצר מדבר על מודלים ואלגוריתמים לומדים, ועבור לדבר על התיאוריה שמאחורי הלמידה. נושא זה נקרא **PAC Learning - Probably Approximately Correct** (probably =  $\text{correct}$ ).

על מנת לשוער את הטיעות האמיתית שלנו, ובונסף להבין כמה דוגמאות דרושות לנו על מנת ללמידה בקרה אינטואיטיבית מספק. ניתן לכך הגדירות מודיקות בהמשך.

כמו שעשינו עד כה, אנו נחים מוגדים ללמידה (דיקוטומיה, פונקציה מסווגת)  $c$ . יש לנו דוגמאות  $X$  (במיוחד הדוגמאות  $X$ ,  $h$ , שטסכים (תהיי קוניסטנטית, עד כמה שאפשר) עם הקונספט שלנו):

$$\text{samples } x \in X \quad \text{concept } c \in C \subseteq P(X) \quad h \in H \subseteq P(X)$$

רכזה למוצאו את ההיפותזה המתאימה ביותר ל- $x$ , מתקן הנהה שהוא מיניג טוב את הדאטא של ראיון, הקונספט). טיעות שאנו מיצרים על האדטא-**out-of-sample error** (אך בפועל לא-**in-sample error**).

באופן אינטואיטיבי, טבעי להגיד שאותו -  $h$  הוא על הדאטא שלא ראיינו, אך בכלל נכון. בפועל יותר על הדוגמאות של תיאורית הלמידה - אנו נcols להגיד דברים על הטיעות הזה, על בסיס הדאטא-**training data**.

נסתכל על דוגמא כדי להבין את הרעיון הכללי. נטיל מבוקע הוגן 100 פעמים. נניח שرك 8 פעמים יצא פאל, והשאר עז. אנו מובן לא זודעים את תוכנת ההתלה בהאה (צורך המוחש, הלהלה הבאה היא הקונספט - דיווח שלא ראיינו). אך בכלל שאנו יודעים שرك 8 מתוך 100 יצאו פאל, יש לנו תחושה שאנו נון יכולים להעריך מה תהיה התוצאה של ההתלה הבאה. אם נעה את מספר הטעטלות ל-1,000, ומוכרים רק 80 יצאו פאל, התחושה הזאת תתחזק - ככל שיעלה מספר הדוגמאות שנראתה ב-, אנו מובן לא יותר על הדוגמאות שלא ראיינו.

רכזה בעת להגדיר את הטיעות האמיתית של ההיפותזה על הקונספט (הדאטא שלא ראיינו). נניח שהקיטות פונקציית התפלגות כלשהי  $D$  על הדוגמאות  $X$  (הדוגמאות יכולות להתפלג נורמלית, לדוגמא, אך אין לא בהכרח חיות במרחב אוקלידי - אף משנה משנה אייזו התפלגות זו, אך הפונקציה מקיימת את התנאים הבסיסיים של פונקציית התפלגות). נגידו את הטיעות להיות **התשובות שההיפותזה לא הסכימה עם הקונספט, בוגר לדגמה רגונומלית** במשהו  $x \in X$ :

$$\text{TrueError}(h) = \text{error}_D(h) = P(c(x) \neq h(x))$$

**משפט:** אם מרכיב היפותזות  $H$  הוא סופי, ו- $D$  היא סדרה של  $m$  (כאשר  $m$  גדול שווה מ-1) דוגמאות רנדומליות בלתי תלויות של קונספט מטרה  $c$ ,

אזי לכל אפסילון בין 0 ל-1, החשובות שקיימת היפותזה ששייכת למרכיב ה- $VS_{H,D}$  עם ( $H, D$  (הגדירה לעיל: תת קבוצה ממרכיב היפותזות

$$|\mathcal{H}|e^{-\varepsilon m} \cdot \text{error}_D(h) > \varepsilon \quad \text{היא קטנה מ-} |\mathcal{H}|$$

כמה דוגמאות יספקו? להלן הוכחת החסם.  
נרצה לחסום את הסיכוי לקבל היפותזה עם שגיאה אמיתית גדולה מאפסילון  $\varepsilon$  מ- $m$  דוגמאות.

This bounds the probability that any consistent learner will output a hypothesis  $h$  with  $\text{error}_D(h) \geq \varepsilon$

We want this probability to be at most  $\delta$

$$\begin{aligned} |\mathcal{H}|e^{-\varepsilon m} &\leq \delta \\ \ln(|\mathcal{H}|e^{-\varepsilon m}) &\leq \ln(\delta) \\ \ln(|\mathcal{H}|) + \ln(e^{-\varepsilon m}) &\leq \ln(\delta) \\ -\varepsilon m &\leq \ln(\delta) - \ln(|\mathcal{H}|) \\ m &\geq \frac{1}{\varepsilon}(\ln(|\mathcal{H}|) - \ln(\delta)) \\ m &\geq \frac{1}{\varepsilon} \left( \ln(|\mathcal{H}|) + \ln\left(\frac{1}{\delta}\right) \right) \end{aligned}$$

$$P(1 \text{ hyp. w/ error} > \varepsilon \text{ consistent w/ 1 ex.}) < 1 - \varepsilon \leq e^{-\varepsilon}$$

$$P(1 \text{ hyp. w/ error} > \varepsilon \text{ consistent w/ m ex.}) < (e^{-\varepsilon})^m = e^{-m\varepsilon}$$

$$P(1 \text{ of } |\mathcal{H}| \text{ hyps. w/ error} > \varepsilon \text{ consistent w/ m ex.}) \leq |\mathcal{H}|e^{-m\varepsilon}$$

\* Because of Union Bound

$$A \cap B \leq A + B$$

דוגמאות לשימוש בחסם זהה: (דוגמה מימין ודוגמה נוספת משמאלי)

- Suppose  $H$  contains conjunctions of constraints on up to  $n=13$  Boolean attributes. Then  $|\mathcal{H}| = 3^{13} = 1594323$

- We want to ensure in 95% that our hypothesis will have error  $< 5\%$

$$m \geq \frac{1}{0.05} \left( \ln(1594323) + \ln\left(\frac{1}{0.05}\right) \right) = 346$$

- 1 attribute with 3 values

- 9 attributes with 2 values

$$|\mathcal{X}| = 3 \times 2^9$$

- $H$  contains conjunctions of attributes, then  $|\mathcal{H}| = 4 \times 3^9 = 78733$

- We want to ensure in 95% that our hypothesis will have error  $< 10\%$

$$m \geq \frac{1}{0.1} \left( \ln(78733) + \ln\left(\frac{1}{0.1}\right) \right) = 143$$

## VC dimension

עד כה דיברנו על מרחב היפותזות סופי,icut נדר על מרחב היפותזות של מפרידים לינאריים במימד דו ממדי. מימד ה- $H$ -dimension (Vapnik-Chervonenekis dimension) הוא ממד של קיבולת (סיבוכיות, expressive power, עושר, או גמישות) של אלגוריתם סיוג סטטיסטי, המוגדר **כקריזינליות** (הגודל) של הקבוצה הגדולה ביותר של נקודות שהאלגוריתם יכול לנפץ (shatter).

- Let  $S(H, X) = \begin{cases} T & H \text{ Shatters } X \\ F & H \text{ Can't shatter } X \end{cases}$
- If  $S(H, X) = F$  this means there is a specific assignment  $y_1, y_2, \dots, y_m$  for which  $\forall h \in H \exists i h(x_i) \neq y_i$

אם  $H$  לא מנפצת את  $X$ , אז קיימת השמה  $y_1, \dots, y_m$  כך שלכל היפותזה  $h$  ממרחב היפותזות, קיים  $i$  בין 1 ל- $m$  מסוים עבורו לא מתקיים:  $h(x_i) \neq y_i$

ברור שאם הגודל של מרחב היפותזות קטן ממספר החסימות, אז מרחב היפותזות לא יוכל לנפץ את קבוצת הנקודות (משום שתיהה השמה שלא תהיה מספקת ע"י המרחב) מרחב היפותזות שמנפה את כל הנקודות  $x$  לערך -1 ו-1 אינו מנפץ את  $X$  מכיוון שקיימות השמות שלא מסכימות אותו (למשל  $(Y_3, Y_4)$ ). מרחב היפותזות שמנפה כל  $x$  באותו אופן שהוא ממנה את  $x$  גם אינה מנפצת את  $X$  מהיות שקיימות השמות שלא מסכימות אותו (למשל  $(Y_3, Y_2)$ )

**גדר נפוץ** : מרחב היפותזות  $H$  מנפץ קבוצת נקודות  $X = \{x_1, x_2, \dots, x_m\}$  (ששייך למרחב הדגימות) אם ורק אם לכל השמה  $\{y_1, y_2, \dots, y_m\}$  כך ש  $y_i = 1$  או  $y_i = -1$  קיימת היפותזה  $h$  (מרחב היפותזות  $H$  שמקיימת לכל  $i$ :  $h(x_i) = y_i$ ).

דוגמה:

- Let  $U$  be some universe and let  $X = \{x_1, x_2\}$ . how many possible assignments  $Y$  does  $X$  have?
 

$Y_1$	$Y_2$	$Y_3$	$Y_4$
$x_1 = -1$	$x_1 = 1$	$x_1 = -1$	$x_1 = 1$
$x_2 = -1$	$x_2 = -1$	$x_2 = 1$	$x_2 = 1$
- Let  $H$  by some hypothesis space.
  - Can  $S(H, X) = True$  if  $|H| < 4$ ? No.
  - Can  $S(H, X) = True$  if  $h(x_2) = -1 \forall h \in H$ ? No.
  - Can  $S(H, X) = True$  if  $h(x_1) = h(x_2) \forall h \in H$ ? No.

הגדולה ביותר של  $X$  שמנפצת ע"י מרחב היפותזות  $H$  – **VC Dimension**

הגדולה ביותר של  $X$  שמנפצת ע"י מרחב היפותזות  $H$ .

- נשים לב כי מספיק למצואת קבוצה אחת בעלת גודל נתון  $sh(H)$  יcollה לנפץ.
- אם קבוצות שרירותיות גדולות סופיות של  $U$  יכולות להתנפץ על ידי  $H$ , אז  $VC(H) = infinity$ .
- זה מגדיר מרחב היפותזות  $H$ .

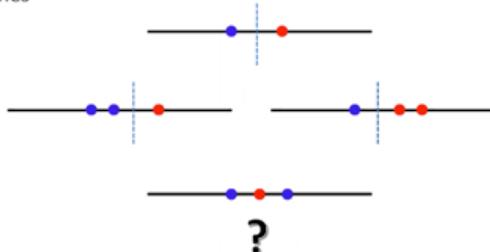
ה-**VC Dimension** של מרחב היפותזות  $H$  על מימד דגימות  $X$ , מוגדר להיות העצמה של תת הקבוצה הגדולה ביותר של  $X$  ש- $H$ -  
יכולה לנפץ.  
מספיק למוגן קבוצה אחת בגודל  $m$  ש- $H$ -יכולה לנפץ, כדי להוכיח ש- $VC(H) \geq m$ .  
צריך להראות ש- $H$  לא יכולה לנפץ אף קבוצה בגודל  $m+1$  על מנת להוכיח ש- $VC(H) < m+1$ .

דוגמה:

גודל מרחב היפותזות של מפרידים לינאריים הוא 2. הראיינו קבוצת נקודות אחת שכל השמה שלא ניתן לנפץ ולכן  $H$  גדול שווה מ-2, ויש להוכיח שכל קבוצה של 3 נקודות לא קיימת השמה שמנפצת ולכן  $H$  קטן ממש מ-3. (בහמשך נראה כיצד מוכחים פורמלית)  $VC(1\text{-dimensional linear separators}) = 2$  ולכן 2

### Shattering – example I

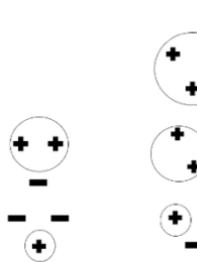
- 1-dimension space
- $H$  – linear lines



נרא הוכחה פורמלית:

יהי  $U$ , מרחב הדוגמאות של כל הנקודות במישור דו-ממדי כלומר  $(y, x)$  ששייך ל- $\mathbb{R}^2$ . מצאו את ה-VC dimension שבו מרחב ההיפותזות הוא כל המוגלים (החלק הפנימי של כל מעגל מסווג כחיובי), והוכחו אותו.

First, we'll show that  $VC \geq 3$ :



$\times \quad \times$

הוכחה:  $VC(H) = 3$

- הצעד הראשון הוא להראות כי  $VC$  גדול שווה 3. לכן נראה קבוצה אחת של 3 נקודות (קבוצה של משולש הפוך שווה שוקיים), ונראה את כל החשומות שמנצחות אותה (אכן מקיימות את הפלסים במעגל).

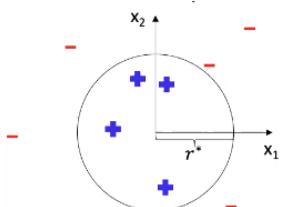
- הצעד הבא הוא להוכיח כי  $VC < 4$  להלן הוכחה פורמלית:

Second, we'll show that  $VC < 4$ :

We show this by constructing a counterexample in several cases

- If the four points are collinear, the labeling +--+ (going along the line) is impossible, among numerous others
- If the convex hull of the four points is a triangle, then the labeling with + (the three points of the triangle) and - (the interior point) is not possible
- If the convex hull of the four points is a quadrilateral, then let  $(a_1, a_2)$  be the points separated by the long diagonal and  $(b_1, b_2)$  be the points separated by the short diagonal. At least one of the labelings  $+(a_1, a_2), -(b_1, b_2)$  or  $+(b_1, b_2), -(a_1, a_2)$  must be impossible:
  - If they were both possible, then there would be some satisfying circle  $c_1$  for the first labeling and some other circle  $c_2$  satisfying the second labeling, and the symmetric difference of these circles  $((c_1 \setminus c_2) \cup (c_2 \setminus c_1))$  would consist of four disjoint regions, which is impossible for circles

Since some set of 3 points is shattered by the class of circles, and no set of 4 points is, the VC dimension of the class of circles is 3



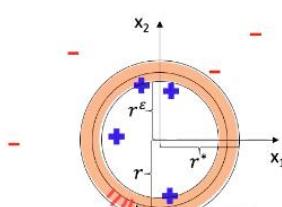
חישוב ישיר של סיבובות הדוגמה (במו דוגמת מרחב המלבנים)

דוגמא: לפניו משחקים למדוד מעגל קונקרטי לא ידוע במישור האוקלידי עם 2 ממדים. יהי  $r$  הרדיוס של מעגל המטרה. כל דוגמה בדתא אימון הוגלה מתוך התפלגות לא ידועה  $D$  ומכליה 2 פיצרים (מיקום הדוגמה על הצירים  $(x_1, x_2)$ ) וערך מטרה (+1 אם היא בתוך המעגל ו-1 אחרת). מרחב הקונספטים שלו הוא עיגולים שמרכזם הוא ראשית הצירים. בניית אלגוריתם שמנצח את המעגל הקטן ביותר שמקhil את הפלסים – עברו כל הפלסים שלו נמצאת המיקום של הקיצוניים ביותר ועל פיהם נגידר רדיוס היפותזה  $r$ .

- $r^*$  – הרדיוס של מעגל הקונספט
- $r$  – היפותזה שמנצח את המעגל
- $r - r$ -epsilon – הטבעת (טבעת הטיעות, הכתומה) הגדולה ביותר שנוצרת עם הסטברות (לייפול בתוכה) שהיא במקסימום אפסילון.

$$r^\epsilon = \operatorname{arginf}_r \Pr[(x_1, x_2) \in A_r] \leq \epsilon$$

- מקרה 1: אם רדיוס אפסילון קטן שווה  $m - r^*$  אז החסתברות לייפול בטבעת קטנה מאפסילון (אוior שלישי) – הטבעת מוכלת בתחום טבעת האפסילון.



Case 2: Otherwise, what is the probability of missing the annulus of radii  $r^\epsilon, r^*$  with  $m$  training examples?

$$(1 - \epsilon)^m \leq \exp(-\epsilon m)$$

With sample size  $m \geq \frac{\ln(\frac{1}{\delta})}{\epsilon}$ , we get

$$\exp(-\epsilon m) \leq \exp\left(-\ln\left(\frac{1}{\delta}\right)\right) = \exp(\ln(\delta)) = \delta$$

מקרה 2:

נניח כי רדיוס

אפסילון גדול מ-  $r$

So if the probability of the annulus is very small, the error it incurs is also small  
With enough examples, it is very unlikely to miss the annulus

### המשך מהרצאה 9 – PAC Learnability

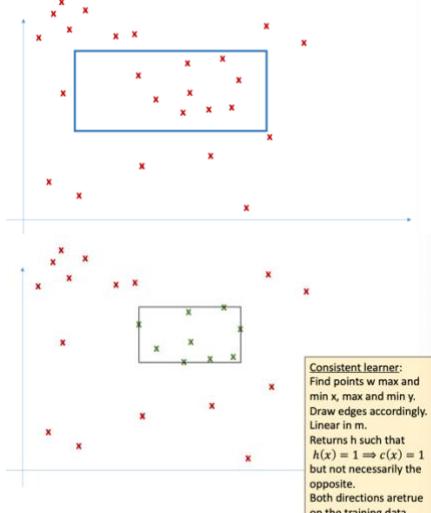
#### Definition

##### C is PAC-learnable by L using H

If for all  $0 < \epsilon < \frac{1}{2}$ ,  $0 < \delta < \frac{1}{2}$ , and for all  $c \in C$  and distributions  $\pi$  over  $X$ , the following holds:

with data drawn independently according to  $\pi$ ,  $L$  will output, with probability at least  $(1-\delta)$ , a hypothesis  $h \in H$  such that  $\text{error}_\pi(h) \leq \epsilon$ ,

$L$  operates in time and sample complexity that is polynomial in  $1/\epsilon$ ,  $1/\delta$ ,  $n$ .



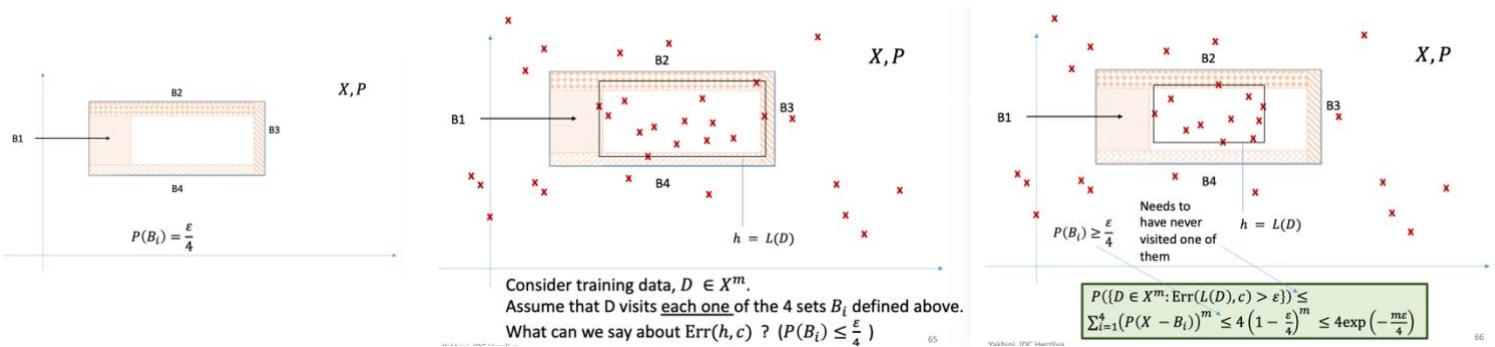
ולכל התפלגות פאי על מרחב האינטגרטים  $X$  ולכל קונספט  $C$  שיש למרחב הקונספטים  $C$  מתקיים: אם דאטה שטוחת באופן בלתי תלוי לפיה התפלגות פאי,  $L$  יהיה אוטופוט, בהסתברות שהיא לפחות  $(1 - \min_{\epsilon} \Delta_{\text{lat}})$ , היפותזה זו ממוחך היפותזות  $H$  כך ש-

בחזורה לדוגמה: אנחנו לא נראה את המלבן הכלול, אנחנו רואים את הקלאסים השונים.Cut נגידר אלגוריתם שנטען שהוא קוסיטטנטי: הוא ייחס את המלבן השחור שמכיל בתוך המלבן הכלול. לכן הטעות של המלבן השחור על ה- $D$  היחסטרות לטעות בין הקונספט להיפותזה: ההסתברות שעבור נקודת מסוימת המלבן הכלול והשחור לא יסכימו. וידוע לנו שהאלגוריתם בנה את המלבן השחור כך שהוא מוכל בתוך המלבן השחור.

נרצה לחסום את הסיכוי להגעה לטעות שגדולה מאפסילון.

#### A bound on sample complexity / סיבוכיות הדגימות

לכל קונספט  $C$  במרחב הקונספטים, נסומן את כל ה- $D$ -datasets שיכולים להוביל ל- $h = L(D)$ .  
שים קיט טעות גדולה מאפסילון  $\epsilon > \text{Err}(h, c)$  למספר סופי של קובוצות (תתי קובוצות של  $X^m$ ). לאחר מכן נשערך את ההסתברות של כל מת-קובוצה של דגימות ולבסוף את האיחוד שלן. מכאן נוכל להסיק כי החסם על סיבוכיות הדגימות כפונקציה של אפסילון ודلتא.



דוגמא: בהינתן המלבן הלבן היפותזה הנוכחית, והמלבן החיצוני הוא הקונספט. נפלח את השיטה שבתוך הקונספט ל-4 פלחים אשר הסיכוי שדעתה "תיפול" שם הוא רבע אפסילון. אם היו נקודות בתוך המלבנים B1-4, אז הטעות תהיה קטנה מepsilone (ולכן זה לא רע, רע, נחשיב כרע אם הטעות שלה גדול מאפסילון). הסיכוי שלא נברך בכל ה-4 נקודות הוא קטן מepsilone, لكن גודיל את היפותזה ש"תגיע" לדגימות אלה (כפי שיתן לראות באיזור האמצעי). ההסתברות של קבצת כל "הרעים" המוכלים באיחוד – קטנה מסכום ההסתברויות של מי שכן נמצא מוחץ לכל אחד מהשתחים  $B_i$ , לכן קיילנו אי-שוויון של טור טיילור (כפי שראינו בהרצאה הקודמת).

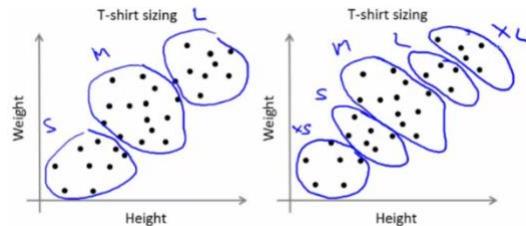
ההסתברות של "הרעים" (בעל טעות יותר מאשר מepsilone) חסומה מלמעלה, אבל היינו רוצחים שהחסם יהיה קטן מepsilone. נוון להבין כי ה- $m$  הינו באופן שיתקיים כי החסם שהתקבל קטן מdalta. ואז הבתו שהתקבל בשוקרי והמנוי יקיים את הדרישות שאנו צריכים. נוון להבין כי ה- $m$  הינו תלוי ב" $4^n$ " זהה שהגדכנו מראש השתחים  $B_i$ . ומכאן נובע שעבור צורה אחרת המלבן ה-4 היה משתנה ועל כל מספר הדגימות ישנה.

## UnSupervised Learning and the K-Means Algorithm

הנחה הבסיסית עד כה הייתה כי עבור ה-training data-labels (ערך של פונקציה או קלאס). הטרת ההנחה מביאה אותנו לשיטה של למידה לא מפוקחת / regularity / structure / learning-unsupervised. איזי במקומם יהיה עליינו ללמידה : מבנה / גורליות / דמיון .clustering algorithms. ולבסוף לא מפוקחת לפעמים משתמש ב-similarity/grouping. וכך גם לקבוצות.

### Clustering

נשתמש בפתרונות שיש בדעתה במטרה לפצל / להפריד את ה-clusters training sets (קבוצות). איברים בכל קבוצה אמרורים להיות "יותר דומים" אחד לשני מאשר אלמנטים בקבוצות אחרות. לכן, המפתח ל-clustering הוא דמיון ואיך מודדים אותו.



**דוגמאות לשימוש ב-Clustering**  
משתמשים בכלאסטוריינג עבור רשות חברות, אינטראקציה של פרוטויאינים, דמיון בין רצויות השמעה.

פילוח שוק – בניית מוצר שמתאים לצרכים של תת-קבוצות באוכלוסייה.

### מדד דמיון / Similarity Measures

ניתן להסתכל על בחיפוש אחר הקיבוץ "הטבעי" dataset. השאלה איך נדע שדוגמאות ב-cluster אחד יותר דומות אחת לשניה מאשר דוגמאות ב-cluster אחר, מערבת שני נושאים עיקריים :

- איזי מודדים דמיון בין דוגמאות? (למשל, מרחק אוקלידי קטן = דומים)
- איזי נוכל להעריך את החלוקה / partitioning ה-cluster set של?

### מטריקה / Distance Metric

=	במרחב שמודדרת עליו מטריקה, יש פונקציה שמקבלת שני אינסטנסים במרחב שמיימת:
Minkowski Metric : $L_k(a,b) = \left( \sum_{i=1}^d  a_i - b_i ^k \right)^{\frac{1}{k}}$ $k \geq 1$	$d(x_1, x_2) \geq 0$
Manhattan Distance : $L_1$	$d(x_1, x_2) = 0 \Leftrightarrow x_1 = x_2$
Euclidean Distance : $L_2 = \left( \sum_{i=1}^d  a_i - b_i ^2 \right)^{\frac{1}{2}}$ $\ a - b\ _2^2$	$d(x_1, x_2) = d(x_2, x_1)$
Infinity Norm : $L_\infty = \max( a_i - b_i )$	$d(x_1, x_2) \leq d(x_1, x_3) + d(x_3, x_2)$

### Algorithmics – constructing clusters

#### האלגוריתם הכל פשוט (אך שימושי) הוא

- היפר-פרמטר : מרחק threshold
- כל עוד יש עדין איברים שעוד לא חולקו לקבוצות בדתא תעבור :

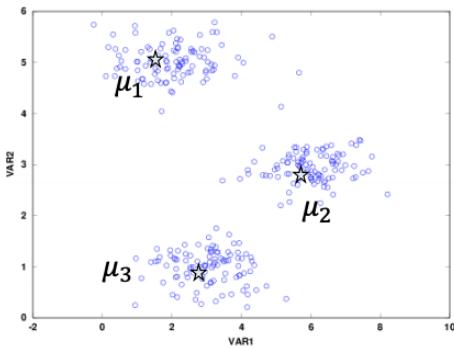
  - תבחר אלמנט s,seed, ותיציר קלאסטר Cs
  - סמן את s כ-clustered (חולק לקבוצות)
  - כל עוד אין אלמנטים s שלא חולקו עם מרחק T < d(e,Cs) תעבור :
  - הכנס את כל האיברים s המקיימים את האישווין לעיל ל-cluster Cs וסמן אותם כ-clustered

### היעיון באלגוריתם זהה :

- החיפוש / מבנה הנתונים – חישוב גרף השכניות יעלה ( $2^n \cdot O$ ), לכן אנחנו צריכים לחשב על שיטות יעילות יותר להוספה כל האלמנטים עם מרחק קטן ממש מ- T. ראיינו כבר שיטות כאלה עברו אלגוריתם KNN.
- תלות בסדר רצומי של הבחירה
- מרחק h-threshold T חייב להיות קבוע – Tים שונים יכולים להוביל לתוצאות שונות.
- איזי משערכים את התוצאה?

## Criterion Function

משימת ה-Clustering : נקבל דאטה של דוגימות { $x_1, \dots, x_m$ } וצרח לחלק אותן ל- $k$  קבוצות זרות :  $C_1, \dots, C_k$ .  
האתגר : לחפש נוסחה ולבצע את משימת הקלסטרינג אוטומטית באמצעות פונקציית קרייטריוון.  
הפונקציה צריכה לזריכת קלטת הדאטא  $D$  ולהגדיר את הקלאסטרים כאלמנטים ומספרם ממשי.



### לאיכות הקלסטרינג יש 2 אספקטים :

- מדידת הקומפקטיות של כל ענן דוגימות, המיצג קלאסטר.
- מדידת כמה רוחקים העננים אחד מהשני (הקלאסטרים).

### מרכז הcovard / Cenroid Base clustering

$$\text{כל קלאסטר מיוצג על ידי הצנטרואיד שלו} = \text{מרכז הcovard שלה}$$

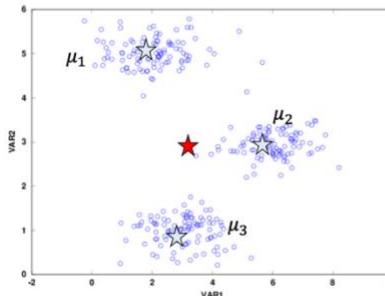
$$G(D, \{C_i\}_{i=1}^k) = \frac{1}{m} \sum_{i=1}^k \sum_{x \in C_i} \|x - \mu_{C_i}\|^2$$

הפונקציה האובייקטיבית שלנו תלויה במרקם אוקלידי הינה:  
הממוצע של מרחקי הנקודות מהמייצג שלו.

זהוי פונקציית המטרה שמודדת כמה ענן הוא קומפקטי בפני עצמו – נרצה להביא אותה למינימום כי ככל שהיא קטנה יותר כך העננים קומפקטיים יותר, והיא פורטת ורק חצי מהבעיה.

$$S_T(D) = \frac{1}{m} \sum_{x \in D} \|x - \mu\|^2 = G(D, \{C_i\}_{i=1}^k) + \sum_{i=1}^k \frac{|C_i|}{m} \cdot \|\mu_{C_i} - \mu\|^2$$

אבל, כאשר מזערים את הקומפקטיות של העננים אנחנו למעשה מגדילים את המרחקים בין העננים.



### Law of Total Variance

ה-Law of Total Variance של total variance של total scatter או scatter clustering מוגדרת כ:  
clear monotone total scatteriani. ניתן להראות כי קיים tradeoff בין הפיזור של שני הורומים: כשהאחד גדול השני קטן. לכן, כאשר מזערים את המרחקים בתוך הקלאסטרים, הדבר מקסם את הפיזור של הקלאסטרים (המרחקים בין העננים).

### The Scatter Criterion / קרייטריוון הפיזור

$$G((x \in D)(\mu_1, \mu_2, \dots, \mu_k)) = \sum_{i=1}^k \sum_{x \in C_i} \|x - \mu_i\|_2^2$$

לכן המינימום של:

$$\sum_{i=1}^k \frac{|C_i|}{m} \cdot \|\mu_{C_i} - \mu\|^2$$

גם יביא למינימום את הסקאטור של מרכז הcovard /-ה-

לכן, יש לנו ייצוג של התוצאות הכלולות של ייצוג ודוגמאות עיי קבוצה של  $k$  קלאסטרים ספציפיים.

הפתרון: ברגע שיש לנו את פונקציית הקרייטריוון, בעיית הקלסטרינג נחפת מוגדרת-היתוב. באופן תיאורטי, שימוש בחיפוש אrox' יכול למצאו אופטימלי. יש בערך  $k/m$  דרכים לחלק  $m$  אלמנטים ל- $k$  קלאסטרים (פתרונות המדויקים נקראים Stirling numbers). וכן, יש אפיו יותר דרכים אם אנחנו מփשים אחר ה- $k$  הטוב ביותר. גישה פרקטית אפשרית לפתרון תהיה: הגדלת ה-region / cluster region או שימוש בשיטות חיפוש יוריסטיות אחרות.

## K-Means Algorithm

אחד אלגוריתמי-clustering היותר פופולריים והשימושים. נניח כי נקבע את מספר הקלאסטרים שבו אנו מעוניינים מראש להיות  $k$  (זהו מודל היפר-פרמטרי). נחפש אחר חלוקת הדאטא ל- $k$  קבוצות (זרות), שהאיחוד שלה היא כל הדאטא, שmbiah לא מינימום את ה-Euclidean norm error criterion (זרות).

$$G((x \in D)(\mu_1, \mu_2, \dots, \mu_k))$$

אלגוריתם זה יכול לעבור על כל מטריקה, אבל אז הדואליות יכולה לא להתקיים.

**אלגוריתם:** אנחנו מתחילה באוף רנדומי  $k$  ערכי מי. ובלולאה אנו מבצעים: נכניס את כל  $n$

הדוגמאות ל- $k$  הקרוב ביותר אליהן, נחש מחיש את  $k$  ערכי מי על פי החלוקת שהתקבעה.

מבצע את הולאה עד איטרציה בה לא יהיה שינוי בערכי המי והציג אותו.

### k-Means Clustering

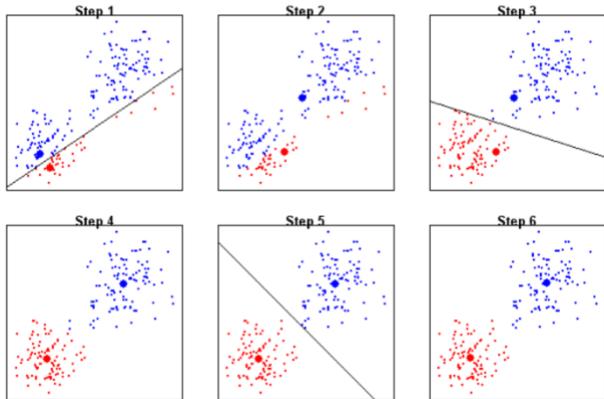
Initialize  $\mu_1, \dots, \mu_k$  (randomly)

Loop:

Assign all  $n$  samples to their nearest  $\mu_i$   
Re-compute  $\mu_1, \dots, \mu_k$  using their cluster members

Until no change in  $\mu_1, \dots, \mu_k$

Return  $\mu_1, \dots, \mu_k$



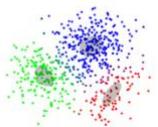
### שתי לולאות פנימיות:

- השמה: נורץ בלולה על כל הדגימות ונכנסים אותם ל"מייצגים" קרובים ביותר אליהם.
- דרך כלל על ידי בדיקת מרחק אוקלידי.
- чисוב מחדש עבור המיצגים: נורץ בלולה על כל הקלאסטרים ונחשב נציגים חדשים.
- בגרך כלל על ידי חישוב מרכז הכביד, הцентрואיד, התוחלת.

### התכונות מובטחות

בכל לולאה פנימית הפונקציה  $\frac{1}{m} \sum_{i=1}^m \|x_i - \mu_{c_i}\|^2$  היא (באופן חלש מאוד) ממווצעת. בשלב ההשמה: אם דגימה קרובה יותר לנציג של קלאסטר אחר, אז היא מקבלת השמה חדשה והפונקציה מומשורת מחדש של הנציגים: הцентрואיד של תת-קבוצה ממזער את הממוצע של המרחקים בין כל קבוצה. יש מספר סופי של השמות אפשרויות (אםنم סופי וдол מודרך אך עדין סופי), אז הפונקציה חסומה מלמטה (על ידי המינימום של ההשומות האפשריות). מכאן שהיא באחרה מותכנות, אבל לא בהכרח למינימום גלובלי, אלא מקומי.

### Fuzzy Clusters

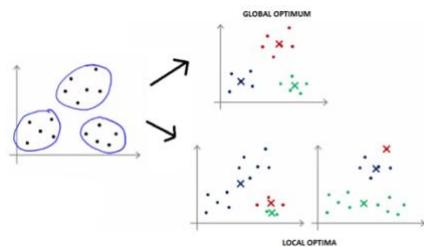


### הרחבות ייריסטיות להימנע מהתכונות לא רצויות:

- הרצת  $k$ -means  $m$  פעמים עם איתוחלים שונים ובחירה המינימום הлокלי הטוב ביותר.
- כאשר הנציג איננו מייצג אף דגימה:

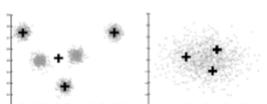
  - יותר עליו (ונחפוך לאלגוריתם  $(k-1)$  means clustering)
  - נבחר נציג חדש במקומו
  - נבחר את הקלאסטר בעל השגיאה הגדולה ביותר ונפצל אותו ל-2

### דוגמא להתכונות שלא לאופטימום:



### איך נדע באיזה $k$ לבחור?

#### Wrong $k$ ...



$$G = \frac{1}{m} \sum_{i=1}^m \|x_i - \mu_{c(x_i)}\|^2$$

באופן כללי, ככל שה- $k$  גדול יותר, כך ערך המינימום של הפונקציה קטן.

( $G=0$  נקבע מכך).

- אין תשובה שלמה/מדויקת תיאורטית לשאלת זו.

בפעמים רבות הבחירה תלואה בדעתה / במטרת העסוק (למשל כמו סוגים שונים אמרורים להיות למוצר?).

בפרקטיקה – נשתמש בקריטריון ספציפי כדי להניע לכיוונו את הבחירה של  $k$ )

- 

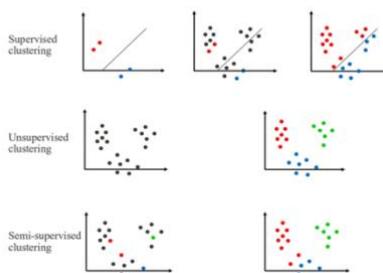
- 

- 

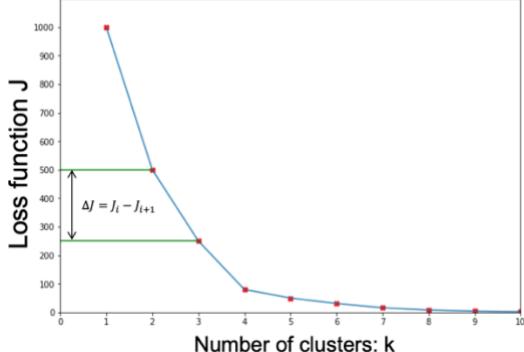
- 

- 

### Semi-supervised clustering



חיפש אחר "מפרק / elbow" בבירזומים של הגראף – מקום בו הגדייה של  $k$  איננה משפרת כל כך את הביצועים.



## למידה חישובית ניפוי | תרגול 10 – UnSupervised Learning

הדבר המרכזי ב-**unsupervised learning** הוא שאין לנו **label**, כלומר עלינו למדוד על הדadata: מבנה, דמיון. לכן, נרצה להסיק מהו המבנה של הדadata ה-unlabeled.

### Clustering

חולוקת הדגימות ה-unlabeled לחתוי קבוצות של קלאסטרים. דפוסים בתוך קלאסטר יהיו מאוד דומים, ודפוסים מחוץ קלאסטרים שונים יהיו מאוד שונים. אלגוריתם הקלאסטרינג ימצא קלאסטרים, תתי קבוצות של הדadata, גם אם לכטורה אין באמות קלאסטרים בדadata. ומטרתו למדוד את המבנה השוכן בחובו של מרחב הפיכרים של הדadata. לכן אין שיטה "טובה ביותר" למצוא את הקלאסטרינג הטוב ביותר, אבל ככל להציג שבחינות הנחות התכנסנו לתוצאות הטובה ביותר.

יש להגיד באלגוריתם קלאסטרינג איך מודדים דמיון, מהי פונקציית המטרה וכו'...

### מדדית דמיון

- עלינו לדעת איך למצוא דמיון, וכל למשל נוכל לדבר על מרחק – ככל שהדגימות קרובות יותר כך הוו יותר דומות.
- מרחק חייב לקיים את התכונות הבאות

- Non-negativity:  $d(x_1, x_2) \geq 0$
- Identity of indiscernible:  $d(x_1, x_2) = 0 \Leftrightarrow x_1 = x_2$
- Symmetry:  $d(x_1, x_2) = d(x_2, x_1)$
- Triangle inequality:  $d(x_1, x_2) \leq d(x_1, x_3) + d(x_3, x_2)$

ראינו סוגים מרחקים שונים: מנהטן, אוקלידי וכו'.

**הגישה הפשוטה:** נתחילה בדוגמה שמשומנת clustered ונכניס אותו לתוך קלאסטר זה. נכניס לקלאסטר זה את כל הדגימות בעלות מרחק (מדד הדמיון) שטמך מייחסו לאות אחד מהאינטנסיסים בклאסטר – נחזיר על כך עד שלא יהיה דגימות שנוכל להסיף לקלאסטר. שאין להן cluster. נחזיר על שני העדינים עד שלא יהיו דגימות לא מסומנות.

### איך נדע אם החלוקה לדגימות היא הטובה ביותר עבור k קלאסטרים כלשהם?

- נמקסם את המרווח (המרחקים) בין קלאסטרים
- נמיציר את המרחקים בין הדגימות בתוך קלאסטר ספציאלי

### K-Means

### K-Means

- Initialize randomly the k-means  $\mu_1, \dots, \mu_k$
  - Repeat
    - For each instance
      - Assign it to the nearest cluster w.r.t its mean  $\mu_i$
      - Re-computes  $\mu_i$  for each cluster
  - Until no change in  $\mu_1, \dots, \mu_k$  (or any other stopping condition)
  - Return  $\mu_1, \dots, \mu_k$
- \* Usually uses simple Euclidean distance in feature space

Centroid-based clustering – קלאסטרים מיוצגים על ידי וקטור מרכז, שהוא לא בהכרח חלק מהדadata טט.

בהתיקון קובצת דגימות ( $x_1, \dots, x_n$ ) כאשר  $x$  הוא וקטור  $d$  ממדי.

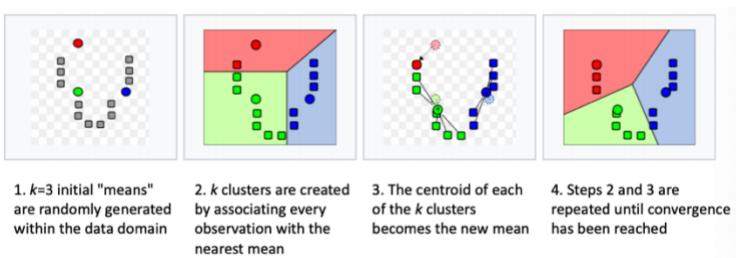
אלגוריתמים פועלים במטרה לחלק את n הדגימות ל-k (קטן שווה מ- $n$ ) קלאסטרים  $\{C_1, \dots, C_k\}$  כך שסכום מרחקי הריבועים

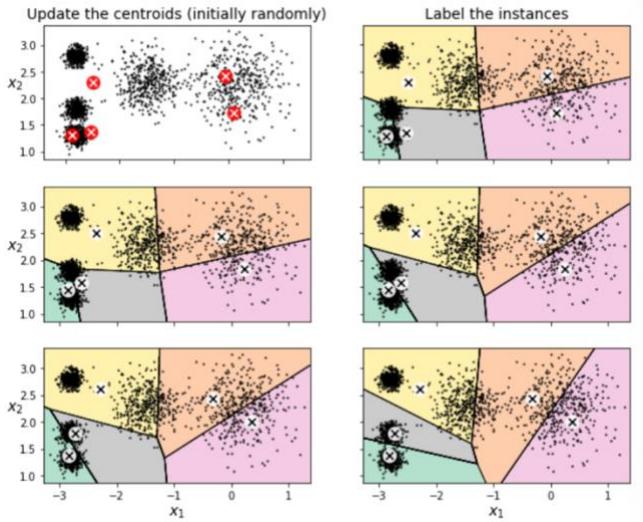
$$\sum_{i=1}^k \sum_{x \in C_i} \|x - C_i\|^2$$

הפתרון הטוב ביותר: יש  $k$  דרכים לחלק את n האיברים ל-k

$$\sim \sum_{k=1}^n \frac{k^n}{k!}$$

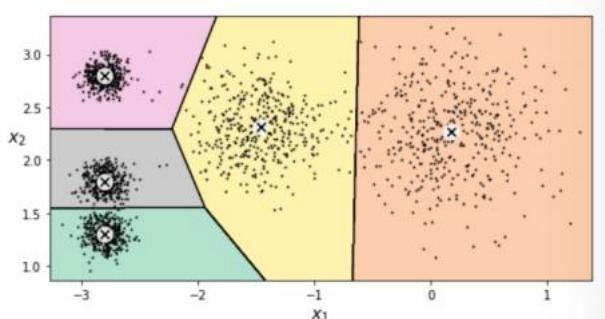
קבוצות ואם נחפש גם אחר k טוב יותר נקבל בערך: לכן מציאת פתרון אופטימלי לביעית אופטימיזציה זו היא מורכבת גם עבור  $k=2$  שכן בדרך כלל משתמשים בגישות / אלגוריתמים יורייסטיים.





דוגמה לIMPLEMENTATION K-Means Clustering על דיאגרמת Voronoi

דוגמאות קוד מופיעות בתרגול

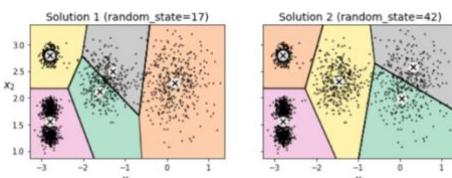


$$\text{להלן הפונקציה שורצחה למוצר:} \quad \underset{i=1}{\overset{k}{\sum}} \sum_{x \in D_i} \|x - \mu_i\|^2$$

ונוכל לכתוב אותה גם באופן הבא (כאשר מיו-סי-איי היא התוחלת של הקלאסטר ש-א-ז שייך אליו):

$$\text{בכל שלב הפונקציה} \quad \frac{1}{m} \sum_{i=1}^m \|x_i - \mu_{c_i}\|^2 \quad \text{מוגדרת באופן הבא:}$$

- בשלב ההשמה של דגימה או קלאסטר: אם א-ז לא משנה את הקלאסטר – אין שינוי. אם א-ז משנה את הקלאסטר שאליו הוא שייך, הוא משוויך קלאסטר הקרוב ביותר – נפחית.
- בשלב החישוב מחדש של התוחלת מיו : הממוצע יניב error square min – ולכן נפחית.
- ככל תהיה התכנסות, אבל לא בטוח לאופטימום.

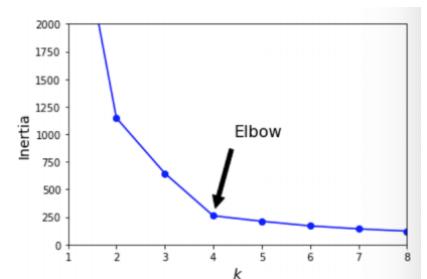


דוגמה להתקנסות למינימום מקומי : נבעצ' שני איטחולים במקום שונה ונקבל תוצאות שונות, ככלומר התקבלה התקנסות למינימום מקומי – ונרצה להימנע מ מצב זה. لكن גדייר אינרגטיה.

**אינרגטיה / inertia :** כדי לבחור את המודל הטוב ביותר עליו לשערק את הביצועים של ה-k-means. לצערנו, קלאסטרינג היא מושימה לא מפוקחת או אין לנו ערכי target. אבל נוכל לפחות למדוד את המרחקים בין כל דגימה לצנטרOID שלה. אינרגטיה היא סוג של מטריקה והיא מוחשבת את סכום המרחקים המרוביים בין כל training instance לבין הצנטרOID הקרוב ביותר אליה.

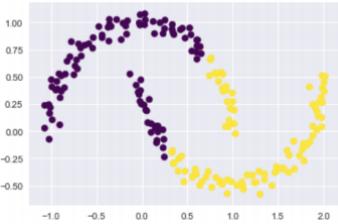
**איטחולים רבים / multiple initializations :** גישה לפrown בעיית ההתקנסות למינימום מקומי הינה להריץ את אלגוריתם k-means מספר פעמים עם איטחולים רנדומליים שונים בכל ריצה, ו לבחור את הפתרון שמזער את האינרגטיה. (בשוקפית 35 מקרים בקוד)

**מציאת המספר האופטימלי של קלאסטרים :** לא תמיד נוכל לקחת את הערך  $k$  אשר מזעיר את האינרגטיה מכיוון שהוא הולך וקטן ככל שנגדיל את  $k$ . ככל שיש יותר קלאסטרים, כך כל דגימה תהיה יותר קרובה לצנטרOID הקרוב ביותר אליה, ולכן האינרגטיה תלך ותקטן. אבל, אבל, נוכל להציג את האינרגטיה כפונקציה של  $k$  ולנתח את הגראף – כך שנשאף לבחור את ה- $k$ -בו מופיע "מרפק", בו השינוי באינרגטיה מפסיק להיות משמעותי. בדוגמה הנ'ל יש מרפק עבור  $k=4$ , המשמעות היא שפחות קלאסטרים מ-4 יהיו לא טובים, ויותר מ-4 לא יעזור יותר מידי ועשויים לחצות קלאסטרים ל-2.



## Hard Clustering vs. Soft Clustering

- Hard** – נניח כי כל דגימה ניתנת השמה "קשה" (כלומר חד משמעות), לקלאסטר אחד בלבד. כל דגימה יודעת לבדוק לאן היא שייכת. גישת Hard היא זו שהשתמשנו בה עד עכשו.
- Soft** – נוון הסתברויות שדגימה תהיה שייכות לכל אחד מהקלאסטרים (למשל עבור דגימה זו וüber 3 קלאסטרים, יהיה לה 70% להשתיך לקלאסטר A, 25% להשתיך לקלאסטר B ו- 5% להשתיך לקלאסטר C).
- איך אפשר להשתמש ב-Soft k-means?** עם מרחקים. למשל, לכל דגימה נחשב וקטור מרוחקים מכל התוחלות, נרמל את הווקטור ונשתמש במוצע המשקלות כדי לחשב את התוחלות החדשות.

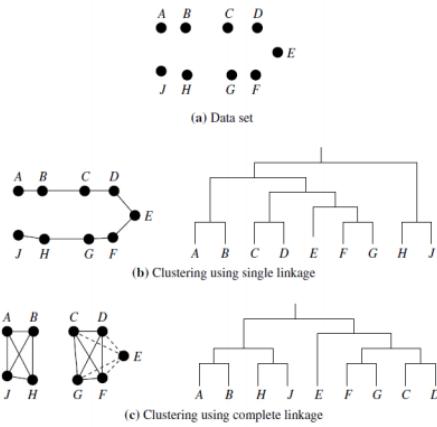


**בעיה נוספת שיכולה לעלוות הינה:** ההנחה כי הנקודות יהיו קרובות ביותר למרכו הכלוב שלן מאשר לתוחלות אחרות בין clusters, היא הנחה לאינארית. לכן, שיטת k-means אינה מתאימה עבור גילוי clusters עם צורות שאינן קווארות או עם clusters בעלי גודלים שונים. בנוסף לכך, האלגוריתם רגיש לרעש ו-outlier data points מכיוון שמספר קטן של data כזו יכול להשפיע על ערך התוחלת.

### נפטרו זאת על ידי קלאסטרינג היררכי / Hierarchical Clustering

שתי הגישות העיקריות לקלאסטרינג היררכי הן divisive ו-agglomerative.

- ב- agglomerative clustering** אנחנו מתחילה עם כל דגימה כקלאסטר ומוגדים את הזוג הקרוב ביותר של קלאסטרים עד שניגע לקלאסטר אחד. זהו תהליכי איטרטיבי שמסתכם בצדדים הבאים: נחשב את מטריצת המרחוקים על כל הדגימות, ניצג כל נקודת דאטה כקלאסטר שהוא סינגלטון, נמוג את שני הקלאסטרים הטובים ביותר בהתאם linkage criterion על ה- linkage criterion, נעדכן את מטריצת הדמיון/המרחוקים, נחרור על צעדים 2-4 עד שניגע לקלאסטר אחד.
- ב- divisive hierarchical clustering** אנחנו מתחילה עם קלאסטר יחיד שמכיל את כל הדגימות שלו ובאופן איטרטיבי מפרצים אותו לקלאסטרים קטנים יותר עד שנישאר עם קלאסטר לכל דגימה.



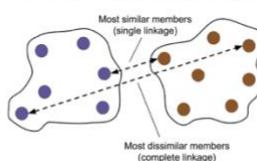
### Linkage Measure

הlinkage criterion קובע באיזה מർחק להשתמש מבין קבוצה של דגימות.

האלגוריתם ימוך זוגות של קלאסטרים שմזער את הקритריון הזה.

מדידות linkage נפוצות:

- Single linkage uses the minimum of the distances between all observations of the two sets
- Complete linkage uses the maximum distances between all observations of the two sets
- Ward minimizes the variance of the clusters being merged
- Average uses the average of the distances of each observation of the two sets

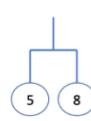


האוטופוט הרומי של קלאסטרינג היררכי הינו **Dendogram** / שומרה את מערכת היחסים והיררכיאלית בין קלאסטרים.

דוגמא:

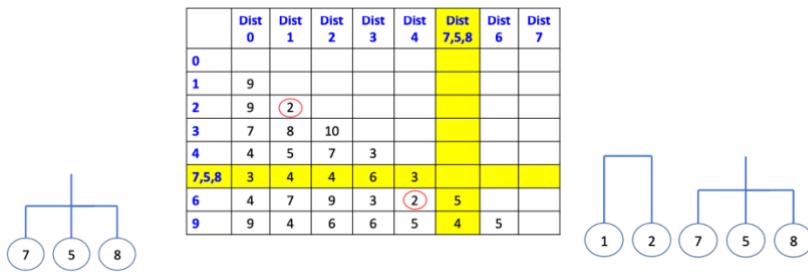
נתונה לנו הדרטה הבאה:  
 $\{(1,2), (4,8), (3,9), (7,3), (4,3), (2,4), (5,2), (3,5), (2,5), (6,6)\}$   
 ראי את האלגוריתם agglomerative hierarchical clustering.

Point	X1	X2	Point	Dist 0	Dist 1	Dist 2	Dist 3	Dist 4	Dist 5	Dist 6	Dist 7	Dist 8	Dist 9
0	1	2	0										
1	4	8	1	9									
2	3	9	2	9	2								
3	7	3	3	7	8	10							
4	4	3	4	4	5	7	3						
5	2	4	5	3	6	6	6	3					
6	5	2	6	4	7	9	3	2	5				
7	3	5	7	5	4	4	6	3	2	5			
8	2	5	8	4	5	5	7	4	1	6	1		
9	6	6	9	4	6	4	5	6	5	4	5		

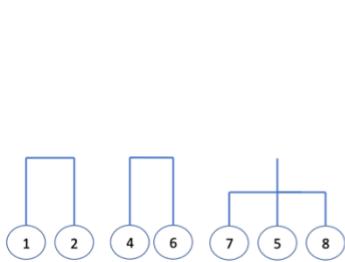


	Dist 0	Dist 1	Dist 2	Dist 3	Dist 4	Dist 5,8	Dist 6	Dist 7	Dist 9
0									
1	9								
2	9	2							
3	7	8	10						
4	4	5	7	3					
5,8	3	5	5	6	3				
6	4	7	9	3	2	5			
7	5	4	4	6	3	1	5		
9	9	4	6	5	5	5	5	4	

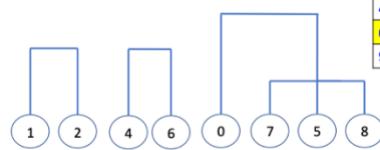
המ�ך  
הצומגמה:



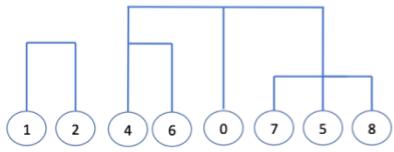
	0	1,2	3	4	7,5,8	6	9
<b>0</b>							
<b>1,2</b>	9						
<b>3</b>	7	8					
<b>4</b>	4	5	3				
<b>7,5,8</b>	3	4	6	3			
<b>6</b>	4	7	3	(2)	5		
<b>9</b>	9	4	6	5	4	5	



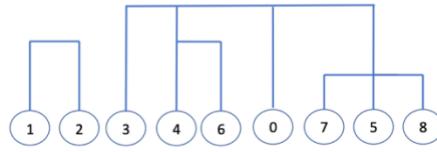
	0	1,2	3	4,6	7,5,8	9
<b>0</b>						
<b>1,2</b>	9					
<b>3</b>	7	8				
<b>4,6</b>	4	5	(3)			
<b>7,5,8</b>	(3)	4	6	(3)		
<b>9</b>	9	4	6	5	4	



	1,2	3	4,6	0,7,5,8	9
<b>1,2</b>					
<b>3</b>	8				
<b>4,6</b>	5		(3)		
<b>0,7,5,8</b>	4	6	(3)		
<b>9</b>	4	6	5	4	

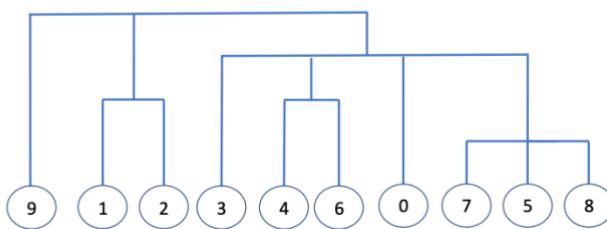


	1,2	3	4,6,0,7,5,8	9
<b>1,2</b>				
<b>3</b>	8			
<b>4,6,0,7,5,8</b>	4	(3)		
<b>9</b>	4	6	4	



	1,2	3,4,6,0,7,5,8	9
<b>1,2</b>			
<b>3,4,6,0,7,5,8</b>	(4)		
<b>9</b>	(4)	(4)	

The final dendrogram is:



## הערכת ביצועים – 11/confusion matrix

		Predicted class	
		positive	negative
True class	positive (#P)	#TP	#P - #TP (FN)
	negative (#N)	#FP	#N - #FP (TN)

מסוגים = מסוג מותאים אובייקט לסת מוגדר מראש של קטגוריות או קלאסים, בהתבסס על סט של ערכי פיצירים שצפינו בהם. בהרצאה זו נתמקד בשיעור הביצועים של מסווגים שפותרים את המקרה של שני קלאסים: "חיווי" / "שלילי".  
במצב של שתי מחלקות יש מטריצה confusion (confusion matrix). נניח כי יש לנו 100 דגימות כאשר הערך האמייתי הוא 70 הם חיובים ו-30 הם שליליים. אם בקלט החזוי חיזנו שרק 50 הם חיובים אז נסמן 50 שהם #TP ו-20 שהם #FP (true positive). המונח הדעתה האמייתי הם חיובים לא כוון. **לסיום, האלכסון הראשי הוא חיוי נכוון והאלכסון המשני הוא חיוי שגוי.**

4 המספרים במטריצה זו  $2 \times 2$  אמורים להישם ל-100, כמספר הדגימות.

- Reduce the 4 numbers to two rates  
 $\text{true positive rate} = \text{TPR} = (\#TP)/(\#P)$   
 $\text{false positive rate} = \text{FPR} = (\#FP)/(\#N)$
- These rates are independent of class ratio
- We can/should compute confidence intervals for both

•  $\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$  – כל אלו שהם חיובים באמת, TP הם כל אלו שחזינו שהם

– **True Positivity Rate**: Real Positive Rate. נקרא גם:

**נרצה שזה יהיה מספר גדול.**

– **Specificity** – הסיכוי לחזות שליליים באופן נכון.

– **False Positive Rate = 1 – specificity**

Error = 1 - accuracy

### מדדיה הביאוניים בסקלארים:

$$\bullet \text{Accuracy} = \frac{\text{Correctly Classified}}{\text{All Instances}} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}$$

$$\bullet \text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\bullet \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (\text{aka Sensitivity})$$

$$\bullet \text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \quad \text{FPR} = 1 - \text{Specificity}$$

$$\boxed{\text{We often study } \text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \text{ and } \text{FPR} = \frac{\text{FP}}{\text{TN} + \text{FP}}}$$

### דוגמה: 3 מסווגים

taboo על שלושה מסווגים שונים שמשווים 200 דגימות. נשים לב שה-

data test או מה הדבר, אבל האלגוריתמים יכולים להיות שונים /

פרמטרים שלהם שונים / האתחול שלהם שונה / סט פיצירים שונה.

את הירוק למשל בוטוח לא ניקח מפני שמול האדום יש לו יותר נזק

וגם FPR יותר גבוהה.

נשים לב שתוצאות השחן FN – הן תוצאות שימושן שחזינו שלילי אבל

האמת הוא חיובי, נגיד מdad ביצועים (נקרא לפעמים expected benefit)

(האמת הוא חיובי, נגיד מdad ביצועים (נקרא לפעמים expected benefit)

$$\pi = \alpha * \text{TPR} - \text{FPR}$$

שתופס את האיזון הזה. השם המעודך יהיה תלוי באalfa (אי שלילי), תוחם המשקלים שניטרניים

לשני סוג הטיעויות. עברו אלף גודלה מ-1 נעניק יותר חשיבות ל-TPR-FPR, אבל

(למשל במקרה של סרטן, שרצה למטען תוצאות של FP), ובעור אלף

FPR. נעניק יותר חשיבות ל-FPR.

בעור אלפיות שונות נקבל עדיפות שונה כפי שניתן לראות <>

(השיפוע הוא אחד חלקי אלף)

True	Predicted	
	pos	neg
pos	40	60
neg	30	70

True	Predicted	
	pos	neg
pos	70	30
neg	50	50

True	Predicted	
	pos	neg
pos	60	40
neg	20	80

**Classifier 1**

TPR = 0.4

FPR = 0.3

**Classifier 2**

TPR = 0.7

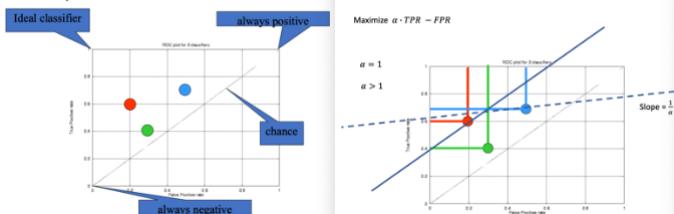
FPR = 0.5

**Classifier 3**

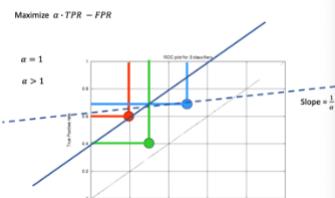
TPR = 0.6

FPR = 0.2

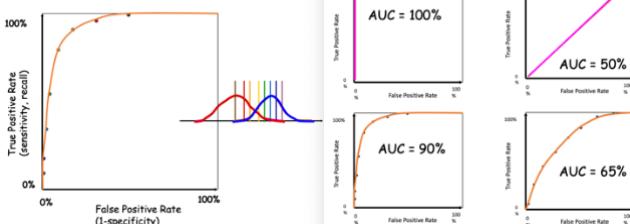
ROC plot for the 3 Classifiers



Cost considerations



ROC curve



### ציירת ROC Curve

מסוג מניב נקודת ROC יחידה.

אם למסוג יש פרמטר "רגישות", השינוי שלו ייתנו לנו סדרה של נקודות ROC (confusion matrices).

לחילופין, אם המסוג מונב על ידי אלגוריתם למידה, סדרה של נקודות ROC (test data) יכול להיווצר על ידי שינוי התוכנות של תחילה האימון כמו ההיפר פרמטרים של האלגוריתם.

כל שחשתח מותחת ל-curve (AUC = area under curve) curve יouter גודל כך נהיה יותר מרווחים מהבדיקה / הפיציר!

•

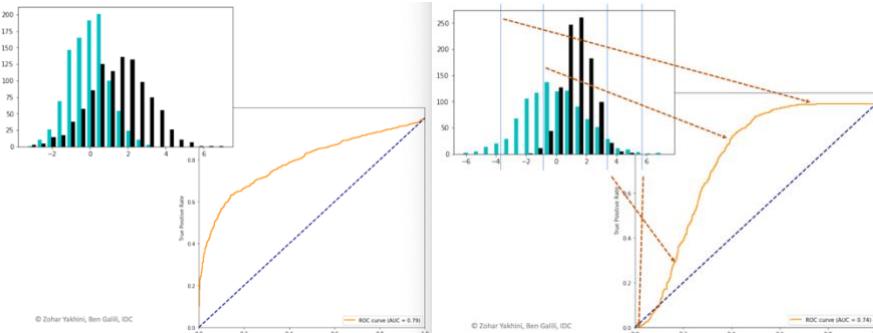
•

•

•

•

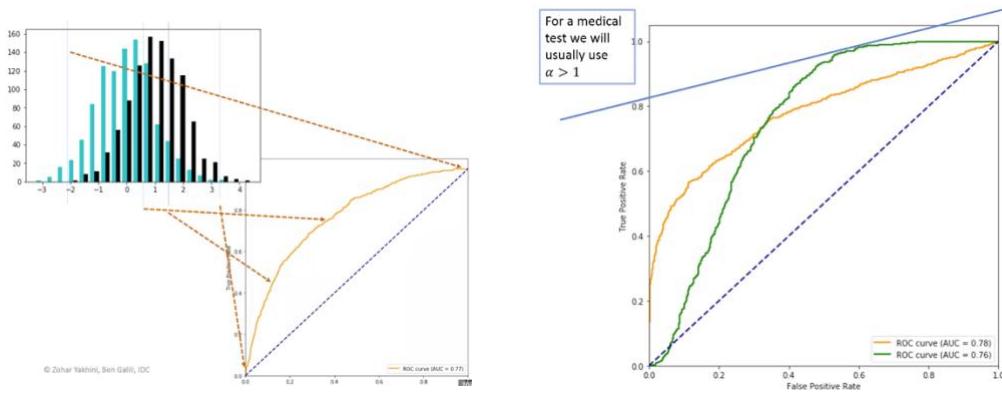
•



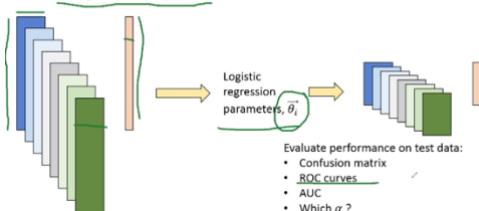
**דוגמה :**  
יצרנו שני גausיאנים כשותוחת של אחד מהם  
שונוות. נשנה את התוחלות ונראה כי ה-AUC  
משתנה ולעתים נעשה יותר טוב. אף יותר גדול  
מי-1 משמעותי יותר עברו FN-h-NFN (רכזה "להעניש"  
יותר FN בבדיקות לסרטן למשל)  
**תליי באלפא איזה AUC מבין שניים יותר טוב:**

פוזיטיב = השחור  
נגטיב = הכחול

נשים לב שבוגמה זו, כל מיניג קלסיפיקציה וכך גם צהיר ROC Curve. כך למשל בעמודות השמאלות של הגרפ הימני (החולות)  
הבודדות = משמעוון כל מי שבਮאל חולה וכל מי שבמיון (ברא), נקבל TPR של 100%, קלומר 1 כי הוא מפלג את הדאטנה נכון.

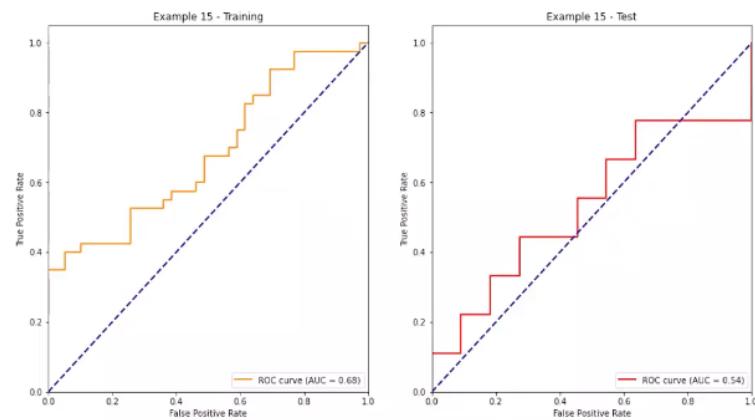
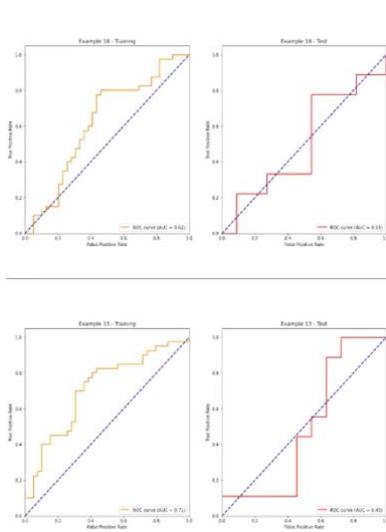


Data: different sets of vectors of 4 features (results of a certain measurement technology), 99 instances w labels representing AMI status: H (negative) and MI (positive)



**דוגמה : בחרית פיצ'רים ו- thresholds עבור קלסיפיקציה LoR**  
הדאטה שלנו היא קבוצות שונות של וקטורים בעלי 4 פיצ'רים. כל סט של 4 פיצ'רים נבנה מטריצה.  
הוקטור ההורוד מייצג את הליבלים, מרכיבים וגרסיה לוגיסטי, שלומודת וקטור טטה-5-ממדי.  
ניגש ל-ROC Curve של הלמידה שלנו.  
**נרצה AUC הכי טוב ב-test!**

**נשים לב כי בגרפים כאן, הגרפ הכתום מותאר את עקומה ה- training Curve של ה-test – שלא מעניין אותנו, לעומת הגרפ האדום שמתאר את ה-test.**



## כמירות חישובית מתייחסת תרגול 11

### שאלות חוזרת למבון

Give an argument for preferring pinv

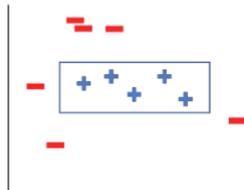
- Guaranteed to find global minimum

Give an argument for preferring GD.

- Computationally in-feasible to solve analytically for very large datasets

Let  $U = \mathbb{R}^2$  and  $H$  be the set of axis aligned rectangles, s.t. all points inside the rectangle are label +.

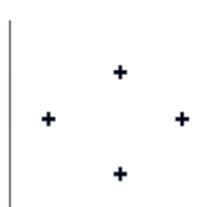
What is the VC Dimension of  $H$ ?



First let's show that  $VC(H) \geq 4$

Easy to see that we have an  $h \in H$

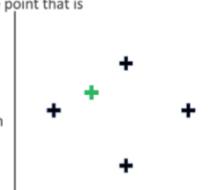
For each labeling,



Now, let's show that  $VC(H) > 5$

For any set of 5 points there must be some point that is "internal", i.e., is neither the extreme left, right, top or bottom point of the five.

If we label this internal point as negative and the remaining 4 points as positive then there is no axes-aligned rectangle which could realize this labeling

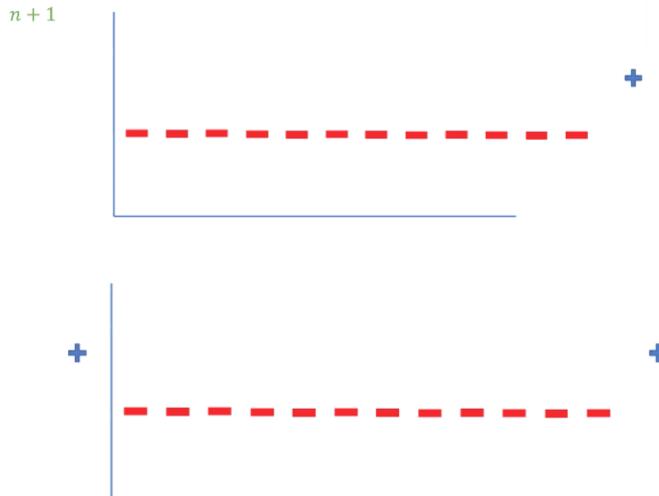


- When will Full Bayes get better results compared to Naive Bayes? Explain

- Naive Bayes assumption is that the features are independent given the class
- Full Bayes can achieve better results only when there is some dependency between the features given the class
- But, this is not enough. Full Bayes will get better results only when ignoring this dependency will change the prediction = Naive Bayes will get max posterior for different class

\* Obviously, most of the time Full Bayes is not practical and therefore we will use only Naive Bayes

Suppose we train a hard-margin linear SVM on  $n > 100$  data points in  $\mathbb{R}^2$ , yielding a hyperplane with exactly 2 support vectors. If we add one more data point and retrain the classifier, what is the maximum possible number of support vectors for the new hyperplane (assuming the  $n + 1$  points are linearly separable)?



Let  $U = \mathbb{R}$  and  $H$  be a finite union of intervals on the line.

What is the VC dimension of  $H$ ?

- $VC(H) = \infty$
- Why? For every  $m$  we can pick the set  $X = \{1, 2, 3, \dots, m\}$  then for every labeling  $l$  we simply pick the hypothesis :  $h = \{(a_i - 0.1, a_i + 0.1) : a_i \in X \wedge l(a_i) = 1\}$

