סיכום IML

30 ביוני 2022

1 שבוע 1:

1.1 תרגול 1 ־ חזרה על לינארית:

1.1.1 נורמות:

- **מטריקה ־ הגדרה:** מעל קבוצה X נגדיר מטריקה $d : X \times X \Rightarrow R$, ומקיימת שלשה תנאים:
1: $x = y \iff d(x, y) = 0$. **2:** $d(x, y) = d(y, x)$. **3:** $d(x, y) \le d(x, z) + d(z, y)$.

- **נורמה ־ הגדרה:** פונקציה, נסמנה כך: $|| \cdot || : R^d \Rightarrow R_{\ge 0}$ ותקיים שלשה דברים:
1: $V = 0 \iff ||V|| \ge 0, ||V|| = 0$. **2:** $||\alpha V|| = |\alpha| \cdot ||V||$. **3:** $||V + U|| \le ||V|| + ||U||$.

- **שלשת הנורמות בהם נשתמש:**
1 ־ נורמת 1: $||V||_1 = \Sigma_i |v_i|$.
2 ־ הנורמה האוקלידית: $||V||_2 = \sqrt{\Sigma_{i=1}^n v_i^2}$.
3 ־ נורמת אינסוף: $||V||_\infty = max_i |V_i|$.

- **הגדרה ־ כדור היחידה:** $B = \{X : ||X|| \le 1\}$.

- **אי שוויון ההפוך:** $\|U\|_1 \ge \|V\|_2 \ge \|V\|_\infty$ (לא הוכחנו).

- **הגדרה ־ וקטור מנורמל:** נסמן $\hat{V}$, ומתקיים $\hat{V} = \frac{V}{||V||}$.

1.1.2 מרחבי מכפלה פנימית:

- **מכפלה פנימית:** פונקציה שמקבלת שני איברים מהשדה ומחזירה סקלאר. $\langle \cdot, \cdot \rangle \Rightarrow R$. ומקיימת שלש תכונות:
1 סימטריה : $\langle u, v \rangle = \langle v, u \rangle$. **2 לינאריות :** $\langle \alpha V + U, W \rangle = \alpha \langle V, W \rangle + \langle U, W \rangle$ **3 חיוביות:** $\langle V, V \rangle \ge 0, \langle V, V \rangle = 0 \iff V = 0$.

- **הגדרה ־ מ"פ סטנדרטית:** $\langle v, u \rangle = \sum_i v_i u_i = v \cdot u$.

- **משפט הקוסינוסים:** $a^2+b^2-2ab\cdot cos(\theta)=c^2$. עבור θ הזווית בין a,b.
באותו האופן:$\|v\|^2+\|u\|^2-2\|v\|\|u\|\cdot\cos(\theta)=\|v-u\|^2$. עבור θ הזווית בין v,u

- **זהות:** $\|v-u\|^2==\|v\|^2+\|u\|^2-2\langle v,u\rangle$

- **מסקנה:** $\cos(\theta)=\frac{\langle v,u\rangle}{\|v\|\|u\|}$.

- **הגדרה - ניצב (אורתוגונלי):** $V\perp U \iff \langle V,U\rangle=0$.
הערה: וורטוק האפס אורתטגונלי להכל.

1.1.3 טרנספורמציות לינאריות:

- **הגדרה - העתקה לינארית:** עבור שני מ"ו V,W נגדיר העתקה לינארית כ: $T:V\Rightarrow W$ אם היא מקיימת את התכונות הבאות:
1 אדטיביות: $T(U+V)=T(U)+T(V)$. **2 כפל בסקלר:** $T(\alpha V)=\alpha\cdot T(V)$.

- **הגדרה - טרנספורמציה אפינית:** היא טרנספורמציה לינארית + וקטור. עבור ווקטור V,W מתקיים: $F(V)=T(V)+W$.

1.1.4 מטריצות:

- **מטריצות:** עבור מטריצה $A\in R^{m*n}$ מתקיים:
1: $Ker(A)=\{x:Ax=0\}$.
2: $Im(A)=\{Ax:x\in R^m\}=\{w\in W\mid \exists x\in V \quad Ax=W\}$ כל צ"ל של העמודות של A.
3: $Im\left(A^T\right)=R(A)$ צירוף כל השורות של A.
4: $Ker\left(A^T\right)$.

- **הגדרה - דרגה** $(rank)$ **של מטריצה:** עבור מטריצה $A\in R^{m\cdot d}$ אזי $rank(A)\leq min(m,d)$.
ואם $rank(A)=min(m,d)$ נאמר שהיא **מדרגה מלאה.**

- **הגדרה - ע"ע:** יהי V מ"ו ו T טרנספורמציה לינארית, אם קיים ווקטור $0\neq v\in V$ וסקלר λ כך שמתקיים: $T(v)=\lambda v$ אזי λ הוא ע"ע של T ו v הוא ו"ע השייך לע"ע λ.
עבור מטריצות: תהי $A\in F^{n\cdot n}$ מטריצה **ריבועית**, ווקטור $v\in F^n$ אם קיים סקלר λ כך ש $Av=\lambda v$ אזי v יקרא ו"ע של ע"ע λ.

- **טענות:**
1: $Ker(A)\perp Im\left(A^T\right)$. **2:** $Ker\left(A^T\right)\perp Im(A)$.

- **הגדרה - בסיס אורתונואמלי:** $||V_i||=1$ וגם $\forall i\neq j,\langle V_i,V_j\rangle=0$

- **הגדרה - מטריצה סימטרית:** $A=A^T$

- **הגדרה - מטריצה הפיכה:** עבור מטריצה **ריבועית** $A \in R^{m \cdot m}$ נאמר שמטריצה $B \in R^{m \cdot m}$ הופכית ל A אם $AB = I_m$.
ונאמר ש A, B הופכיות זו לזו.

- תהי $X \in R^{m \cdot m}$ ווקטור $Y \in R^m$ אנו רוצים למצוא ווקטור W כלשהו כך ש $XW = Y$.
עבור A הפיכה מתקיים כי $W = X^{-1}Y$.

- **אם** $A \in R^{m \cdot m}$ **הפיכה התנאים הבאים שקולים:**
1: $rank(A) = m$. **2:** $Im(A) = Row(m)$ (כל ווקטור ב $Row(m)$ יכול לצאת כתוצאה של הטרנספורמציה). **3:** $Ker(A) = V_0$.

1.1.5 הטלות אורתוגונליות:

- **הגדרה - הטלה:** וקטור P ההטלה של U על V - הוא הנקודה הקרובה ביותר לווקטור U שנמצא ב $span\left(V\right)$. נחשב כך:

$$P = \langle U, \hat{V} \rangle \hat{V} = \frac{\langle U, V \rangle}{\|V\|^2} \cdot V$$

- **הגדרה - קבוצה אורתונואמלית:** כל $v \in V$ מקיים כי $||v|| = 1$ וגם הם אורתוגונלים (ניצבים זה לזה).

- **הגדרה - מכפלה חיצונית:** $V \otimes U = V \cdot U^T$, התוצאה היא מטריצה כך שבפינה השמאלית עליונה נמצא $u_1 \cdot v_1$ ובפינה הימנית תחתונה $u_n \cdot v_n$.

- **מטריצות הטלה:** יש לנו ת"מ לינארי R^n ואנו רוצים למצוא לכל וקטור את הנקודה שהכי קרובה אליו בת"מ שלנו.
ונעשה זאת בעזר כפל במטרצית הטלה שתמונתה הוא ת"מ שמעניין אותנו.
יהי $V \subseteq R^n$ (ת"מ של R^n) והוא $k - dim$. יהיו $v_1 ... v_k$ בסיס אורתונואמלי.
מטריצת ההטלה P **מוגדרת האופן הבא:** $P = \sum_{i=1}^{k} V_i \otimes V_i = \sum_{i=1}^{k} V_i \cdot V_i^T$.

- **תכונות של מטריצת הטלה:**
1 סימטריה: $P = P^T$.
2: $P^2 = P$.
3: יש לה k ע"ע של 1 (הוקטורים v_i שראינו בהגדרה) וכל השאר של 0.
4: $(I - P) \cdot P = 0$, ($(I - P)$ היא גם מטריצת הטלה ומטילה על המרחב שניצב אלינו).
5: $\forall x \in R^n, \forall u \in V$ מתקיים: $\|x - Px\| \leqslant \|x - u\|$. כלומר - לכל ווקטור ב V, המרחק האוקלידי בינו לבין x גדול שווה מהמרחק של x מהטלה אורתוגונלית אל אותו המרחב.
6: $\forall u \in V, Pu = P$.

1.1.6 מטריצות חיוביות בהחלט ולמחצה, ופירוקים:

- **הגדרה - מטריצת** PSD**: מטריצה סימטרית** תיקרא PSD ונסמן $A \succeq 0$ אם לכל וקטור x מתקיים $x^T A x \geq 0$.

- **הגדרה - מטריצת** PD**:** מטריצה תיקרא PD ונסמן $A \succ 0$ אם לכל וקטור $x \neq 0$ מתקיים $x^T A x > 0$.

- **שקילויות ל** PSD**:**
1: $x^T A x \geq 0$. **2:** $\lambda_i(A) \geq 0$ לכל i. **3:** $\exists B : B^\top B = A$.
עבור λ ע"ע.

- **שקילויות ל** PD**:**
1: $x^T A x > 0$ עבור $x \neq 0$. **2:** $\lambda_i(A) > 0$ לכל i. **3:** $\exists B : B^\top B = A$, עבור B הפיכה.
מסקנה: מטריצת PD תמיד הפיכה (כי כל ע"ע שלה חיוביים ולכן הגרעין שלה ריק).
מסקנה: $PD + PSD = PD$.

- **הגדרה - מטריצה לכסינה:** מטריצה ריבועית A תיקרא לכסינה אם קיים בסיס כך ש $D = P^{-1}AP$ עבור D מטריצה אלכסונית.

- **הגדרה - מטריצה אורתוגונלית:** A אורתוגונלית אם $A \cdot A^T = I$, או $A = A^{-1}$.

- **פירוק** EVD**:** אם $A = A^T$ אזי $A = UDU^T$ עבור U מטריצה אורתוגונלית (השורות והעמודות שלה הן בסיס אורתונורמלי), ו D מטריצה אלכסונית.
טענה: $A^n = UD^nU^T$.

- **משפט - איזומטריה:** לכל x ולכל מטריצה U אורתוגונלית מתקיים - $||Ux|| = ||x||$

- **המשפט הספקטרלי -** EVD **- פירוק לע"ע:** תהי A מטריצה ריבועית סימטרית, אזי A לכסינה, והבסיס המלכסן שלה הוא אורתוגונלי.

- **הגדרה - ערכים סינגולרים:** עבור מטריצה $A \in R^{m \cdot n}$, נגדיר - u וקטור סינגולרי משמאל, v וקטור סינגולרי מימין, וערך סינגולרי $\sigma \in R^+$. אם $Av = \sigma u$.
כלומר - זאת הרחבה ל ע"ע **לכל מטריצה** ולא רק למטריצה ריבועית.

- **פירוק** SVD **- פירוק לערכים סינגולרים:** כל מטריצה $A \in R^{m \cdot n}$ ניתנת להצגה כך - $A = U\Sigma V^T$, עבור: U, V^T אורתוגונליות, ו Σ אלכסונית (לא בהכרח ריבועית, האלכסון הראשי בלבד יכול להכיל ערכים שונים מ 0).
למעשה פירוק זה אומר כי - ניתן לתאר כל העתקה לינארית כסיבוב, מתיחה לפי ערכים סינגולרים וסיבוב במרחב אחר.

- **כיצד נפרק מטריצה** A **ל** SVD**:**
1: נחשב את $Y = A^T A$.
2: נמצא את הערכים והווקטורים העצמיים של המטריצה Y.
3: נוציא שורש מהערכים העצמיים שמצאנו - אלו הערכים הסינגולריים
4: המטריצה Σ היא מטריצה שעל האלכסון הראשי שלה יש את הערכים הסינגולריים השונים מ 0.
5: הווקטורים העצמיים של AA^T הם העמודות של המטריצה U.
6: את המטריצה U ניתן למצוא כך: $U = AV\Sigma^{-1}$. בנוסף הו"ע של $A^T A$ הם הווקטורים הסינגולרים הימניים - המטריצה V.

1.2 הרצאה 1:

1.2.1 סטטיסטיקה - שיערוך\אמידה:

- **הגדרה - אומד** $Estimator$: עבור m דגימות האומד של הדגימות הוא $\bar{x}=\frac{1}{m}\sum_i x_i$.
  הדגימות יקראו מדגם, כל אחת מהדגימות תקרא $example$ והסט יקרא $Sample$.

- ניתן להתייחס אל הדגימות כאל תוצאות של מ"מ (יכול להשתנות לפי הקירוב).

- אנו מניחים שהדגימות מגיעות מפונקציית התפלגות הסתברותית כלשהי נסמנה ב P.

- נאמר שהדגימות שוות התפלגות אם הן מגיעות מאותה ההתפלגות.

- **הגדרה -** iid: נאמר שהדגימות iid אם הן ב"ת ושוות התפלגות. **הגדרה -** $X_1,\ldots,X_m \overset{i.i.d}{\sim} \mathcal{P}, \quad \forall i X_i = x_i$

- **פרמטר ההתפלגות:** לכל התפלגות יש פרמטר $\theta\in\Theta$ שניתן לדעת את ההתפלגות על פיו, הפרמטר מגיע מתוך סט מסויים. Θ יהיה וקטור הפרמטרים.
  נרצה להתאים את θ^* שמתאים הכי טוב לפרמטר האמיתי θ, נעשה זאת באמצעות הפונקציה $\delta:\mathbb{R}^m\to\Theta$.
  הסט של הפונקציות הרלוונטיות יסומן כך $\Delta:=\{\delta:\mathbb{R}^m\to\Theta\}$. נקרא לה **מחלקת ההיפותזה,** פלט הפונקציה נקרא **האומד.**

- **האומד של התוחלת הוא:** $\hat{\mu}_X := \frac{1}{m}\sum x_i$.

- **האומד של השונות:** $\hat{\sigma}^2_X := \frac{1}{m-1}\sum\left(x_i-\hat{\mu}_X\right)^2$.

- **הגדרה - הטעות של האומד :** $d:=\delta\left(X_1,\ldots,X_m\right)-\theta$.

- **הגדרה - הטיה** $(Bias)$:

$$Bias_\theta\left[\delta\left(X_1,\ldots,X_m\right)\right] := \mathbb{E}_{X_1,\ldots,X_m|\theta}[d] = \mathbb{E}_{X_1,\ldots,X_m|\theta}\left[\delta\left(X_1,\ldots,X_m\right)-\theta\right]$$

- **הגדרה - אומד חסר הטיה:**

$$\forall\theta\in\Theta \quad Bias_\theta\left[\delta\left(X_1,\ldots,X_m\right)\right]=0$$

(ההטיה שלו שווה ל 0, כלומר מרחב המדגם דגם את האוכלוסיה על הצד הטוב ביותר והאומדן היה קרוב למציאות).

- **הגדרה - שונות** $Variance$:

$$Var(\delta) := \mathbb{E}_{X_1,\ldots,X_m|\theta}\left[\left(\delta\left(X_1,\ldots,X_m\right)-\mathbb{E}_{X_1,\ldots,X_m|\theta}\left[\delta\left(X_1,\ldots,X_m\right)\right]\right)^2\right.$$

שונות של מ"מ:

$$Var(A)=\mathbb{E}\left[A^2\right]-\mathbb{E}^2[A]$$

- **הגדרה ־ פונקציית הנראות** $(likelihood)$**:** יהי $X \sim P(\theta)$, ותהי f פונקצית הצפיפות, אזי פונקציית הנראות היא:

$$\mathcal{L}(\boldsymbol{\theta} \mid x) := f_{\boldsymbol{\theta}}(x)$$

- **הגדרה ־ אומד נראות מירבית:**

$$\hat{\boldsymbol{\theta}}^{MLE} := \underset{\boldsymbol{\theta}\in\Theta}{argmax}\mathcal{L}(\boldsymbol{\theta} \mid x)$$

כלומר ־ נחפש את האומד שימקסם את פונקציית הנראות.

תכונה ־ ניתן לבדוק מקסימום גם על log של הפונקציה.

- **תכונות של אומדים ־ סיווג לאומד טוב:**

δ היא פונקציה שמופעלת על המדגם, לכן הפלט של הפונקציה הוא גם מ"מ ולכן יש לו התפלגות, נשתמש בה כדי להגדיר דברים שמעניינים אותנו.

1: אומד חסר הטיה ־ בתוחלת הוא מתכנס לפרמטר האמיתי אותו אנו אומדים.

2: פונקציית הנראות.

- **א"ש מרקוב:**

$$\mathbb{P}(X \geq a) \leq \frac{\mathbb{E}[X]}{a}$$

- **א"ש צ'בישב:**

$$\mathbb{P}(|X - \mathbb{E}[X] \geq \varepsilon|) \leq \frac{Var(X)}{\varepsilon^2}$$

- **הגדרה ־ התפלגויות מרובות משתנים (ווקטור מקרי):** נסתכל כאן על מ"מ **שאינו** ערך בודד אלא ווקטור של מ"מ.

יהי $X_1, \ldots, X_d$ סט סופי של מ"מ מעל אותו מרחב הסתברות. הווקטור המקרי $X := (X_1, \ldots, X_d)^\top$ הוא מיפוי של מרחב ההסתברות ל R^d.

- **הגדרה ־ השונות המשותפת (**$covariance$**):** יהי $X := (X_1, \ldots, X_d)^\top$ ווקטור מקרי, השונות המשותפת תוגדר כך:

$$\Sigma_{ij} := COV\left(X_i, X_j\right) = \mathbb{E}\left[\left(X_i - \mathbb{E}\left[X_i\right]\right)\left(X_j - \mathbb{E}\left[X_j\right]\right)\right]$$

- **הגדרה ־ התפלגות נורמלית של ווקטור מקרי:** נאמר ש ו"מ מתפלג נורמלי אם:

$$f(X) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp\left\{-\frac{1}{2}(X-\mu)^\top \Sigma^{-1} (X-\mu)\right\}$$

במקרה זה נסמן $X \sim \mathcal{N}(\mu, \Sigma)$.

- **הגדרה ־ הסתברות שולית של ו"מ:** יהי $X := (X_1, \ldots, X_d)^\top$ ווקטור מקרי, ונגדיר תת קבוצה של קאורדינטות $A \in [d]$ and $B = [d] \backslash A$. אזי

$$f(X_A) := \int_{X_B} f(X_A, X_B)\, dX_B$$

- **טענה עבור התפלגות נורמלית:**

Claim: Let $X \sim \mathcal{N}(\mu, \Sigma)$ be a bivariate Gaussian, i.e $X = (X_1, X_2)^\top$, with a diagonal covariance matrix:

$$X \sim \mathcal{N}\left(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}\right)$$

The marginal distribution of coordinate i is

$$f(X_i) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{1}{2}\left(\frac{X_i - \mu_i}{\sigma_i}\right)^2\right)$$

- **ייצוג מידגם בעזרת מטריצה:** במטריצה השורות יהיו הנתונים עבור כל אובייקט, והעמודות יהיו התכונות. נתייחס ל $X_{i,j}$ כתכונה ה j של האובייקט ה i.

- **האומד של השונות המשותפת עם ובלי $bias$:**

In matrix notation, for $\mathbf{X} \in \mathbb{R}^{m\times d}$ whose rows are the samples $\mathbf{x}_1, \ldots, \mathbf{x}_m$ the:

- biased sample covariance matrix is given by

$$\hat{\Sigma} := \frac{1}{m}\sum_{i=1}^{m}(\mathbf{x}_i - \hat{\mu})(\mathbf{x}_i - \hat{\mu})^\top = \frac{1}{m}\widetilde{\mathbf{X}}^\top\widetilde{\mathbf{X}}$$

  for $\widetilde{\mathbf{X}}$ being the centered matrix: $\widetilde{\mathbf{X}}_{\cdot,i} := \mathbf{X}_{\cdot,i} - \hat{\mu}$.

- unbiased sample covariance matrix is given by

$$\hat{\Sigma} := \frac{1}{m-1}\sum_{i=1}^{m}(\mathbf{x}_i - \hat{\mu})(\mathbf{x}_i - \hat{\mu})^\top = \frac{1}{m-1}\widetilde{\mathbf{X}}^\top\widetilde{\mathbf{X}}$$

2 שבוע 2:

2.1 תרגול 2 - חזרה על אינפי:

- **נגזרת:** נגזרת $\frac{df(x)}{dx} = lim_{a\Rightarrow 0}\frac{f(x+a)-f(x)}{a}$.
- **פונקציה רב מימדית:** פונקציה $f: R^d \Rightarrow R$ (הפונקציה f מקבלת ווקטור ומחזירה סקלר).
- **נגזרת חלקית:** $\frac{\partial f(x)}{\partial x} = lim_{a\Rightarrow 0}\frac{f(x+a\cdot e_i)-f(x)}{a}$.
- **גרדיאנט:** ווקטור הנגזרות החלקיות - $\nabla f = \left[\frac{\partial f(x_1)}{\partial x}, ..., \frac{\partial f(x_d)}{\partial x}\right] \in R^d$ (אינטואיציה - מצביע לכיוון העליה הכי גדולה בפונקציה).

  הערה: אם הגרדיאנט שווה ל 0 אזי אנו בנק' קיצון.

- **יעקוביאן** - $Jacobian$: תהי $f: R^d \Rightarrow R^m$, ניתן לכתוב את הפונקציה בתור ווקטור $f(x) = [f_1(x), ..., f_m(x)]$. יעקוביאן היא מטריצה ממימד $m\cdot d$, שתסומן ב $J_x(f)$, שמכילה את כל הנגזרות החלקיות. כלומר $J_{i,j} = \frac{\partial f_i(x)}{\partial x_j}$.

למעשה מתקיים כי $J_x(f) = \begin{bmatrix} (\nabla f_1)^T \\ . \\ . \\ (\nabla f_m)^T \end{bmatrix}$

הערה: מתקיים $J_x(f) = (\nabla f)^T$, כלומר ־ היעקוביאן הוא טרנספוז של הגרדיאנט.

הערה: יעקוביאן של פונקציה $f(x) = Ax$ עבור A מטריצה, שווה ל $J(Ax) = A$.

- **הסיאן ־ $Hessian$:** תהי פונקציה $f : R^d \Rightarrow R$ **גזירה פעמיים**, אזי המטריצה $H(f) = \nabla^2(f)$ היא נגזרת של f פעמיים, פעם אחת לפי x_i ופעם שניה לפי x_j.
כלומר $H_{i,j} = \frac{\partial f^2}{\partial x_i \partial x_j}$.
טענה: $H(f) = H(f)^T$ כלומר ־ הסיאן היא מטריצה סימטרית.

- **כלל השרשרת:** נגזרת עבור הרכבה של פונקציות $f(g(x)) = (f \circ g)(x)$: אזי הנגזרת היא:

$$\frac{\partial f \circ g}{\partial x} = f'(g(x)) \cdot g'(x)$$

- **מטריצת יעקוביאן להרכבה של פונקציות:** $J_{\mathbf{x}}(f \circ g) = J_{g(\mathbf{x})}(f) J_{\mathbf{x}}(g)$.

- **הפונקציה $Softmax$:** פונקציה $S : R^d \Rightarrow R^d$, ומוגדרת כך (עבור ווקטור x):

$$S(x) = \frac{e^{x_k}}{\sum_j e^{x_j}}$$

למעשה היא מקרבת לנו את המקסימום.

2.1.1 קירובים:

נבחר נקודה שסביבה אנו רוצים לקרב $- x_0$.

- **קירוב טיילור:** $T_f(x) = \sum_{n=0}^{\infty} f^{(n)}(x_0) \frac{(x-x_0)^n}{n!}$.
קירוב מסדר ראשון: $f(x) \approx f(v_0)' + f(v_0)(v - x_0)$.
קירוב מסדר שני: $f(x) \approx f(v_0) + f(v_0)'(v - x_0) + \frac{1}{2} f(v_0)^{"} (v - x_0)^2$.

- **הכללה:** עבור פונציה $f : R^d \Rightarrow R$.
קירוב מסדר ראשון: $f(x) \approx f(v_0) + \langle \nabla(x_0), x - x_0 \rangle$.
קירוב מסדר שני: $f(x) \approx f(v_0) + \langle \nabla(x_0), x - x_0 \rangle + \frac{1}{2}(x - x_0)^T \cdot H \cdot (x - x_0)$.

2.1.2 קמירות $Convex$:

- **הגדרה ־ קבוצה קמורה:** באופן אינטואיטיבי ־ קבוצה קמורה היא סט של נקודות, כך שכל שתי נקודות שנבחר, הקו שעובר בניהן נמצא בקבוצה.
פורמלית: קבוצה $C \subseteq R^d$ קמורה אם לכל v, u ולכל $\alpha \in [0,1]$ מתקיים כי $\alpha v + (1-\alpha)u \in C$.

- **טענה:** כל נורמה היא קבוצה קמורה.
- **תכונות:**
  1: חיתוך של קבוצות קמורות גם קמור.
  2: סכום של קבוצות קמורות היא גם קבוצה קמרה.
  3: כפל בסקלר עם קבוצה קמורה היא קבוצה קמורה גם כן.
- **פונקציות קמורות:** פונקציה $f : C \Rightarrow R$ תיקרא קמורה אם לכל שתי נקודות על גרף הפונקציה, הקו המחבר בניהן יהיה מעל גרף הפונקציה.
  פורמלית: אם לכל v, u ולכל $\alpha \in [0,1]$ מתקיים כי $f\left(\alpha v + (1-\alpha)u\right) \le \alpha f\left(v\right) + (1-\alpha)f(u)$.
- **תכונה:** פונקציה קמורה מקיימת כי נקודות הקיצון שלה הן גלובליות.
- **טענה:** נורמה היא פונקציה קמורה.
- **תנאי שקול לקמירות של פונקציה:** תהי פונקציה $f : R^d \Rightarrow R$, היא קמורה אמ"מ $H(f) \succeq 0$ (אם ההסיאן של הפונקציה היא מטריצת PSD).
- **הגדרה - $epi\ graph$:** השטח שנמצא מעל גרף הפונקציה.
- **טענה:** אם $epi\ graph$ היא קבוצה קמורה, אזי הפונקציה קמורה.

2.2 הרצאה 2:

- **ההבדל בין מערכות לומדות לתכנות פרוצדורלי:** אוספים דוגמאות של איך לבצע את המשימה כך שהמחשב לומד איך לבצע משימה בפעם הבאה שהוא יראה אותה, כמו כן המחשב לומד להגיב לשינויים במשימה. בנוסף מתקיים שיפור בכל פעם והמערכת לומדת להשתפר.
- **למידה מפוקחת - $Superviser$:** עבור קלט $x \in X$ (X נקרא **הדומיין**) נאמן את המערכת על קלטים דומים עם פונקציה $f(x)$ הנקראת $rule$ שמחזירה $y \in Y$ (הסט Y נקרא התגובה - $response$). לאחר האימון נציג למערכת קלט x, המערכת תפעיל עליו פונקציה $\hat{f}(x)$ ואם מתקיים שוויון בין הפונקציות $f(x), \hat{f}(x)$ אזי המערכת למדה.
- **בעיות רגרסיה:** כאשר $Y \in R$ נאמר שהבעיה היא בעיית רגרסיה.
- **תופעת $no\ free\ lunch$:** תופעה שאומרת שעבור סט של דגימות אם לא נניח שומדבר על הפונקציה f , לא נצליח ללמוד.
- **מחלקת היפותזות:** נניח שהפונקציה $f \in \mathcal{H}$ קבוצת פונקציות. לקבוצה $\mathcal{H}$ נקרא מחלקת היפותזות.
- **$Batch\ learning$:**
  נבחר מחלקה היפותזית.
  נכין סט אימון $\{(x_i, y_i)\}_{i=1}^{m}$.

נלמד את הפונקציה $f \in \mathcal{H}$.
כאשר יתקבל דאטה חדש - $\{\tilde{x}_j\}_{j=1}^k$ נבצע את הפרדיקציה $\{\hat{y}_j\}_{j=1}^k$ כאשר $\hat{y}_j = h(\tilde{x}_j)$.

- $online\ learning$**:** מקבל דגימה ואמור לתת תחזית.
- מסתבר שקיימות מערכות שמקבלות רק x־ים ללא y־ים והן מגיבות בלי שתהיה להן $response$ כלל.
- **בקורס שלנו נתעסק ב:** $Batch\ learning$. הדומיין של הדגימות שלנו יהיה $X \in R^d$. נתעסק בבעיות רגרסיה $Y = R$. נתעסק בבעיות קלסיפיקציה $Y = \pm 1$ (בינארי).

2.2.1 המודל הלינארי:

- **מחלקת ההיפותזה הלינאריות:**

$$\mathcal{H}_{\text{lin}} = \left\{ (x_1, \ldots, x_d) \mapsto w_0 + \sum_{i=1}^{d} x_i w_i \mid w_0, w_1, \ldots, w_d \in \mathbb{R} \right\}$$

$w_1 ... w_n$יקראו **משקולות.** w_0 יקרא **החותך**.
נשים לב: ההעתקה כאן היא **העתקה אפינית** (ה"ל עם היסט w_0)

- ללמוד את הפונקציה h הכוונה היא ללמוד את המשקולות
- **נגדיר בריגרסיה לינארית:** בקורדינטה ה 0 של הווקטור X מופיעה הספרה 1 .
בנוסף - $w = (w_0, w_1, \ldots w_d)^\top$ הקאורדינטה הראשונה ב w היא w_0.
לכן מחלקת ההיפותזה הלינאריות:

$$\mathcal{H}_{\textbf{lin}} = \left\{ \mathbf{x} \mapsto \mathbf{x}^\top \mathbf{w} \mid \mathbf{w} \in \mathbb{R}^{d+1} \right\}$$

- **הנחות:**

אלגוריתם הלמידה ייצר כלל החלטה $h_s \in H_{lin}$ ־ כלומר כלל הלמידה תלוי במדגם האימון S. (סימון נוסף הוא $\hat{f}$)
נניח שקיימת f כך ש $y = f(x)$.
המקרה הרליזבילי: אם קיימת $f \in H_{lin}$. ומתקיים $y_i = x_i^T w \quad for \quad i = 1...m$ וקיים $w \in R^{d+1}$. (נמצא את w ע"י פתרון מערכת משוואות)
המקרה הרליזבילי בצורה מטריציונית: $Xy = w$ ומתקיים $y \in Im(X)$. וקיים פתרון יחיד רק אם עמודות המטריצה בת"ל, ויש אינסוף פתרונות אמ"מ יש תלות לינארית בין העמודות.

- **במקרה הלא הרליזבילי:** נגיד שהמערכת הלומדת חייבת לייצר $h \in H$, אך לא בהכרח שהיא לינארית ויכול להיות שאין פתרון למערכת המשוואות (לא קיים w שמקרב) ומתקיים $y \notin Im(X)$.
- **הגדרה - פונקציית** $Loss$**:** פוקנציה שמקבלת את ה $response$ האמיתי ואת ה $resonse$ שחזינו ומחזירה את ההפרש ביניהם. $L(y, h(s)) = |y - h(s)|$

- **הגדרה - פונקציית** $Squared\ Loss$:

$$L(y,h(s)) = (y-h(s))^2$$

- **עיקרון ה** $Empirical\ Risk\ Minimization - ERM$: עקרון שמראה לנו איזה מחלקת היפותזות לבחור על פי הדגימות שקיבלנו.

נסכום באופן הבא: ונבחר את הפונקציה h שמחזירה לנו את הסכום המינימלי

$$\sum_{i=1}^{m} L(y_i, h_s(x_i))$$

במקרה שלנו:

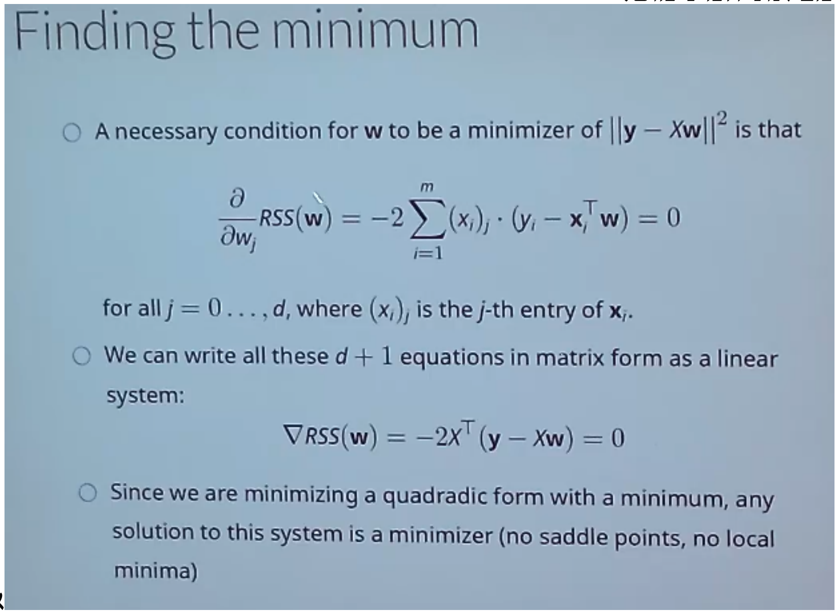
$$\sum_{i=1}^{m}\left(y_i - x_i^\top \mathbf{w}\right)^2 = \|\mathbf{y}-X\mathbf{w}\|^2 = \left(\mathbf{y}-X_{\mathbf{w}}\right)^\top\left(\mathbf{y}-X_{\mathbf{w}}\right)$$

סכום ריבועי הסטיות מהאמת נקרא $sum\ of\ squares$

- **הגדרה -** $residual$: $y_i - \mathbf{x_i}^\top \mathrm{w}$
- **הגדרה -** RSS: $RSS(\mathbf{w}) = \|\mathbf{y}-X\mathbf{w}\|^2$
- **פתרון במקרה הלא הרליזבילי:** נשתמש תמיד בפתרון זה כי אין לנו אינדיקציה האם המערכת רליזבילית.

נמצא את הפונקציה המינימלית ולאחר מכן נמצא את המינימום שלה ע"י נגזרת.

מציאת המינימום:



אנו מנסים למצוא פתרון למערכת המשוואות הבא:

$$\nabla RSS(\mathbf{w}) = -2X^\top(\mathbf{y}-x\mathbf{w}) = 0 \iff X^\top \mathbf{y} = X^\top X w$$

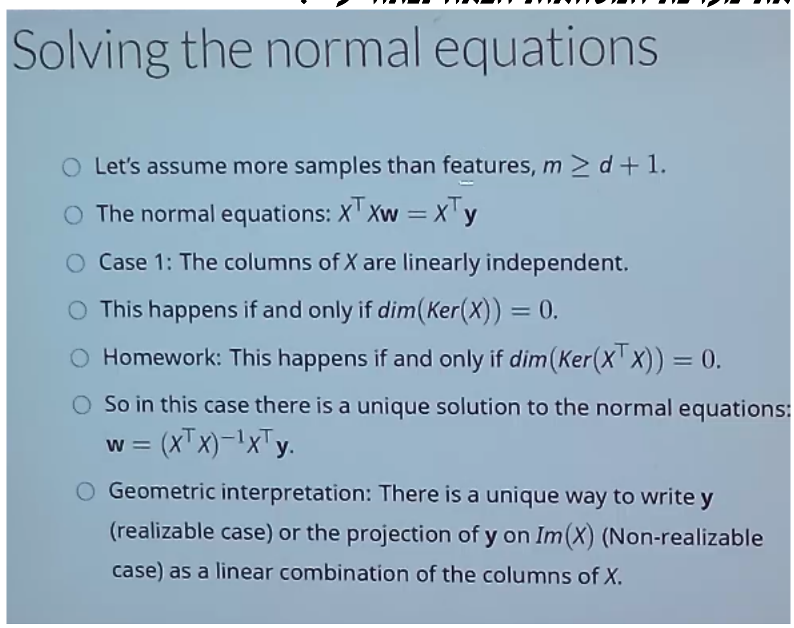
- **המשוואות הנורמליות:**

$$X^\top \mathbf{y} = X^\top X w$$

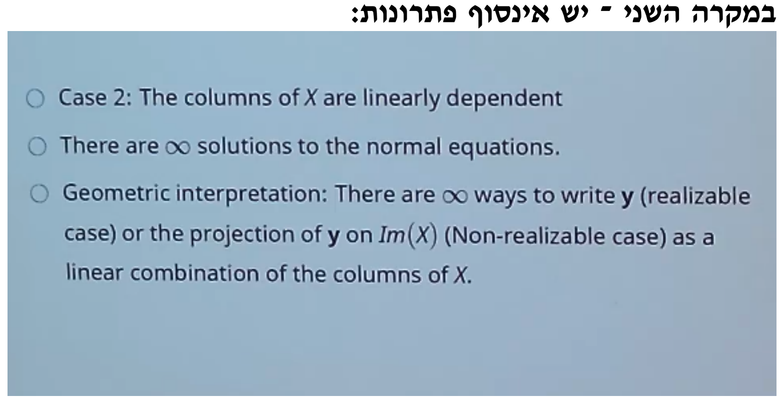
מערכת משוואות הנורמליות מתקבלת ע"י גזירת RSS.
ומתקיים שוויון לכל $j = 1...d$ עבור φ_j העמודה ה j של המטריצה.

$$X^\top \mathbf{y} = X^\top X w \iff \langle \varphi_j, \mathbf{y} - Xw \rangle = 0$$

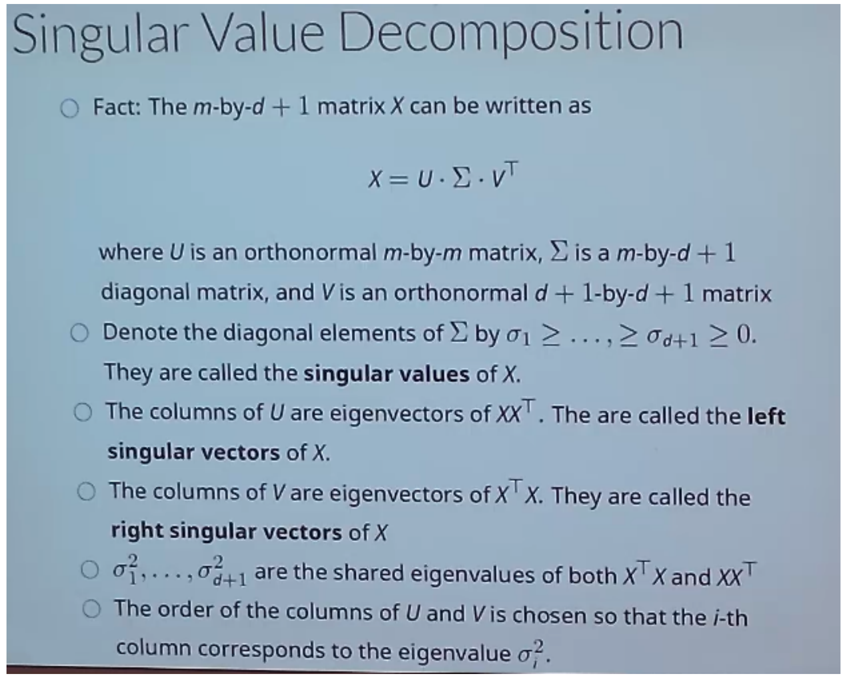
את מערכת המשוואות הבאה נפתור ע"י:



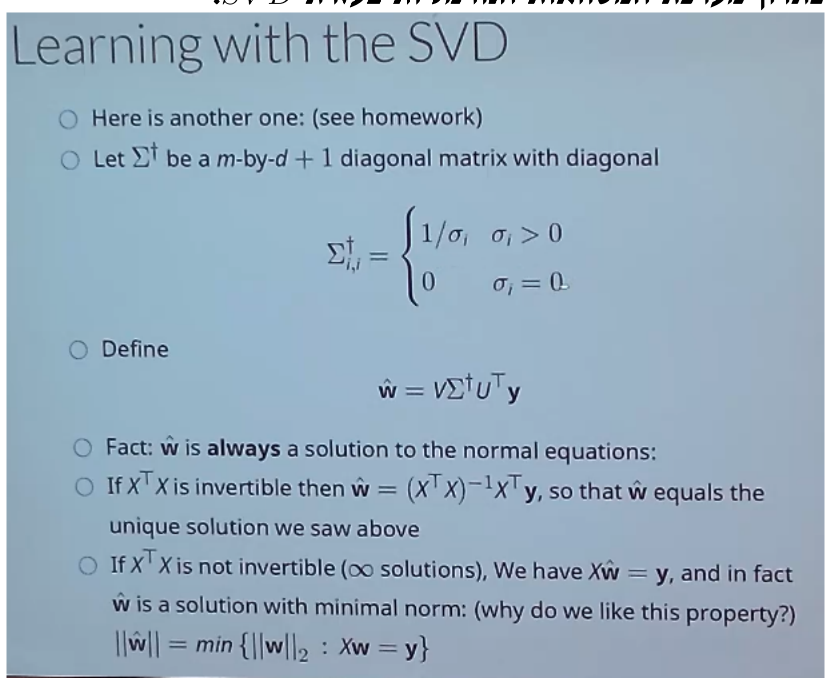
במקרה השני - יש אינסוף פתרונות:



את הפתרון נעשה עם SVD:



- **פתרון מערכת המשוואות הנורמליות בעזרת** SVD:



- **משפט:** $\hat{w}$ הוא **תמיד** פתרון, גם במקרה הרליזבילי וגם במקרה הלא רליזבילי.
בנוסף: יש לו את הנורמה המינימלית.

- **נשים לב:**
ניצבות: מכיוון שמרחב העמודות פורשות את ת"מ $Im(X)$, מתקיים $\mathbf{y} - X\hat{\mathbf{w}} \perp Im(X)$.
נגדיר: $\hat{\mathbf{y}} = X\hat{\mathbf{w}} \in Im(X)$.

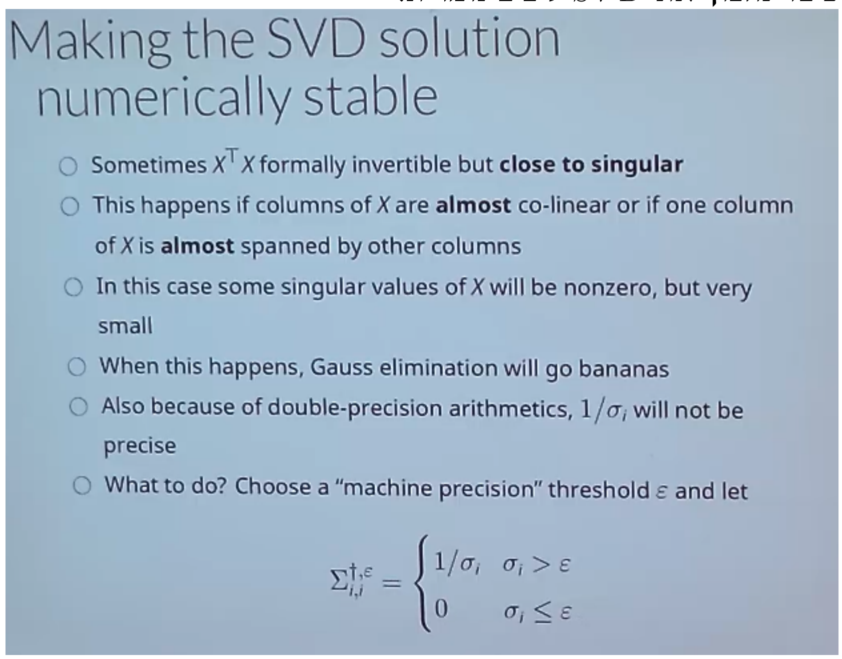
נגדיר: $\hat{\mathbf{z}} = \mathbf{y} - \hat{\mathbf{y}} \in Im(X)^{\perp}$.

במילים אחרות: $x\hat{w}$ הוא הטלה אורתוגונלית של y על $Im(X)$.

לפי משפט פיתגורס: $\|\mathbf{y}\|^2 = \|\hat{\mathbf{y}}\|^2 + \|\hat{z}\|^2$. **כלומר -** הנורמה של הווקטור w בריבוע = לנורמה של ההטלה בריבוע + הנורמה של ההפרש בריבוע.

- **הגדרה - אלגוריתם יציב נומרית:** אלגוריתם שלא מחזיר תוצאות רחוקות מהאמת אף על פי שהמטריצה קרובה מאד למטריצה לא הפיכה, והערכים הסינגולרים שלה כמעט 0 .

- **כיצד נהפוך את** SVD **ליציב נומרית:**



2.2.2 $Noisy\ case$:

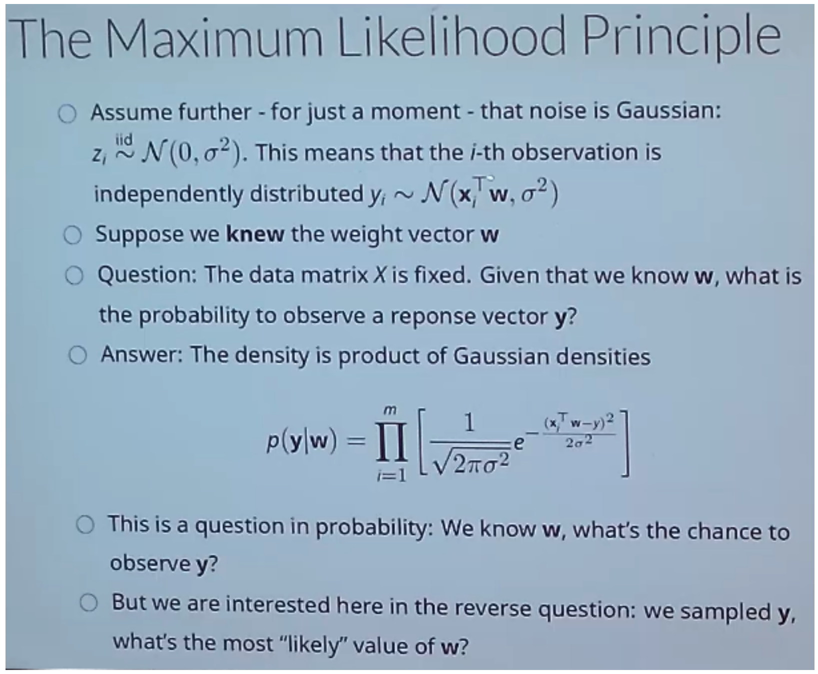
- **במקרה של רעש:** נשתמש במודל הסתברותי.

- **הנחות:** נאמר שהחריגה מהמודל הלינארי היא אולי מקרית ולכן:
לא נניח יותר שהלייבלים שאנו צופים בהם הם $(\mathbf{x}_i, f(\mathbf{x}_i))$. אלא נניח ש $(\mathbf{x}_i, f(\mathbf{x}_i) + z_i)$ עבור $z_1, \ldots z_m \overset{\text{lid}}{\sim} (0, \sigma^2)$.
כלומר - קיים ווקטור Z שהוא ווקטור הרעש ונצטרך "לנקות" את הרעש ע"י הטלה של הווקטור Y.

- **נלמד באופן הבא:** $y_i = x^{\top}w + z_i$, את ההטלה נעשה באופן הבא: על נטיל הווקטור $y = Xw + z$.

- **נגדיר את פונקציית ה** $Loss$: $\mathbf{w}_s := argmin\|y - X\mathbf{w}\|^2$.

- **עקרון ה MLE – Maximum Likelihood :**



- $L(\mathbf{w} \mid \mathbf{y}) = \frac{1}{(2\pi\sigma^2)^{m/2}} \prod_{i=1}^{m} \left[e^{-\frac{\left(\mathbf{x}_i^\top \mathbf{w} - \boldsymbol{y}\right)^2}{2\sigma^2}} \right]$: likelihood function

- **מתקיים:**

$$\hat{\mathbf{w}} := argmaxL(\mathbf{w} \mid \mathbf{y}) = argmax \log L(\mathbf{w} \mid \mathbf{y}) = argmin \sum_{i=1}^{m} \left(\mathbf{x}_i^\top \mathbf{w} - y_i\right)^2$$

כלומר: ניתן לחשב מקסימום על לוג באותה המידה.

3 שבוע 3:

3.1 תרגול 3:

- **רגרסיה לינארית:** נניח על הבעיה שהיא לינארית, ונחפש את הפתרון בתוך קבוצות הפונקציות הלינאריות.
רגרסיה: אנו מנסים למצוא קשר בין קבוצת המשתנים $X \in R^d$ לבין קבוצת הרספונסס $Y \in R$. כלומר - למצוא את הפונקציה f שעבורה $f(x) = y$. וככל שנדגום יותר התוצאה תהיה מדוייקת יותר.

- הנחה: אנו מניחים ש f היא לינארית ודטרמיניסטית (אין לה מימד הסתברותי).

- **סט האימון** S**:** נגדיר את סט הדוגמאות שלנו להיות $S = \{(x_i, y_i)\}_{i=1}^m$ כאשר הזוגות הם זוגות של ווקטור דוגמאות ורספונס.

- **נחפש את** $\hat{f}$**:** שמנאבת לנו את התצאות על סמך הדגימות.

- **הגדרה - הטסט** $\setminus$ $Predicrtion\ rule$**:** דוגמה חדשה x_{m+1} שעדיין לא ראינו ואנו צריכים לבדוק את $\hat{f}$ עליה.

- **מחלקת היפוטזה:** נניח הנחה על הפונקציה f (לדוגמה - f לינארית) ונחפש את $\hat{f}$ בתוך משפחת הפונקציות הללו.

$$H_{\text{reg}} = \left\{ h([x_0...x_d]) = w_0 + \sum_{i=0}^{d} w_i x_i \right\} \quad w_i \in R$$

למעשה זאת משפחת פונקציות אפיניות.

w_0 נקרא - ההזזה $intercept$. ו $w_1...w_d$ נקראים המשקולות (המשקל שאנו נותנים לכל פיצ'ר בווקטור). הווקטור W נקרא ווקטור המשקולות.

- **הערה:** אם נרצה לדייק את הפונקציה שלנו "יותר מידיי" אנו נתייחס יותר מידיי לרעש ממנו אנו צריכים להתעלם.

- **כיצד נהפוך את הפונקציה האפינית לפונקציה לינארית:** נוסיף לווקטור X קורדינטה התחלתית $x_0 = 1$ כך מתקיים $|X| = d+1$.

  בנוסף נכניס את w_0 לתוך הווקטור W. כך נקבל שני וקטורים (X, W) ממימד $d+1$.

  כעת מחלקת ההיפוטזות היא:

$$\mathcal{H}_{\text{lin}} = \left\{ f(\mathbf{x}) = \mathbf{x}^\top \mathbf{w} \mid \mathbf{w} \in \mathbb{R}^{d+1} \right\}$$

  ואנו מחפשים W שיפתור לנו את מערכת המשוואות.

- **ייצוג באמצעות מטריצה:**

$$x_{m \times d+1} = \begin{bmatrix} -x_1^T- \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ -x_m^T- \end{bmatrix}, Y = \begin{bmatrix} y_1 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ y_m \end{bmatrix}$$

  ואנו מחפשים W כך ש $XW = Y$.

- **הנחה:** נניח ש $m \geq d+1$ כלומר - יש יותר דוגמאות מפיצ'רים (יש מספיק מידע כדי לפתור את הבעיה).

- **הנחת הרילזביליות:** אנו מניחים כי $h_S \in H_{reg}$. כלומר - h_S שייכת למחלקת ההיפותזות שאנו מחפשים בה, וקיימת h_S כזו, ומתקיים $Y \in Im(X)$ כלומר Y בתוך תת המרחב שנפרש ע"י X.

- **הנחת האי רלזביליות:** אין פתרון יחיד למערכת המשוואות, לכן נרצה את הW הטוב ביותר שיחזיר לנו $Loss$ מינימלי. כלומר $Y \notin Im(X)$ כלומר Y לא נמצא בתוך תת המרחב שנפרש ע"י X.

- **פונקציית** RSS**:** $RSS_{x,y}(w) = ||Xw - y||^2$, אנו מחפשים ווקטור W שיביא לנו את ה $Loss$ הנמוך ביותר, עקרון ERM.

  ניתן לבחור כל פונקציית $Loss$, אנו בחרנו את RSS.

- **טענה למקרה הלא סינגולרי:** עבור מטריצה $X \in R^{m\times d+1}$ ומתקיים $m \geq d+1$ אם $dim(Ker(X)) = 0$ אזי $\hat{W} = \left(X^\top X\right)^{-1} X^T Y$ הוא הפתרון האופטימלי.

- **הגדרה ־ $dager$ " + ":** X^+ מסמן את הפסואודו אינברס (פירוק סינגולרי) של X.

- **במקרה הסינגולרי ־ פסואודו אינברס:** עבור מטריצה $X \in R^{m\times d+1}$ ו $U\Sigma V^\top$ פירוק SVD של X אזי $X^+ = V\Sigma^+ U^T$ עבור $\Sigma^+_{i,i} = \begin{cases} 1/\sigma_{ii} & \sigma_{ii} \neq 0 \\ 0 & \sigma_{ii} = 0 \end{cases}$

- **טענה במקרה הסינגולרי:** עבור מטריצה $X \in R^{m\times d+1}$ ומתקיים $m \geq d+1$ אם $dim(Ker(X)) = 0$ אזי $\hat{W} = X^+ Y$ הוא הפתרון האופטימלי.

- **הפתרון לבעית הריגרסיה:** $\hat{W} = X^+ Y$ הוא הפתרון תמיד גם מקרה הסינגולרי וגם בלא סינגולרי.

- **הערה:** כאשר צריכים למדל אובייקטים שאין בניהם יחס סדר, נתן לכל אחד מהאובייקטים פיצ'ר משלו שיקבל ערך של $1\backslash 0$ כך שסכום כל האובייקטים =1.

3.2 הרצאה 3 ־ סיווג:

- **הגדרה ־ בעיית קלסיפיקציה:** בשונה מבעיית רגרסיה ששם $Y \in R$, בבעיות קלסיפיקציה הקבוצה בדידה ומתקיים $\mathcal{Y} = \{1, \ldots, k\}$. אנו נתמקד בבעיות קלסיפיקציה בינאריות.

- **כיצד נבדוק שסיווגנו נכון:** נבדוק את מספר הטעויות שנעשו, כך ־

$$L_s(h) := \sum_{i=1}^{m} \mathbf{1}_{y_i \neq h(x_i)} = |\{i \mid y_i \neq h(x_i)\}|$$

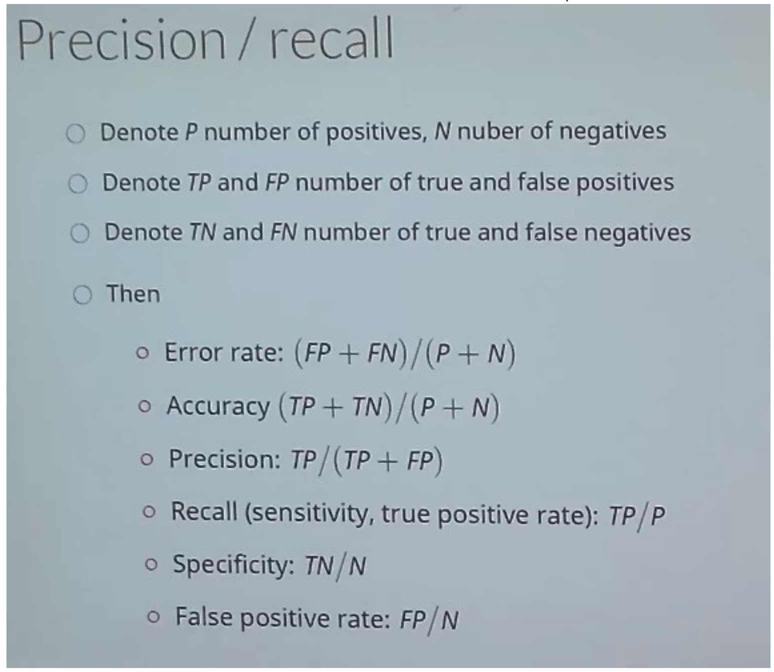
ונשאף לקחת את המסווג שהביא לנו את טווח הטעות הנמוך ביותר.

- **הגדרת המסווג:** נגדיר $positive$ ו $negative$ כדי שנדע באופן ברור כיצד להתייחס ל $response$. בנוסף נגדיר $false\ positive$־ האם החזרנו חיובי וטעינו. ו $false\ negative$ ־ האם החזרנו שלילי וטעינו.

- **טעות מסדר ראשון:** תמיד נגדיר את האופציה הגרועה מבחינתו להיות הטעות מסדר ראשון. לדוגמא ־ המכם לא יחזיר אמת על מטוס אוייב ־ $false\ negative$.

- **טעות מסדר שני:** האופציה השניה לטעות תהיהטעות מסדר שני. לדוגמה ־ המכם החזיר אמת על ציפור ־ $false\ positive$.

- **הגדרה - $Precision\backslash Recall$:**



- **כלל החלטה:** פונקציה מהמרחב שלנו ל $\{-1,1\}$, עבור כל דגימה נקבל סיווג.

- **הגדרה - גבול של כלל החלטה:** גבול שמחלק בין הדגימות שהחזירו 1- לבין הדגימות שהחזירו 1.

- **שאלות שנשאל כדי להבין מסווג חדש:**
  1: מה מחלקת ההיפותזות (מה קבוצת כל כללי ההחלטה מהם נבחר כלל).
  2: איך נראה הגבול של כלל ההחלטה.
  3: מה העיקרון שלפיו נבחר את כלל ההחלטה מתוך H ($EMR, likelihood$).
  4: איך מיישמים את העיקרון מבחינה חישובית.
  5: לאחר האימון - איך נשמור את המודל המאומן, באיזה מבנה נתונים נשתמש.
  6: איך נחשב חיזוי לנקודה חדשה.
  7: האם המודל בר פירוש.
  8: האם אלגוריתם הלמידה נותן בנוסף לתחזית של $\{0,1\}$ גם הסתברות לכך שהוא צדק.
  9: האם זה מודל בודד או משפחת מודלים.
  10: מהן הבעיות אותן נפתור עם המודל.

3.2.1 $HALF-SPACE\ CLASSIFIER$:

- **אלגוריתם $HALF-SPACE\ CLASSIFIER$:**

1 מחלקת ההיפותזות: פונקציות שנותנות לחצי מישור "+" ולחצי השני "-" $x\mapsto sign(\langle x,w\rangle+b)$. מחלקת ההיפותזות היא :

$$H_{\text{half}}=\left\{h_{\mathrm{w}}\mid \mathbf{w}\in\mathbb{R}^d\right\}$$

כלומר ־ תוצאת המכפלה הפנימית היא כלל ההחלטה, אם קיבלנו תוצאה חיובית אנחנו בצד החיובי של העל מישור, אחרת אנו בצד השני, (עבור ווקטור W קבוע מגדיר את על המישור ־ המשלים האורתוגונלי שלו הוא על המישור).

2 איך נבחר את W**:** נתייג את הטעות שלנו באופן הבא, אם החזרנו מ"פ חיובית עבור סיווג שלילי או מ"פ שלילית עבור סיווג חיובי אזי נאמר ש $y_i\langle \mathbf{w},\mathbf{x}\rangle \leq 0$. לכן נספור את הטעויות כך:

$$L_s\left(h_{\mathrm{w}}\right) = \#\left\{i \mid y_i\langle \mathbf{w},\mathbf{x}\rangle \leq 0\right\}$$

ונבחר את המינימלי ביותר ־ ERM.

3 האלגוריתם: נמצא את W בעזרת אלגוריתם תכנון לינארי. (נקראת בעיית פיזביליות ־ בעיה שצריך להחזיר וקטור אחד שפותר את מערכת המשוואות)

4 איך ניתן תחזית: ע"י מכפלה פנימית.

5: איך נאחסן: נאחסן את W.

הערה: יישום ERM במודל זה היא בעיית $np-hard$.

- **הגדרה ־ בעיית אופטימיזציה קמורה:** בעיית אופטימיזציה ־ נחפש את המינימום של f_0 עבור $f_i(x) \leq b_i$ לכל $i \in [n]$.
  אם כל הפונקציות $f_0...f_n$ הן קמורות אזי הבעיה היא **בעיית אופטימיזציה קמורה**.
  ואם הפונקציות הן לינאריות הבעיה נקראת בעיית **תכנון לינארי**.

3.2.2 $SUPPORT\ VECTOR\ MACHINES-SVM$:

- **אלגוריתם** SVM**:** נמשיך להשתמש באותו על מישור, מחלקת ההיפותזות היא אותה מחקה אך עיקרון הלמידה שונה. אנו מתשמשים בהנחה שהדאטה מופרד לינארית.

- **נגדיר מרחק:** על המישור שמוגדר ע"י הווקטור W ועבור נקודות בוחן x, נגדיר את המרחק בניהם כך:

$$d(\mathbf{x}, L) = \min\{\|\mathbf{x} - \mathbf{v}\| : \mathbf{v} \in L\}$$

- **טענה:** אם W מנורמל ($||W|| = 1$) אזי $d(\mathbf{x}, L) = |\langle \mathbf{w},\mathbf{x}\rangle|$.

- **נגדיר את השול ־** $Margin$**:** השול מוגדר להיות $\min_i |\langle w, x_i\rangle|$ עבור $i \in [n]$.
  הווקטורים שמשיגים את המרחק המינימלי נקראים support vectors.

- **עיקרון הלמידה** $Max\ Margin$**:** אנו רוצים למקסם את השול. אנו נסתכל רק על ווקטורים מנורמלים, ולכל אחד מהם נסתכל על המינימום של הערך המוחלט של מ"פ

$$\underset{w:\|w\|=1}{argmax} \min_{i\in[m]} |\langle w, \mathbf{x}_i\rangle| \text{ s.t. } \forall i, y_i \left\langle \mathbf{w}, \mathbf{x}_i\right\rangle > 0$$

ומתקיים שוויון לנוסחה הבאה:

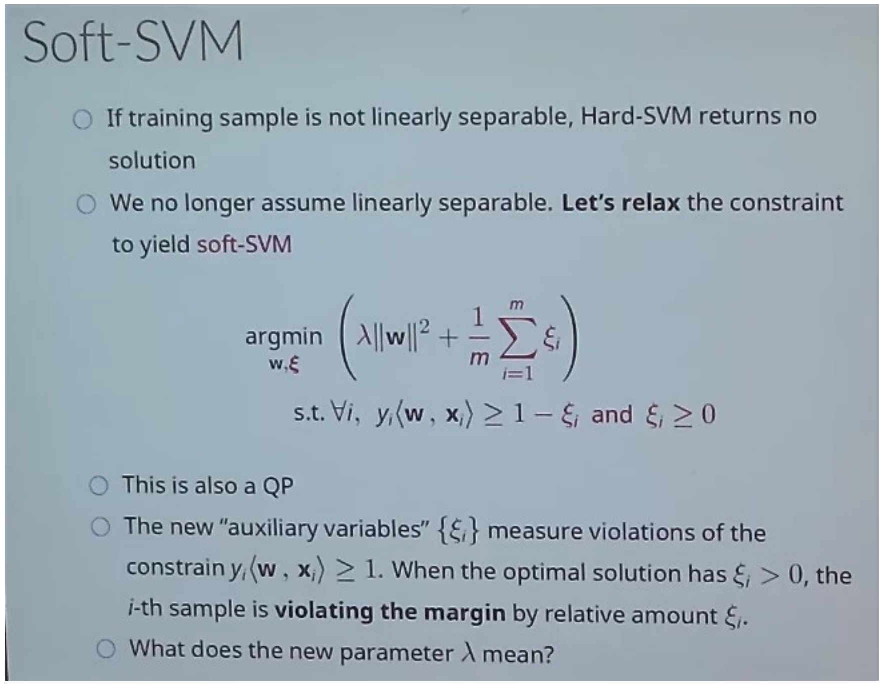
$$\underset{w}{argmin}\|\mathbf{w}\|^2 \text{ s.t. } \forall i, y_i \left(\mathbf{w}, \mathbf{x}_i\right\rangle \geq 1$$

נסתכל רק על W שמפרידים את הדאטה ל 2, ומתוכם נבחר את האחד הם הנורמה הקטנה ביותר .

עיקרון הלמידה: נבחר את W שעבורו המרחק הוא מקסימלי.

- **משפחת אלגוריתמי למידה** $Soft\ SVM$: כעת לא נשתמש בהנחה שהדאטה מופרד לינארית משום שאין דאטה כזה. נאפשר הפרות ־ נקודות מסויימות יוכלו לעבור לצד השני של העל מישור, נעשה זאת כך:

$$\underset{\mathbf{w},\xi}{argmin}\left(\lambda\|\mathbf{w}\|^2+\frac{1}{m}\sum_{i=1}^{m}\xi_i\right)$$
$$\text{s.t. } \forall i, y_i\left\langle\mathbf{w},\mathbf{x}_i\right\rangle\geq 1-\xi_i \text{ and } \xi_i\geq 0$$



כאשר ξ_i מייצגים את ההפרות ואם $\xi_i=0$ אין הפרות, ופרמטר רגולריזציה λ מייצג את מספר ההפרות שאנו מוכנים להכיל.

אנחנו חייבים לאפשר הפרות כי ההנחה שהמידע מופרד לינארי לא תקפה יותר.

- **תכונות:** אם λ קטן אזי אנחנו זזים אחרי כל הפרה וצמדים אליה, לכן השונות שלנו תהיה יותר גדולה.

3.2.3 אלגוריתם $LOGISTIC\ REGRESSION$:

- **מוטיבציה:** אנו נרצה לדבר על ההסתברות להיות שייכים ל $class$ מסויים.

- **הרעיון:** בדומה למודל רגרסיה לינארית עם רעש גאוסי שמקיים ־

$$y_i=\left\langle\mathbf{x_i},\mathbf{w}\right\rangle+z_i \text{ so } y_i\sim\mathcal{N}\left(\left\langle\mathbf{x_i},\mathbf{w}\right\rangle,\sigma^2\right)$$

אצלנו: מכיוון שאנחנו בבעיות קלסיפיקציה נניח ש $y_i\sim Ber\left(p_i\right)$ והדגימות ב"ת.

- **הפונקציה** $linkfunction$: פונקציה מונוטונית עולה שממפה את הישר הממשי לקטע $[0,1]$, ככל שהמ"פ גדולה יותר p - גדול יותר. (זהו קשר בין מ"פ לפרמטר ברנולי)

$$linkfunction \phi : \mathbb{R} \to (0,1)$$

- **הנחה:** $p_i = \phi\left(\langle \mathbf{x_i}, \mathbf{w} \rangle\right)$ ולכן $y_i \sim Ber\left(\phi\left(\langle \mathbf{x_i}, \mathbf{w} \rangle\right)\right)$
בנוסף באותו האופן של רגרסיה לינארית:

$$\mathbf{x} = (1, x_1, \ldots, x_d) \text{ and } \mathbf{w} = (w_0, w_1, \ldots, w_d)$$

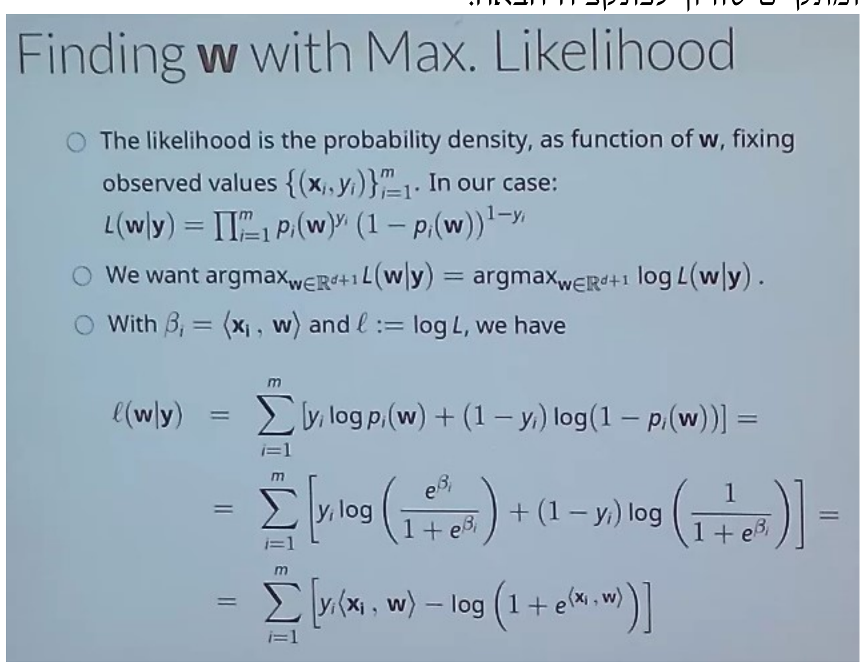
- **הפונקציה הלוגיסטית:** יכולה לשמש כ $linkfunction$ - $\pi(x) = \frac{e^x}{1+e^x}$
- **מחלקת ההיפותזות:** ממפה את X לפונקציה הלוגיסטית של מ"פ

$$\mathcal{H}^d_{\text{logistic}} = \{\mathbf{x} \mapsto \pi(\langle \mathbf{x}, \mathbf{w} \rangle)\}$$

- **עיקרון הלמידה:** יש לנו מודל הסתברותי ולכן נשתמש ב $max\ likelihood$ עבור הפוקציה

$$Prob(Y = \mathbf{y} \mid \mathbf{w}) = \prod_{i=1}^{m} p_i(\mathbf{w})^{y_i} \left(1 - p_i(\mathbf{w})\right)^{(1-y_i)}$$

כאשר $p_i = \pi\left(\langle \mathbf{x_i}, \mathbf{w} \rangle\right) = \frac{\exp(\langle \mathbf{x_i}, \mathbf{w} \rangle)}{1+\exp(\langle \mathbf{x_i}, \mathbf{w} \rangle)}$.
ומתקיים שוויון לפונקציה הבאה:



הפונקציה היא פונקציה קמורה ולכן בעיית אופטימיזציה קמורה.

- **הגדרה - אלגוריתם** $Interpretability$**:** $Interpretability$ היא תכונה של אלגוריתם לומד שניתן לפירוש והוא מאפשר לענות על השאלות הבאות - אילו פיצ'רים היו שימושיים ואלו לא קשורים. בנוסף ניתן לתחקר את פעולת האלגוריתם במקרה של טעות.

- **טענה:** האלגוריתם של $LOGISTIC\ REGRESSION$ הוא אלגוריתם $Interpretability$.

- **שלב אחרון:** אנו צריכים לבחור כיצד לסווג את התוצאות שחזרו מהפונקציה, מכיוון שהפונקציה מחזירה מספר בקטע $[0,1]$ ולא מספר קבוע. לכן נבחר ערך α כך ש:

$$\hat{y} := \begin{cases} 1 & h(x) > \alpha \\ 0 & h(x) \leq \alpha \end{cases}$$

והפונקציה הזאת תהיה הפונקציה הסופית שתסווג את הדגימות.

למעשה אנו מחפשים α כך שנקבל $true\ positive$ טוב ו $false\ positive$ נמוך ככל הניתן.

3.2.4 $K-NEARESTNEIGHBORS\ (K-NN)$:

- **הרעיון:** אין מודל ואין אימון. עבור כל נקודה חדשה נבחר את K הנקודות בסט האימון שהכי קרובות לנקודה החדשה (קרובות לפי איזו נורמה שבא לנו) ונלך לפי הרוב.

- **כיצד נבחר את הפרמטר** K**:** נהחר את ה K שמחזיר לנו את טעות הסיווג הקטנה ביותר.

- **המימוש:** נעבד את הדאטה ונשמור אותו במבנה נתונים מסויים שמחלק את המרחב האוקידי לכמה מקטעים וכך כשנקבל דגימה חדשה נדע מי השכנים הקרובים מבלי לחשב שוב.

  שיטה נוספת: נחפש את השכנים הקרובין במימדים כטנים יותר ונעשה ממוצע בניהם.

- **טענה:** שיטה זאת אינה $Interpretability$.

- **מתי נשתמש:** אם אין לנו רעיונות אחרים, רעיון זה צריך לעבוד תמיד.

3.2.5 $CLASSIFICATION\ TREES$

- **הרעיון:** עצי החלטה שמסווגים את כל המקרים.

- **חלוקת עצים -** $Tree\ Partition$**:** חילוק המרחב האוקלידי לאיחוד של קבוצות זרות. את החלוקה ניתן לייצג בעזרת עץ בינארי.

  אנו נתייחס רק לקבוצות שמחלקות לאורך אחד מהצירים (הקווים מקבילים לראשית).

- **מחלקת ההיפותזות:** עבור חלוקה $\mathbb{R}^d = \biguplus_{i=1}^{N} B_j$ נתאים לכל קבוצה לייבל $c_j \in \{0,1\}$. מחלקת ההיפותזות תכיל פונקציות קבועות למקוטעין מהצורה:

$$h(\mathbf{x}) = \sum_{j=1}^{N} C_j \mathbf{1}_{B_j}(\mathbf{x})$$

ונגדיר את המחלקה: $\mathcal{H}_{CT}^{k}$ עבור מחלקת היפותזות שמחלקת את המרחב ל k קבוצות (יש לעץ לכל היותר k רמות).

- **איך נבחר עץ - ERM:** נבחר מבין כל מחלקת ההיפותזות את העץ המתאים ביותר.
החסרון: $over\ fitt$, לכן צריך להגביל את **עומק העץ** וגם את **מספר הנקודות** בכל קבוצה.

- **כיצד נמצא את המינימלי:** נשתמש בהיוריסטיקה $CART$ (כי מציאת המינימום היא בעיית $NP-hard$).
כיצד נגדל עץ: עבור d צירים ו m נקודות יש לנו $d \cdot m$ אופציות לחיתוך. נתחיל מקופסה אחת ונחלק כל קופסה ל 2 קופסאות, נעצור כשנגיע ל K רמות או למספר מינימלי של נקודות בכל קופסה (חיפוש חמדן).
עיקרון הלמידה: מתוך כל החיתוכים נבחר את החיתוך שמביא לנו ERM מינימלי, ואת הסימן של כל קבוצה נבחר על פי הרוב.

- **איך נעשה פרדיקציה:** עבור נקודה חדשה נעבור על העץ ונחליט.

- **איך נאחסן:** נשמור את פיצולי העץ ואת התוצאות בעלים.

- **טענה:** ממוצע על מספר עצים הוא מסווג מאד חזק.

- **טענה:** שיטה זאת היא $Interpretability$.

4 שבוע 4:

4.1 תרגול 4:

- **סט הדגימות S במקרה של רעש ε:**

$$S=\left\{\left(x_{i,}f\left(x_i\right)+\varepsilon\right)\right\}_{i=1}^{m}$$

Y לא נמצא בתמונה של X, אך עדיין אנחנו רוצים למצוא את ה $Loss$ המינימלי וכך נקבל את התוצאה הטובה ביותר.

- **שיטה נוספת - $Max\ likelihood$:** נניח שכל ε נדגם מהתפלגות גאוסיאן סביב 0 - $\varepsilon_i \overset{i.i.d}{\sim} N\left(0,\sigma^2\right)$.
ועבור y_i: $y_i \overset{i.i.d}{\sim} N\left(x^\top w,\sigma^2\right)$. מתפלג סביב $x^\top w$.
ועבור הווקטור Y: $y\sim N\left(Xw,I_m\sigma^2\right)$.
למעשה - אנו מחפשים את ה w הסביר ביותר בהינתן ווקטור Y. ונמצא אותו באמצעות חישוב ההסתברות של $P(Y\mid w)$.

- $liklehood$:

$$\frac{1}{\left(2\pi\sigma^2\right)^{m/2}}\exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{m}\left(\mathbf{x}_i^\top\mathbf{w}-y_i\right)^2\right)$$

ואנו נחפש את ה w שממקסם את הפונקציה הזו. או באופן דומה את המינימום הבא:

$$=\underset{\mathbf{w}\in\mathbb{R}^{d+1}}{argmin}\|\mathbf{Xw}-\mathbf{y}\|^2$$

למעשה זאת הפונקיה של ERM.

4.1.1 $Polyfit$:

- אנו נשאל האם מחלקת ההיפותזות הבאה יכולה להתאים לעוד בעיות חוץ מרגרסיה לינארית.

$$H_{reg} = \left\{ x \to x^T w : w \in R^{d+1} \right\}$$

- **שימוש לפולינומים:** נגדיר כל חזקה של x כפונקציה לינארית $h_i : R \Rightarrow R$, ונהפוך את הפולינום לווקטור של מקדמים.
כך למעשה נעביר בעיה פולינומית ללינארית.
נייצג את הפולינום באופן הבא:

$$P(x) = \Sigma_{i=0}^{n} a_i x^i = \Sigma a_i \cdot h_i(x)$$

עבור $h_i(x) = x^i$. פונקציה שמחזירה סקלר, ונחפש את המקדמים a_i.
למעשה נעביר כל דגימה לווקטור $h(x) = [h_0(x)....h_n(x)]$.

- **הפתרון:** אם כל ה x_i שונים, אזי נאמר שהמטריצה היא מדרגה מלאה, והיא הפיכה ולכן נפתור לפי המקרה הלא סינגולרי ונקבל את המקדמים של הפולינום.

4.1.2 $Bias\ \&\ Variance$:

- **אינטואיציה:** מכיוון ש Y הוא רנדומלי, אנו יכולים לדבר על w בתור אומד $\hat{\Theta}$, ונחפש מי המעריך הטוב ביותר שלנו ונוכל לדבר על התכונות שלו.
עבור Θ, נוכל לדבר על האומד שלו $\hat{\Theta}$.

- $Bias$: מייצג את המרחק מהדבר האמיתי, המרחק בין מה שחזינו לבין התוצאה האמיתית.

- Var: מייצג את השונות שלי מעצמי, ההפרשים בין התשובות שלנו בהתאם לדגימות. אם נרדוף אחרי הרעש ה var יהיה גבוה כי תשתנה בכל איטרציה.

- הקשר ל MSE:

$$MSE = \underset{\mathbf{w} \in \mathbb{R}^{d+1}}{argmin} \|\mathbf{Xw} - \mathbf{y}\|^2 = E\left[(\hat{y} - y)^2\right] = Var + Bias^2$$

עבור $\hat{y}$ - הניחוש שלנו בהתאם לדגימות.

4.1.3 מדד R^2:

- **מדד שאומר לנו כמה אנחנו טובים ביחס למודל:**

$$R^2 = 1 - \frac{SSE}{SST}$$

עבור: $SSE = \sum (y_i - \hat{y}_i)^2$ ו $SST = \sum (y_i - E(\hat{y}))^2$
מתקיים $R^2 \in [0,1]$, והמקרה הטוב ביותר הוא כש R^2 קרוב ל 1.

4.2 הרצאה 4 - תיאוריה של למידה חישובית:

4.2.1 הגדרות:

- **נרצה לענות על השאלות הבאות:** מה ניתן ללמוד, איך לומדים, האם יש מינומום דגימות שאפשר ללמוד להן.
- **הנחה:** נניח כי הדגימות מהדומיין X מתפלגות באופן שווה וב"ת מעל הסתברות D, נשתמש באותה הנחה גם לדגימות טסט שנקבל בעתיד.
- **מדידת ביצועים, נחשב את** $Loss$:

$$L_{\mathcal{D},f}(h) \overset{\text{def}}{=} \underset{x\sim\mathcal{D}}{\mathbb{P}}[h(x)\neq f(x)] \overset{\text{def}}{=} \mathcal{D}(\{x\in\mathcal{X}: h(x)\neq f(x)\})$$

- **מה אנו מחפשים:** אלגוריתם A שיוציא לנו כלל החלטה h עם $Loss$ מינימלי.
- **הגדרה -** $PAC-learnability$:

מחלקת היפותזות H נקראת $PAC-learnability$ אם קיימת פונקציה $\tilde{m}_{\mathcal{H}}:(0,1)^2\to\mathbb{N}$ ואלגוריתם למידה A כך ש לכל $\epsilon,\delta\in(0,1)$ ולכל התפלגות D על X, ולכל פונקציה $f:\mathcal{X}\to\{\pm1\}$ שקיימת עבורה $h^*\in\mathcal{H}$ כך ש $L_{\mathcal{D}f}(h^*)=0$.

כך שאם נריץ את A על $m\geq\tilde{m}_{\mathcal{H}}(\epsilon,\delta)$ דגימות (עבור $\tilde{m}_{\mathcal{H}}(\epsilon,\delta)$ דגימות ב"ת שוות התפלגות שנדגמו לפי D ותוייגו לפי f) האלגוריתם יחזיר $h_S=\mathcal{A}(S)$ כך שבהסתברות של לפחות $1-\delta$ (מעל הבחירה של סט האימון) יתקיים עבור ה $Loss$ - $L_{\mathcal{D},f}(h_S)\leq\epsilon$.

- **אינטואיציה:** אנו מנסים למצוא מחלקות היפותזה לא מסובכות שנתן ללמוד אותן למידת PAC.
- **הגדרה - סיבוכיות המדגם של מחלקת היפותזות:** עבור מחלקת היפותזות שהיא למידה PAC, נגדיר את סיבוכיות המדגם עבור ε,δ נתונים. בתור המספר המינימלי של דגימות $\tilde{m}_{\mathcal{H}}(\epsilon,\delta)$ שעבורו ההגדרה הקודמת מתקיימת.
- **הגדרה - מימד** VC: תהי מח' היפותזות $\mathcal{H}\subset\{\pm1\}^{\mathcal{X}}$. עבור תת קבוצה $C\subset\mathcal{X}$ נגדיר H_C להיות הצמצום של C:

$\mathcal{H}_C=\{h_C: h\in\mathcal{H}\}$

עבור h_C פונקציה מצומצמת למרחב C.

נגדיר את המימד VC להיות:

$$VCdim(\mathcal{H}) := \max\left\{|C| \mid C\subset\mathcal{X} \text{ and } |\mathcal{H}_C|=2^{|c|}\right\}$$

- **המשפט היסודי של הלמידה הסטטיסטית:** מחלקת היפותזות H היא למידת PAC אמ"מ מימד ה VC שלה קטן מ ∞.

סיבוכיות המדגם (מס' דוגמאות המינימלי האפשרי) של מח' היפותזות עם מימד VC **סופי היא:**

$$m_H(\epsilon,\delta)\sim\frac{VCdim(\mathcal{H})+\log(1/\delta)}{\epsilon}$$

טענה: עיקרון ERM (מינימום לוס) משיג את המינימום הזה.

4.2.2 תורת PAC:

- **נסתכל על הבעיה כעל משחק בנינו לבין הטבע:** אנו בוחרים את A, והטבע בוחר את D, f.

- **גרסה ראשונה:** נקבע את m ונבחר את $\mathcal{A} : (\mathcal{X}, \mathcal{Y})^m \to \mathcal{Y}^{\mathcal{X}}$, הטבע בתגובה יודע את A ומחזיר לנו את התוצאה הקשה ביותר עבורנו ובוחרת את D, f. בשלב התשלום השופט דוגם m דגימות מתוך X שמתפלגות שאופן שווה על D, מסמן אותם ב f ונותן אותם ל A. אנו צריכי להחזיר כלל החלטה.
נשים לב כי פונקציית $Loss$ היא בעלת מימד הסתברותי גם כן.

- **הגדרה -** $Probably\ correct\ learner$:
טענה: לא נוכל לבחור A כזה כך שבהסתברות 1 הטעות שלנו $(Loss)$ תהיה קטנה מ ε עבור כל D, f.
הפתרון: נרשה ל A להיכשל לחלוטין בהסתברות נמוכה מאד - לכל היותר δ (קבוע).

- **הגדרה -**$Approximately\ correct\ learner$:
טענה: לא ניתן לקבל $L_{\mathcal{D},f}\left(h_S\right) = 0$ על המאורע המשלים, בהסתברות של $1-\delta$ (מכיוון שיכולות להיות נקודות חריגות שלא ראינו בשלב האימון).
הפתרון: כאשר ה $L_{\mathcal{D},f}\left(h_S\right)$ של A חסום ע"י ε, בהסתברות של $1-\delta$. נאמר שהוא $Approximately\ correct$ עם ε.

- **הגדרה -** $ProbablyApproximatelyCorrect$: נאמר שאלגוריתם הוא לומד PAC אם לכל D, f מתקיים:

$$\mathbb{P}_{\mathcal{D}^m}\left\{S \in (\mathcal{X} \times \mathcal{Y})^m \mid L_{\mathcal{D}f}\left(h_s\right) \leq \epsilon\right\} > 1-\delta$$

- **גרסה שניה:** הגודל של S לא יהיה קבוע מראש, אך ε, δ יהיו קבועים מראש. ואח"כ אנו נבחר את m - מספר הדגימות שאנו רוצים, ואת האלגוריתם A. כך אנו יכולים להחליט מהו טווח הטעות שאנו שואפים אליה.
לאחר מכן נריץ כמה פעמים, ואם קיבלנו $\left\{S \sim \mathcal{D}^m \mid L_{\mathcal{D},f}\left(h_s\right) \leq \epsilon\right\}$ אזי הצלחנו במשימה.

- **משפט -** $No\ Free\ Lunch$: אם אין הנחות על f, אזי אי אפשר לנצח בגרסה השניה של המשחק.
כדי לנצח: אנו צריכים להניח מאיזו משפחה f מגיעה, לכן צריך מחלקת היפותזות H. נניח את הנחת הריליזביליות - $x = f(x)$ ונניח $f \in H$ וגם כלל ההחלטה שייך ל H.
טענה: אם מרחב המדגם הינו אינסופי אזי מחלקת ההיפותזות גדולה מידיי כך שאי אפשר ללמוד ממנה (אם הפונקציות לא מסובכות אזי אפשר ללמוד מהן).

- **נשאל את השאלות הבאות:**
מה הגודל של H שניתן ללמוד ממנו?
מהן הקבוצות H שגדולות מידיי שא"א ללמוד מהן?
מהו מספר הדוגמאות המינימלי?
האם יש קשר בין הגודל של מח' בהיפותזות למספר הדוגמאות המינימלי?
האם בהינתן H ניתן למצוא אלגוריתם A יעיל?
האם ניתן למצוא את הלרנר שמשיג את מספר הדוגמאות המינימלי?

- **גרסה שלישית:** נקבע מיהם ε, δ ואת מחלקת ההיפותזות H, ואלגוריתם A שממפה מתוך H. הטבע תבחר התפלגות D ופונקציה $f \in H$.

- **טענה:** אם H סופית, אזי היא למידה PAC.

- **הגדרה - ERM:** עבור מדגם Sוכלל החלטה h- $L_s(h) = \frac{1}{m} |\{i : h(x_i) \neq y_i\}|$.
אנו נבחר את המינמלי, וכך נוכל לפתור כל בעיה עם מח' היפותזות סופית.

4.2.3 מימד VC:

- **אינטואיציה למימד VC:** דרך למדוד מהי רמת המורכבות של מחלקה H, בכדי שנוכל לדעת האם היא מחלקה שניתנת ללימוד PAC או לא.

- **הגדרה - קבוצה מנתצת:** נאמר שקבוצה H היא מנתצת תת קבוצה C, אם **כל הפונקציות** האפשריות מתקבלות כצמצום של H לשם.

- **משפט:** הגודל המקסימלי של תת קבוצה C נותן לנו חסם תחתון על H, ואם הגודל המקסימלי הוא אינסוף (תמיד ניתן למצוא C גדולה יותר שמקיימת את התכונות) אזי לא ניתן ללמוד PAC מהמשפחה H.

- **המשפט הפונדמנטלי:** אם מימד VC של H הוא סופי, אזי ניתן ללמוד PAC.
ומספר הדגימות המינימלי הוא:

$$m_H(\epsilon, \delta) \sim \frac{Vdim(\mathcal{H}) + \log(1/\delta)}{\epsilon}$$

5 שבוע 5:

5.1 תרגול 5 - סיווג:

- **מה אנחנו מחפשים:** אנו רוצים לחלק את העולם ל $classess$, עדיין מעל R^d אבל המרחב שלנו בדיד $\{-1, 1\}$.
הדגימות: x_1, x_2הם נקודות מעל R^2 וערכי ה Y הן מעל $\{-1, 1\}$.

- **הגדרה - על מישור $hyperplane$:** עבור $\mathbf{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$

$$\left\{\mathbf{x} \mid \langle \mathbf{w}, \mathbf{x} \rangle + b = 0, \mathbf{x} \in \mathbb{R}^d\right\}$$

למעשה זה מגדיר קו ישר ב R^2.

- **הגדרה - חצי מרחב $half\ space$:** יהיו (w, b) על מישור, אזי חצי המרחב שלהם מוגדר להיות:

$$\left\{\mathbf{x} \mid \langle \mathbf{w}, \mathbf{x} \rangle + b \geq 0, \mathbf{x} \in \mathbb{R}^d\right\} \iff \left\{\mathbf{x} \mid sign(\langle \mathbf{w}, \mathbf{x} \rangle + b) \geq 0, \mathbf{x} \in \mathbb{R}^d\right\}$$

למעשה זה חצי מהמרחב, מחלק את R^2 לשני חצאי מרחב (w יהיה ניצב לקו הזה).

- **מחלקת ההיפותזות תהיה:**

$$\mathcal{H}_{\text{half}} := \left\{h_{\mathbf{w},b}(\mathbf{x}) = sign\left(\mathbf{x}^{\top}\mathbf{w} + b\right) \mid \mathbf{w} \in \mathbb{R}^d, b \in \mathbb{R}\right\}$$

- **הנחת הריליזביליות:** נניח כי w שייך למחלקת ההיפוטזות. $\exists w \in R^d, b \in R, \forall i \in [m]$ עבורם

$$sign(\langle \mathbf{w_i}, \mathbf{x_i} \rangle + b) \cdot y_i = 1 \iff y_i \cdot (\langle \mathbf{w}_i, \mathbf{x_i} \rangle + b) > 0$$

המקרה ההומוגני: כאשר $b = 0$ (w יעבור בראשית הצירים), ומתקיים $\left\{x \mid w^{\top}x = 0\right\} = w^{\perp}$ נקרא w ניצב.

- **במקרה הלא הומוגני:** עבור המכפלה $\langle \mathbf{w}, \mathbf{x} \rangle + b$ נפתור באופן הבא: נוסיף לווקטור w בקאורדינטה ה 0 את b, ולווקטור x נוסיף בקאורדינטה הראשונה 1, ונפתור את $\langle \mathbf{w}', \mathbf{x}' \rangle$.
מחלקת ההיפותזות תהיה:

$$\mathcal{H}_{\text{half}} := \left\{h_{\mathbf{w}}(\mathbf{x}) = sign\left(\mathbf{x}^{\top}\mathbf{w}\right) \mid \mathbf{w} \in \mathbb{R}^d\right\}$$

5.1.1 עיקרון ERM לבעיות סיווג:

- **טעות:**
במקרה הראשון: $sign(\langle \mathbf{w_i}, \mathbf{x_i} \rangle) \cdot y_i = -1$
במקרה השני: $sign(\langle \mathbf{w_i}, \mathbf{x_i} \rangle) \cdot y_i < 0$.

- **הגדרת ה** $Loss$**:** נספור את כמות השגיאות (עם אינדיקטור)

$$L_S\left(h_{\mathbf{w}}\right) := \sum_{i=1}^{m} 1\left[y_i \left\langle \mathbf{x}_i, \mathbf{w} \right\rangle < 0\right]$$

- **הפרדה:** נרצה להביא את $L_S\left(h_{\mathbf{w}}\right)$ למינימום, תחת הנחת הריליזביליות אנו יודעים כי קיים w שמביא את הפונקציה ל 0.

- **נמיר לבעיית אופטמיזציה:** אנו רוצים את $min\left(L_S\left(h_{\mathbf{w}}\right)\right)$, לכן עבור האילוץ הבא $y_i \left\langle \mathbf{x}_i, \mathbf{w} \right\rangle > 0$ נחפש את w שעומד באילוצים (למעשה זאת בעיית פיזביליות ולא בעית אופט').

- **אלגוריתם** $Perceptron$**:** נרצה למצוא הפרדה על הדוגמאות, (תחת הנחת הריליזביליות האלגוריתם ייעצר, אחרת הוא יכול להמשיך לנצח).

Algorithm 1 Batch-Perceptron

```
procedure PERCEPTRON(S = {(x_i, y_i)}_{i=1}^m)
    w^(1) ← 0                                    ▷ Initialize parameters
    for t = 1, 2, ... do
        if ∃i s.t. y_i ⟨w^(t), x_i⟩ ≤ 0 then
            w^(t+1) = w^(t) + y_i x_i
        else
            return w^(t)
        end if
    end for
end procedure
```

הרעיון: אנו

מתקנים בכל איטרציה לכיוון הטעות.

- **הגדרה - מרחק:** יהי (w,b) על מישור עבור נקודה $x\in R^d$ המרחק בין x לעל מישור מוגדר באופן הבא:

$$d((w,b),x)=min_{v:\langle v,w\rangle+b=0}||x-v||$$

- **הגדרה - השול** $Margin$: יהי (w,b) על מישור, ואוסף נקודות $S=(x_1...x_m)\in R^d$ (x הוא ווקטור), המרחק בין S לעל המישור מוגדר כך:

$$M((w,b),S)=min_{i\in[m]}d((w,b),x_i)$$

- **הגדרה - הווקטור התומך** $Support\ vector$: הווקטור שמגדיר את השול נקרא התומך.

5.1.2 $Hard\ SVM$:

- **בעיית** $Hard\ SVM$: נרצה למצוא את h_w שממקסמת לנו את השול. עבור האילוץ $(y_i\left\langle \mathbf{x}_i,\mathbf{w}\right\rangle+b)>0$:

$$max\left(M((w,b),S)\right)$$

2 באופן שקול:

$$argmax_{w:||w||=1}min_{i\in[m]}\left|\langle w_i,x_i\rangle+b\right|$$

3 עבור פתרון פיזבילי זה שקול:

$$argmax_{w:||w||=1}min_{i\in[m]}y_i\cdot(\langle w_i,x_i\rangle+b)$$

4 באופן שקול: ניתן להוריד את האילוצים

$$argmax_{(w,b)\ \ w:||w||=1}min_{i\in[m]}y_i\cdot(\langle w_i,x_i\rangle+b)$$

5 באופן שקול: עבר האילוץ הבא $y_i\left(\langle\mathbf{x}_i,\mathbf{w}\rangle+b\right)\geq 1$ נחפש את -

$$\underset{(\mathbf{w},b)}{argmin}\quad\|\mathbf{w}\|^2$$

טענה: אם (w^*,b^*) הוא פתרון ל 5, אזי, הפתרון עבור 4 הוא $\hat{w}\frac{\mathbf{w}^*}{\|\mathbf{w}^*\|},\hat{b}=\frac{b^*}{\|\mathbf{w}^*\|}$

- **מרחק לווקטור מנורמל:** יהי (w,b) על מישור, כך ש w מנורמל, ו $x\in R^d$ אזי המרחק בין x לעל מישור הוא $|\langle x,w\rangle+b|$.

5.1.3 $Soft\ SVM$:

- **$Soft\ SVM$:** אנו מחפשים סיווג לשני $classes$. מכיוון שהנחת הריליזביליות לרוב לא מתקיימת בעולם האמיתי נרצה אלגוריתם שמוצא את w הטוב ביותר עם שגיאות מינימליות.

- **שגיאה:** נגדיר שגיאה ע"י המרחק של הנקודה שתייגנו לא נכון מהשול (נשים לב שהשגיאה לא מוגדרת על המרחק מ w).

- **מה אנחנו דורשים:** נרצה עבור כל נקודה שיתקיים האילוץ

$$y_i\left(\langle\mathbf{x}_i,\mathbf{w}\rangle+b\right)\geq 1-\xi_i$$

בעיית האופטימיזציה: ועבור האילוץ הזה נרצה למצוא את $\underset{(\mathbf{w},b)}{argmin}\quad\|\mathbf{w}\|^2$

ונרציה שהשגיאות יקיימו:

$$\xi_i\geq 0\ \ and\ \ \frac{1}{m}\sum_{i=1}^{m}\xi_i\leq C$$

הפרמטר C: אם נבחר C כך שאם הוא יהיה קטן - אנו נאפשר שגיאות קטנות מאד, ז"א שהמדגם יהיה רגיש לרעש (var).

לעומת זאת אם נקח C גדול אנו נאפשר שגיאות גדולות אזי ה $bias$ יהיה גבוה.

- **דרך נוספת:** נגדיר את C בתור אילוץ נוסף, כך:

$$\underset{(\mathbf{w},b),\{\xi_i\}}{argmin}\lambda\cdot||w||^2+\frac{1}{m}\sum_{i=1}^{m}\xi_i$$

בנוסף לאילוץ $y_i\left(\langle\mathbf{x}_i,\mathbf{w}\rangle+b\right)\geq 1-\xi_i$ ונדרוש ש $\xi_i\geq 0$.

מה נקבע מראש: את הפרמטר λ נגדיר מראש, והוא קובע את המשקל של החלק הראשון במשוואה, וזה משפיע על המשקל של החלק השני וכן הלאה.

אם נבחר λ גדול מאד - המשקל של הנורמה יהיה גדול והמשוואה לא תהיה רגישה לרעש.

אם נבחר λ קטן - נהיה רגישים לרעש.

- **הגדרה - פרמטר רגולריזציה:** λ נקרא פרמטר רגולריזציה, משקלו משפיע על הפתרון.

- **הגדרה** $hinge\ Loss$**:** פונקציה שמחזירה 0 אם $a>1$, אחרת $1-a$

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

- **בהינתן סט אימון ניתן להגדיר את** $Soft\ SVM$ **ע"י:**

$$\underset{\mathbf{w},b}{argmin}\left(\lambda\|\mathbf{w}\|^2+L_S^{\text{hinge}}((\mathbf{w},b))\right)$$

עבור: $L_S^{\text{hinge}}((\mathbf{w},b)):=\frac{1}{m}\sum\ell^{\text{hinge}}\left(y_i\left\langle\mathbf{x}_i,\mathbf{w}\right\rangle\right)$

5.2 הרצאה 5 - $Agnostic\ PAC$:

- **בשונה מ** PAC**:** נרשה רעש, לא נניח את הנחת הריליזביליות ונוכל להשתמש בכל פונקציית $Loss$.
- **נאפשר רעש:** נסתכל על מרחב המכפלה $\mathcal{X}\times\mathcal{Y}$ מעל מרחב הסתברות D, כלומר $(x,y)\sim D$ מתפלגים בהתפלגות משותפת מעל D (ולא רק x). כאן אין לנו פונקציה דטרמניסטית f אלא הדגימות הן מעל מרחב התפלגות מסויים.
דרך אחת לחשוב על מרחב המכפלה:

$$\mathbb{P}(Y=+1\mid X=x)=p(x)$$

דרך נוספת: נבחר את y ולאחר מכן נבחר את x בהינתן y.

- **ביטול** $loss$ $0-1$:

נגדיר פונקציית $loss$ **כללית:**

$$\ell:\mathcal{H}\times Z\rightarrow[0,\infty),\text{ where }Z=\mathcal{X}\times\mathcal{Y}.\text{ For }z=(x,y)$$

ואת הביצועים נגדיר כך: עבור $z=(x,y)\in Z$ and $Z=\mathcal{X}\times\mathcal{Y}$ where

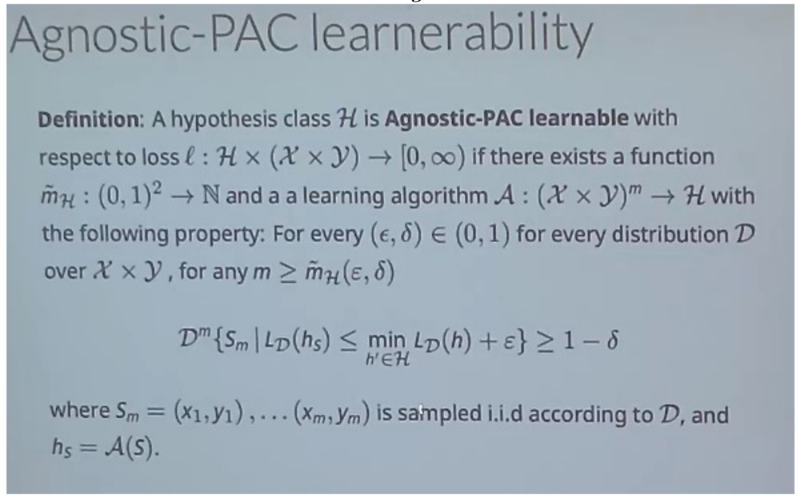
$$L_{\mathcal{D}}(h):=\mathbb{E}_{z\sim\mathcal{D}}[\ell(h,z)]$$

הגדרת $aproximatly\ correct$**:** נקבע ε ונאמר שכלל החלטה הוא $aproximatly\ correct$ אם הוא מקיים

$$L_{\mathcal{D}}(h)\leq\min_{h'\in\mathcal{H}}L_{\mathcal{D}}(h')+\epsilon$$

כלומר, ה $loss$ שקיבלנו הוא לא יותר מאפסילון מעל ה $loss$ המינימלי האפשרי (ה $loss$ הכי קטן מכל כלל החלטה ב H).

- **הגדרה - למידה** $Agnostic\ PAC$:



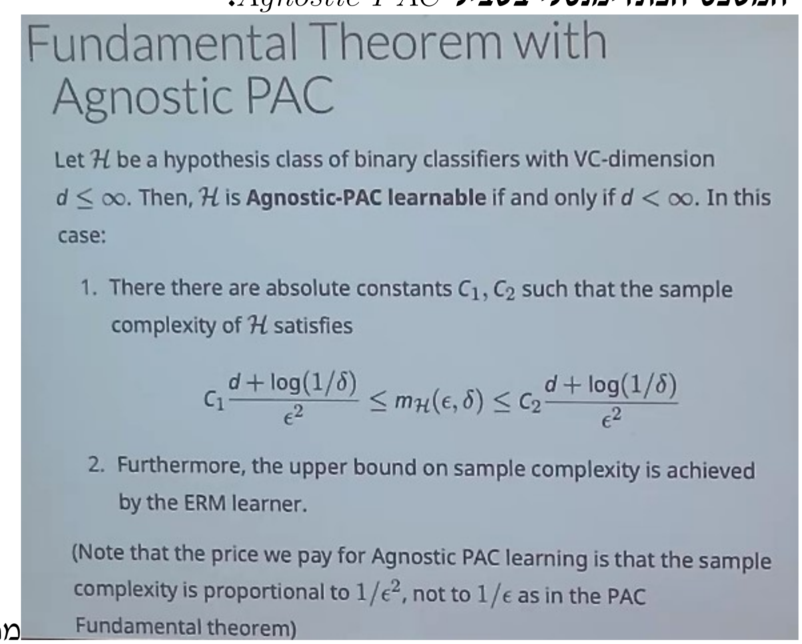
- **הגדרה - לומד** ERM: ER הוא -

$$L_s(h) := \frac{1}{m} \sum_{i=1}^{m} \ell\left(h, z_i\right)$$

לומד ERM **הוא:**

$$\mathcal{A}_{ERM} : S \mapsto argmin_{h \in \mathcal{H} L_s(h)}$$

- **המשפט הפונדימנטלי בשביל** $Agnostic\ PAC$:



מחלקת היפותזות היא למידה $Agnostic\ PAC$ אמ"מ מימד VC שלה הוא סופי.

- **הגדרה - מדגם ε מייצג:** סט אימון S ייקרא ε מייצג אם

$$\forall h \in \mathcal{H} \, |L_s(h) - L_{\mathcal{D}}(h)| < \epsilon$$

- **למה:** יהי S מדגם $\frac{\varepsilon}{2}$ מייצג עבור H, D, l ויהי h_s כל כלל החלטה שהוא $ERM_{H(s)}$ אזי מתקיים:

$$L_{\mathcal{D}}\left(h_S\right) \leq \min_{h \in \mathcal{H}} L_{\mathcal{D}}(h) + \epsilon$$

אם המדגם אפסילון מייצג אזי שגיאת ההכללה של כלל ERM היא לכל היותר אפסילון מעל המינימילי.

- **הגדרה - $UniformConvergence(UC)$:** תהי H מחלקת היפותזות נאמר שהיא UC אם קיים $m_{\mathcal{H}}^{UC} : (0,1)^2 \to \mathbb{N}$ כך שלכל ε, δ ולכל התפלגות D מתקיים

$$\mathcal{D}^m\left(\left\{s \in (\mathcal{X} \times \mathcal{Y})^m \mid S \text{ is } \epsilon\text{-representative } \right\}\right) \geq 1 - \delta.$$

- **למה:** אם H היא UC אזי היא למידה $Agnostic\ PAC$.

- **למה:** אם H מקיימת שמימד VC שלה סופי, היא גם מקיימת את תכונת UC ולכן למידה $Agnostic\ PAC$.

6 שבוע 6:

6.1 תרגול 6:

6.1.1 רגרסיה לוגיסטית:

- **מוטיבציה:** אנו בודקים מה ההסתברות שאנו שהדגימה שייכת ל $class$ מסויים.

- **מרחב ההסתברות:** נניח כי הדגימות מתפלגות ברנולי.

$$p\left(y_i \mid x_i\right) = Ber\left(y_i \mid \emptyset_w\left(x_i\right)\right)$$

ונגדיר את $\emptyset_w\left(x_i\right)$ כך: הרכבה של פונקציית הסיגמואיד $\sigma(x)$

$$\emptyset_w\left(x_i\right) = \sigma\left(\left(x_i\right)^T w\right)$$

עבור $\sigma(x) = \frac{e^x}{e^x+1}$.

- **במקרה של $Multy\ class$:** פונקציית הסיגמואיד רלוונטית רק לשני $classes$, אותו הדבר לגבי התפלגות Ber. לכן במקרה של יותר משני $classes$ נעבוד עם התפלגות $Multynom$ עם ווקטור הסתברות. ובתור פונקציה נשתמש בפונקציה $softMax = \sigma(x)_i = \frac{e^{x_i}}{\sum_{j=1}^{k} e^{x_j}}$

- **מחלקת ההיפותזות:**

$$H_{logistic} = \left\{h_w(x) = \sigma\left(x^\top w\right) \mid w \in R^{d+1}\right\}$$

כאשר אנו מכניסים את ה $intercept$ לתוך $w[0]$.

- **השלב הבא - סיווג למחלקות:** נבחר פרמט $\alpha \in [0,1]$ ונגדיר משתנה אינדיקטור שיבדוק אם $\hat{y}(x) = 1_{h_w}(x) > \alpha$.
כלומר: נקבע סף α כך שכל נקודה שסווגה מעליו תקבל 1, וכל נקודה שסווגה מתחתיו תקבל -1 .

- **איך לומדים:** נלמד בעזרת פונקציית $MaxLikelihood$, אנו רוצים להעריך עד כמה טוב ה w שלנו בעזרת פונקציית $Likelihood$ ונרצה את w המקסימלי.
הפונקציה:

$$Likelihood\left(w \mid X,Y\right) = P\left(y_1...y_m \mid X,w\right) = \prod_{i=1}^{m} P\left(y_i \mid x_i,w\right) =$$

$$\prod_{i=1,y_i=0}^{m} P\left(y_i \mid x_i,w\right) \cdot \prod_{i=1,y_i=1}^{m} P\left(y_i \mid x_i,w\right) =$$

$$\prod_{i=1,y_i=1}^{m} \emptyset_w\left(x_i\right) \cdot \prod_{i=1,y_i=0}^{m} \left(1-\emptyset_w\left(x_i\right)\right) =$$

$$\prod_{i=1}^{m} \emptyset_w\left(x_i\right)^{y_i} \cdot \left(1-\emptyset_w\left(x_i\right)\right)^{(1-y_i)}$$

ניתן למקסם על ה Log, לכן:

$$logLikelihood\left(w \mid X,Y\right) = \sum_{i=1}^{m} y_i \cdot x_i^T w - log\left(1+e^{x_i^T w}\right)$$

- **הפתרון:** נבחר את $h_w \in H_{logistic}$ כך שנקבל w עם לייקליהוד מקסימלי.

$$\hat{w} = argmax_{w \in R^{d+1}}\left(\sum_{i=1}^{m} y_i \cdot x_i^T w - log\left(1+e^{x_i^T w}\right)\right)$$

הערה: אם היינו מגדירים פונקציית $Loss$ באופן הבא:

$$Loss(h_w) = \log(1+\exp(-y\langle w,x\rangle))$$

והיינו מחשבים לפי עיקרון ERM (מזעור הלוס), אזי היינו מגיעים לאותו w.

6.1.2 מסווג בייס:

- **הנחות:** נניח כי המשתנים y מתפלגים מולטינומית, הפיצ'רים ב"ת ו $x_j \mid y=k$ מתפלגים גאוסיאנית כך:

$$y \sim Multinomial(\pi)$$
$$x_j \mid y=k \overset{\text{ind.}}{\sim} \mathcal{N}\left(\mu_{kj}, \sigma_{kj}^2\right)$$

- **פונקציית התפלגות משותפת:** עד עכשיו הסתכלנו על X כדטרמיניסטי ורק על y כמורעש.
כעת נניח כי גם X וגם y הגיעו מהתפלגות כלשהי (ו X אינינו דטר').
פונקציית התפלגות המשותפת:

$$f_{X,Y}(x,y) = f_{Y|X=x}(y) \cdot f_X(x) = f_{X|Y=y}(x) \cdot f_Y(y)$$

- **מסווג בייס עבור x - איך נסווג:**

$$h^{bayes}(x) = argmax_y \left(f_{Y|X=x}(y)\right) = argmax_y \left(f_{X|Y=y}(x) \cdot f_Y(y)\right)$$

יעילות המסווג תחת $missclasification\ error$**:** מסווג בייס הוא המסווג הטוב ביותר עבור הפונקציה $missclasification\ error$ הבאה -

$$L_D(h) = E_{(x,y)\sim 0}\left[h(x) \neq y\right] =$$

כאשר $E_{(x,y)\sim 0}$ מסמן תוחלת על **כל הדגימות בעולם**.

$$= \int\limits_{X,Y} \left(f_{X,Y}(x,y) \cdot 1_{h(x)\neq y}\right) = \int\limits_X \left(f_X(x)\right) \cdot \int\limits_Y \left(f_{Y|X=x}(y) \cdot 1_{h(x)\neq y}\right) =$$

כעת, נסתכל על האינדיקטור המשלים:

$$= \int\limits_X \left(f_X(x)\right) \cdot \int\limits_Y \left(f_{Y|X=x}(y) \cdot \left(1 - \left(1_{h(x)=y}\right)\right)\right) = \int\limits_X \left(f_X(x)\right) \cdot \int\limits_Y \left(f_{Y|X=x}(y) - f_{Y|X=x}(y) \cdot 1_{h(x)=y}\right) =$$

$$= \int\limits_X \left(f_X(x)\right) \cdot \left(1 - f_{Y|X=x}(h(x))\right) \geq \int\limits_X \left(f_X(x)\right) \left(1 - f_{Y|X=x}(h^{bayse}(x))\right) = L_D\left(h^{bayse}\right)$$

למעשה קיבלנו כי לכל h מתקיים כי $L_D\left(h^{bayse}\right)$ קטן ממנו.

6.1.3 $Linear\ Discriminant\ Analysis\ (LDA)$:

- **הנחות על מרחב ההסתברות ממנו הגיעו הדגימות:** אנו לא יודעים שום דבר על ההתפלגות, אלא אנו רואים חלק קטן ממנה, לכן נניח הנחות על המרחב ממנו הגיעו הדגימות. ותחת הנחות אלו נבנה אלגוריתם של h^{bayse}.

- **הגדרה ־ התפלגות מולטינומית:** הכללה של ברנולי ל k מחלקות.
מרחב ההסתברות: $\Omega = \{1....K\}$
מ"מ: $X : \Omega \Rightarrow [0,1]$,
מתקיים ש X מתפלג פולינומי עם ווקטור $\pi \in [0,1]^k$ המקיים $\sum_{i=1}^{k} \pi_i = 1$.
ונאמר ש $P(X=j) = \pi_j$ (הקאורדינטה ה j של הווקטור π מייצגת את הסתברות ש $x = j$) וכתוב ש $X \sim Mult(\pi)$

- **הנחות:**
1: נניח ש $y_i \sim Mult(\pi)$
2: ונניח ש $x_i|y_i \sim N\left(\mu_{y_i}, \Sigma\right)$, לכל מחלקה y יש את התוחלת שמתאימה לה, אך כל המחלקות חולקות את אותה ה Σ. (כאשר y_i מייצג את המחלקה y_i מתוך k מחלקות)

- **איך לומדים:** אנו רוצים לשערך את μ_{y_i}, Σ, כך שנוכל לסווג כל דגימה עם אותן μ_{y_i}, Σ. בנוסף נרצה לשערך את π ־ מאיפה y נדגם.

- **המסווג האופטימלי:** תחת ההנחות לעיל, ועבור $a_k = \sum^{-1} \mu_k$ ו $b_k = \log\left(\pi_k\right) - \frac{1}{2}\mu_k \Sigma^{-1} \mu_k$ נחפש את:

$$\hat{y}(x) = \underset{k}{argmax}\left(a_k^{\top} x + b_k\right)$$

כך נמצא את המסווג הטוב ביותר.

- **שיערוך:** נעשה באמצעות $MaxLikelihood$, בהינתן סט אימון נמצא את הפרמטרים $\Theta_{\mu_{y_i}, \Sigma, \pi}$ כך:

$$Likelihood\left(\Theta|X,Y\right) = f_{X,Y|\Theta}\left(\{(x_i, y_i)\}_{i=1}^{m}\right) = \prod_{i=1}^{m} f_{X,Y|\Theta}(x_i, y_i) =$$

$$= \prod_{i=1}^{m} f_{X|Y=y_i}(x_i) \cdot f_{T|\Theta}(y_i) = \prod_{i=1}^{m} N\left(x_i|\mu_{y_i}, \Sigma\right) \cdot Multy\left(y_i|\pi\right)$$

כשנפעיל Log נקבל:

$$LogLikelihood\left(\Theta|X,Y\right) = \sum_{k=1}^{K}\left[n_k \cdot \log\left(\pi_k\right) - \frac{1}{2}\sum_{i:y_i=k}\left(\mathbf{x}_i - \mu_k\right)^{\top}\Sigma^{-1}\left(\mathbf{x}_i - \mu_k\right)\right] - \frac{md}{2}\log(2\pi) - \frac{m}{2}\log|\Sigma|$$

עבור: n_k ־ כמות הדגימות שבהן $y_i = k$.

מציאת המקסימום: נגזור ונשווה ל 0. אנו מחפשים את μ_{y_i}, Σ, π. נשים לב שאנחנו צריכים לשמור על האילוצים של π, לכן נשתמש בכופלי לגראנז'. נגדיר:

$$g(\pi) = \sum_{k} \pi_k - 1$$

הרעיון הוא - ש $g(\pi)$ יתאפס כשגזור ונשווה ל 0. ונגדיר את $Lagrangian$:

$$LogLikelihood\left(\Theta|X,Y\right)-\lambda\cdot g(\pi)$$

כדי למצוא את π נגזור ונשווה ל 0, ונקבל :

$$\frac{\partial\mathcal{L}}{\partial\pi_k}=\frac{n_k}{\pi_k}-\lambda=0\Rightarrow\pi_k=\frac{n_k}{\lambda}$$

כדי לקיים את האילצים של π נבחר $\lambda=m$.

כדי למצוא את μ_{y_i}: (וכך למצוא את ההתפלגות של X)

$$\hat{\mu}_k^{MLE}=\frac{1}{n_k}\cdot\sum_{i=1}^{m}1_{[y_i=k]}\cdot x_i$$

כדי למצוא את Σ:

$$\hat{\Sigma}^{MLE}=\frac{1}{m}\sum_{i}\left(\mathbf{x}_i-\hat{\mu}_{y_i}^{MLE}\right)\left(\mathbf{x}_i-\hat{\mu}_{y_i}^{MLE}\right)^\top$$

6.1.4 $Quadratic\ Discriminant\ Analysis\ (QDA)$:

- **הנחות:**
  1: נניח ש $y_i\sim Mult(\pi)$
  2: ונניח ש $x_i|y_i\sim N\left(\mu_{y_i},\Sigma_{y_i}\right)$.
  כלומר: כאן יש גם מטריה שונויות שונה לכל מחלקה.
- **שיערוך:** את π ו μ_{y_i} נשערך באותו האופן. אך את מטריצת השונויות נעשה אחרת.
- **מציאת מטריצה השונויות:**

$$\hat{\Sigma_k}^{MLE}=\frac{1}{n_k}\sum_{i:y_i=k}\left(\mathbf{x}_i-\hat{\mu}_{y_i}^{MLE}\right)\left(\mathbf{x}_i-\hat{\mu}_{y_i}^{MLE}\right)^\top$$

6.2 הרצאה 6:

- אין הרצאה השבוע - חופש פסח.

7 שבוע 7:

7.1 תרגול 7 למידת PAC ומימד VC:

7.1.1 למידת PAC:

- **הנתונים:** יש לנו דומיין X, ו Y שמייצג בעיתת קלסיפיקציה בינארית, סט דגימות $S=\{(x_i,y_i)\}_{i=1}^m$ שמתפלג אחיד מעל התפלגות D. ואנו מחפשים מיפוי של $X \Rightarrow Y$.
- **מה אנו מנסים:** ללמוד מחלקת היפותזות H. מחפשים אלגורתים $A(S)$ שבהינתן S מחזירה לנו $h \in H$.
- **הנחת הרליזביליות:** נניח כי הפונקציה שאנו מחפשים נמצאת הקבוצה H.
- **פונקציית $loss$:** יש לנו פונקציית $loss$ שתעריך לנו כמה האלגוריתם שמצאנו טוב.
  אם היינו יודעים את ההתפלגות D ואת f האמיתית, היינו משתמשים בפונקציה:

  $$L_D(h) = E_{x\sim D}[h(x) \neq f(x)]$$

  אך אנו לא יודעים מהן D, f לכן נמצא את המינימום של $risk$:

  $$L_S(h) = \frac{1}{m}|i : h(x)_i \neq y_i|$$

- **הגדרה - מחלקה H למידת PAC, ואלגוריתם לומד:**

Definition 1.1 An hypothesis class $\mathcal{H}$ is *PAC learnable* if there exists a function $m_{\mathcal{H}} : (0,1)^2 \to \mathbb{N}$ and a learning algorithm $\mathcal{A}$ such that:
- For every $\varepsilon, \delta \in (0,1)$
- For every distribution $\mathcal{D}$ over $\mathcal{X}$
- For every labeling function $f : \mathcal{X} \to \{0,1\}$

if the realizable assumption holds with respect to $\mathcal{H}, \mathcal{D}, f$, when running the learning algorithm $\mathcal{A}$ on $m \geq m_{\mathcal{H}}(\varepsilon,\delta)$ *i.i.d* samples drawn from $\mathcal{D}$ and labeled by f, the algorithm returns a hypothesis $h_S = \mathcal{A}(S)$ such that:

$$\mathbb{P}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}(h_S) \leq \varepsilon] \geq 1-\delta$$

- **כיצד נוכיח למידת PAC:**
  1: נציע אלגוריתם לומד A.
  2: נמצא פונקציה m_H. נעשה זאת כך - נתחיל מההסתברות $\mathbb{P}_{S\sim\mathcal{D}^m}\left[L_{\mathcal{D}}\left(h_S\right) \leq \varepsilon\right] \geq 1-\delta$ ונרצה להגיע לביטוי שתלוי ב m.

7.1.2 מימד VC של מחלקות אינסופיות:

- **אינטואציה:** תנאי נוסף ושקול שמגדיר לנו האם מחלקת היפותזות H היא למידת PAC.

- **הגדרה - $Restriction$:** תהי H מחלקת היפותזות ו $C=\{\mathbf{x}_1,\ldots,\mathbf{x}_m\}\subset\mathcal{X}$ סט דגימות, אזי ה $Restriction$ של H על C מוגדרת להיות:

$$\mathcal{H}_C:=\{h_C=(h(\mathbf{x}_1),\ldots,h(\mathbf{x}_m)\mid h\in\mathcal{H})\}$$

כלומר - מה כל אחת מההיפותזות h שלנו עושות על כל אחת מהדגימות. למעשה הקבוצה $\mathcal{H}_C$ היא קבוצת וקטורים ממימד m והגודל המקסימלי שלה הוא $|H|\le 2^{|C|}=2^m$

- **הגדרה - מחלקה מנתצת:** נאמר שמחלקת היפותזות H מנתצת את C, אמ"מ $|H|=2^{|C|}$.
כלומר - עבור כל וקטור ב H_C שמייצג את הלייבלים, יש לנו h כזו שתחזיר את הווקטור הזה על קבוצת הנקודות שלנו.

- **הגדרה - מימד VC:** מימד VC של H הוא הגודל של הקבוצה C הגדולה ביור שניתן לנתץ אותה

$$VC-Dim(\mathcal{H}):=\sup\{m\in\mathbb{N}|\exists C\subset\mathcal{X}|C\mid=m\text{ s.t. }\mathcal{H}\text{ shatters }C\}$$

- **כיצד נוכיח מימד VC של מחלקה H:** אם אנו טוענים כי המימד הוא d, נמצא קבוצה בגודל d ונוכיח כי H מנתצת אותה, וגם נראה שלכל קבוצה בגודל $d+1$ מתקיים כי H לא מנתצת אותה.

7.1.3 תכונות של מימד VC של מחלקות סופיות:

- **טענה:** עבור מחלקה H סופית מתקיים

$$VC-Dim(\mathcal{H})\le log_2(|H|)$$

ולכן היא למידה PAC.

- **טענה:** עבור רוב המחלקות אי השויון מטענה קודמת הוא א"ש חזק. לדוגמה עבור H הבאה

$$H_{\text{Singeltan}}=\{h_z(x)\cdot:1\}$$

מימד VC שלה שווה ל 1 והגודל של H יכול להיות אינסופי.

- **טענה:** קיימת מחלקת היפותזות שמקיימת $VC-Dim(\mathcal{H})=log_2(|H|)$, דוגמה: עבור $X=\{-1,1\}^d$ ומחלקת היפותזות $H=Y^X$ - כל הפונקציו האפשריות $X\Rightarrow Y$, מתקיים:

$$|H|=|Y|^{|X|}=2^{2^d}\Rightarrow log_2(|H|)=2^d$$

ומימד VC שווה ל 2^d כי זה מספר הלייבלים שיש לנו והגדרנו כי H היא כל הצירופים האפשריים.

- **נשים לב !** כי בד"כ מימד VC מוגדר להיות לפי מספר הפרמטרים שיש לנו, מספר הפרמטרים שהפונקציה מוגדרת לפיהם. אך זה לא תמיד כך.

7.2 הרצאה 7 - מטא אלגוריתם:

- **הגדרה - מטא אלגוריתם:** אלגוריתם שנוכל להפעיל אותו על האלגוריתם הלומד שלנו ולשפר את הביצועים שלו.

7.2.1 וועדות - החלטת הרוב:

- **הרעיון:** יש לנו וועדה עם T חברים וכל אחד צריך להצביע בעד או נגד ההחלטה. בדיעבד נו נדע אם ההחלטה שהתקבלה הייתה טובה או לא. נניח כי כל אחד מהם צודק בהסתברות p, והצבעת הרוב מתקבלת.
נמדל כך: עבור T חברי וועדה - $X_1,\ldots,X_T\ i.i.dBernoulli(p)$ וההחלטה היא $\bar{X}:=sign\left(\sum_{t=1}^{T}X_t\right)$. וההחלטה תסוות בינארית $\{-1,1\}$

- **נשים לב:** כי אם $p>0.5$ אזי הסיכוי שנקבל תשובה נכונה גדל עם גודל הקבוצה.

- **חישוב השונות עבור מ"מ ב"ת:** נרצה לדעת האם ההחלטות עקביות,לכן נחשב את השונות, עבור שונות לכל חבר וועדה = σ^2, עבור **ערכי אמת** השונות של ההחלטה היא:

$$\operatorname{var}\left(\bar{X}\right):=\frac{1}{T}\sum_{t=1}^{T}X_t=\frac{\sigma^2}{T}$$

- **מסקנה:** אם יש לנו מ"מ ב"ת עם $p>0.5$ נעדיף לעשות הצבעה.

- **מה קורה אם המ"מ תלויים:** נניח כי כל שני חברים X_i,X_j יש להם קורולציה ρ. ההסתברות לקבל תשובה נכונה יורדת.
כעת השונות היא:

$$\rho\cdot\left(\sigma^2\right)+(1-\rho)\cdot\frac{\sigma^2}{T}$$

- **מה נעשה:** נאמן כמה שיותר מסווגים ונבחר את הצבעת הרוב. אם הם ב"ת אנו נשפר את המצב, אך אם יש להם קורולציה חזקה יש גבול לשיפור שניתן להשיג.

- **רעיון הביצוע:** נקח את הדאטה שלנו ונעשה עליו מניפולציות בכדי ליצור דאטה חדש $(Bootstrap)$. כך נוכל לאמן מודלים שונים על דאטה שונה ולהשיג תוצאה מיטבית.

7.2.2 $Bootstrap$ - יצירת דאטה:

- **כיצד ניצור דאטה חדש:** בהינתן סט אימון $S=\{(x_i,y_i)\}_{i=1}^{m}$ נייצר סט אימון חדש S^{*1} באופן הבא:
נדגום m פעמים מתוך S עם חזרות כך שחלק מהאיברים ב S לא יופיעו ב S^{*1}, וחלקם יופיעו כמה פעמים.
נוכל לחזור על שלב זה B פעמים כך שיהיו לנו B סטים.

- **הגדרה - התפלגות אמפירית:** עבור סט אימון S שמתפלג iid מעל התפלגות D, נרצה לקרב את ההסתברות של D.

$$\widehat{\mathcal{D}}_S((x,y)=(x,y))=\begin{cases}\frac{1}{m} & (x,y)\in S\\ 0 & (x,y)\notin S\end{cases}$$

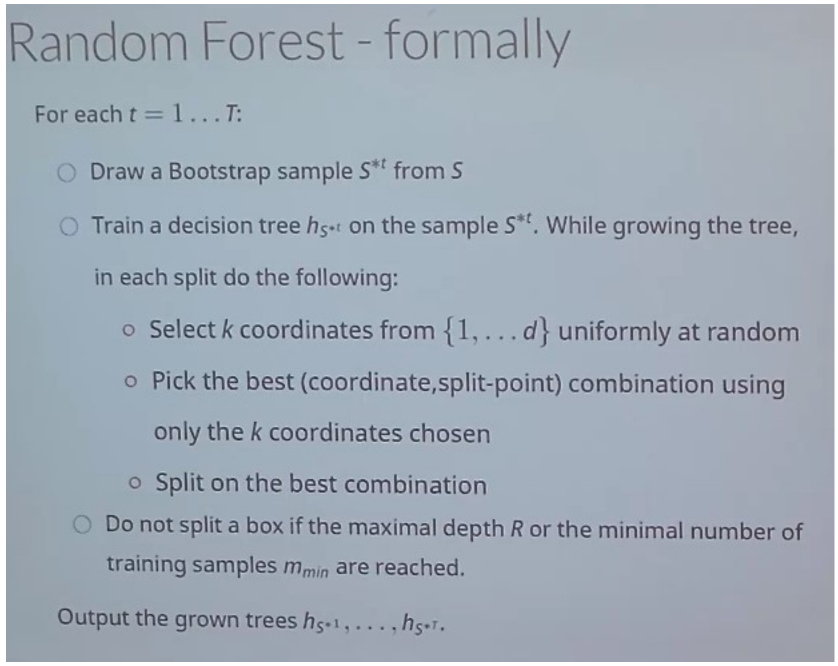
- **טענה:** מדגם $Bootstrap$ מתפלג אחיד וב"ת על $\widehat{\mathcal{D}}_S$. וככל ש M גדול יותר מתקיים כי $\widehat{\mathcal{D}}_S$ מתכנס ל D.
- **כיצד נשתמש - $Bagging$:** ניצור מלא מדגמי $Bootstrap$, נאמן את האלגוריתם על כל מדגם בנפרד ונקח את הצבעת הרוב.

7.2.3 $Bagging$ - גידול יער:

- **כלל ההחלטה:** נקח את האלגוריתם A ואת סט האימון S, ניצור ממנו T מדגמי $Bootstrap$ - $S^{*1}...S^{*T}$, ונפעיל על כל אחד מהם את A והאלגוריתם הלומד שלנו יחזיר את הצבעת הרוב.
- **הערה:** נשים לב לא לבחור אלגוריתם Aשמתקשה לטפל בנקודות חוזרות.
- **עצי החלטה:** מימוש של $Bagging$ עם עצי החלטה מחזיר תוצאה טובה.
- **החסרונות:** צריך לאחסן את כל המודלים והעלות גבוה, בנוסף בשלב החיזוי נצטרך לשאול B כללי החלטה מה ההחלטה שלהם וזה יעלה לנו יותר.
- **למעשה:** אנו מורידים את ה var מבלי להעלות את ה $bias$.
- **הגדרה - די קורלציה:** נרצה להוריד את הקורלציה בין כללי ההחלטה השונים בכדי לדייק עוד את כלל ההחלטה שלנו (כי כרגע הוא חסום ע"י הקורולציה).
- **האלגוריתם $Random\ Forest$:** אלגוריתם לגידול יער, למעשה זה עץ גזום שעליו אנו מפעילים $Bagging$ עם דיקורולציה.

  כיצד נבצע את הדי קורלציה: עבור d קאורדינטות, נבחר $k\le d$ ונגדיר לעץ בגידול שיעשה ספליט רק על k קאורדינטות. (למעשה אנו מעלים את ה $bias$ כי אנו מגבילים את העץ אך זה משתלם חלנו כי אנו מורידים את הקורלציה.)

  פסואודו קוד:



- **כיצד נבחר את** T**:** נצטרך לבדוק מספר אופציות ולבחור את הטובה ביותר, נשים לב כי משלב מסויים התוצאה תישאר קבועה ותפסיק להשתפר.

- $Bias - variance$**:** השונות תרד בכל שלב עד גבול מסויים, אך ה $bias$ יעלה בקצת.

7.2.4 $Boosting$:

- **הרעיון:** נקח אלגוריתם לומד חלש A - השגיאה שלה טובה אך ניתנת לשיפור, ונהפוך אותו לוועדה שיש לה $accuracy$ טוב מאד.
באופן שונה מ $Bagging$ - נבחר מדגמים ונניח כי ההתפלגות על כל אחד מהמדגמים שונה.

- **מימוש:** כל חבר וועדה יאמן את האלגוריתם A על סט אימון S_tשמדמה iid על התפלגות D^t **שונה** לכל סט אימון.
נעשה זאת באופן סדרתי, כאשר נסיים לאמן את h_t, נעדכן את ההתפלגות באופן שמגדיל את התפלגות בסט האימון בנקודות ש h_t טעה עליהן. כלומר נשחק עם D על הנקודות בהן **טעינו** ונעלה את המשקל שלהן, כך נוכל לצפות לשיפור משלב לשלב. ו h_{t+1} ינסה לא לטעות בנקודות ש h_t טעה בהן.

- **כלל ההחלטה:** ניתן משקל לכל עץ החלטה, ונקבל החלטה לפי ממוצע העצים על כל נקודה.

- **מימוש עם משקולות -** $weighted\ boostrap$**:** אם אנו לא יכולים להריץ את האלגוריתם שלנו עם משקולות, נגריל את הנקודות של סט האימון עם החזרות ועם משקולות.
עיקרון ERM **ממושקל:** $L_{S,D^*}(h) = \sum_{i=1}^{m} D_i^i 1_{[y_i \neq h(x_i)]}$ אם נעבוד עם אלגוריתם שעובד עם ERM נחפש את ERM הממוצע וננסה למנם אותו.

- **אלגוריתם עם משקולות:** אם יש לנו אלגוריתם שניתן להריץ אותו עם משולות, אזי נריץ אותו ישר עם משקולות מבלי להפריד למספר סטים של אימון.

- **מטא אלגוריתם** $Adaboost$: נתאר מהי ההתפלגות D^1, אח"כ נתאר מהו כלל העדכון של D^t, ואת המשקולות לפיהן נעדכן את כלל ההחלטה.

1: נתחיל עם $D^1 = uniform$.

2: נעדכן כך -

$$D_i^{t+1} \longleftarrow \frac{D_i^t \cdot e^{-w_i y_i h_t(x_i)}}{\sum_{j=1}^m D_j^t \cdot e^{-w_t y_j h_i(x_j)}}$$

3: המשקולות w_t יוגדרו כך שיקיימו את המשוואה -

$$\sum_{i=1}^m D_i^{t+1} 1_{[y_i \neq h_t(x_i)]} = \frac{1}{2}$$

כלומר, נעדכן את המשקולות כך שכאילו אנו לא יודעים כלום, כי 0.5 זאת החלטה ניטרלית.

וההחלטה תתקבל לפי w_t שמסמן את החשיבות של h_t:

$$h_{\text{boost}}\left(x\right) := sign\left(\sum_{t=1}^T w_t h_t(x)\right)$$

הנוסחה הסגורה ל w_t **היא:** עבור $\varepsilon_t = \sum_{i=1}^m D_i^t 1_{[y_i \neq h(x_i)]}$

$$w_t := \frac{1}{2}\log\left(\frac{1}{\epsilon_t} - 1\right)$$

- **פסואודו קוד:**

Algorithm 1 Adaptive Boosting

```
procedure ADABOOST(training set S = {(x_i, y_i)}_{i=1}^m, base learner A, number of rounds T)
    Set initial distribution to be uniform: D^(1) ← (1/m, ..., 1/m)        ▷ Initialize parameters
    for t = 1, ..., T do
        Invote base learner h_t = A(D^(t), S)
        Compute ε_t = Σ D^(t) 1_[y_i ≠ h_t(x_i)]
        Set w_t = 1/2 ln((1−ε_t)/ε_t) = 1/2 ln(1/ε_t − 1).
        Update sample weights D_i^(t+1) = D_i^(t) exp(−y_i · w_t h_t(x_i)),   i = 1, ..., m
        Normalize weights D_i^(t+1) = D_i^(t+1) / Σ_j D_j^(t+1)   i = 1, ..., m
    end for
    return h_S(x) = sign(Σ_{i=1}^T w_t h_t(x))
end procedure
```

- **כיצד נאחסן:** נשמור את h_t עם סט האימון ואת המשקולות.
- **הגדרה - לומד חלש:**

Definition 1.1 A learning algorithm $\mathcal{A}$ is a *γ-weak-learner* for a hypothesis class $\mathcal{H}$ if there exists a function $m_{\mathcal{H}} : (0,1) \to \mathbb{N}$ such that:
- For every $\delta \in (0,1)$
- For every distribution $\mathcal{D}$ over $\mathcal{X}$
- For every labeling function $f : \mathcal{X} \to \{\pm 1\}$

if the realizable assumption holds with respect to $\mathcal{H}, \mathcal{D}, f$, then when running the learning algorithm on $m \geq m_{\mathcal{H}}(\delta)$ *i.i.d* samples generated by $\mathcal{D}$ and labeled by f, the algorithm returns a hypothesis h_S such that
$$\mathbb{P}\left(L_{\mathcal{D},f}(h_S) \leq \tfrac{1}{2} - \gamma\right) \geq 1 - \delta$$

- $Bias - variance$**:** מחלקת ההיפותזות H_T תגדל כאשר T גדל, אך היא לא תגדל ממש מהר. לכן ה $bias$ יירד כי אנו מגדילים את מחלקת ההיפותזות, אך מכיוון שאנו בוחרים משקולות באופן שקול אזי ה var גדל אך לא יותר מידיי.
- **הערה:** אם נגדיל את T יותר מידיי ה var תעלה אך לא תתפוצץ.
- **מתי נעשה** $Boosting$**:** לא נעשה לכללי החלטה מורכבים יותא מידיי, לדוגמה עבור עצים נקח עצים רדודים.
אין טעם לקחת מחלקת היפוטזות מורכבת, כי עם $Boosting$ אנו מגדילים את מחלקת ההיפותזות מבלי להעלות את ה var יותר מידיי.

7.2.5 סיכום:

- **השוואה** $Boosting\ VS\ Bagging$:

Comparing Bagging and Boosting

	Bagging	Boosting
Learns members:	in parallel	sequentially
Training sample for members:	bootstrap training samples	weighted bootstrap or original S with weighted ERM
De-correlation:	recommended	not necessary
When T is too large	Does not overfit	may overfit
Improvement:	reduces variance	reduces bias
With decision trees, use:	deep trees	shallow trees
Parallel implementation:	yes - easy	no
Committee vote:	unweighted	weighted

8 שבוע 8:

8.1 תרגול 8 - $Boosting$:

- **פונקציות** $loss$: אמרנו כי בהינתן D - כל הדאטה בעולם נמדוד את השגיאה כך:

$$h^* = argmin_h L_D(h)$$

אך מכיוון ש D אף פעם לא ידוע לנו נמדוד לפי סט אימון S:

$$h_S = argmin_h L_S(h)$$

ומתקיים:

$$L_D(h_S) = \underbrace{L_D(h^*)}_{\varepsilon\ aproctimation=bias} + \underbrace{L_D(h_S) - L_D(h^*)}_{\varepsilon\ estimation=var}$$

החלק הראשון מייצג את ה $bias$ והחלק השני ייצג את ה var כי $L_D(h^*)$ מייצג את הממוצע.

- **הגדרה - לומד חלש** $(weak\ learner)$:

Definition 1.1 A learning algorithm $\mathcal{A}$ is a *γ-weak-learner* for a hypothesis class $\mathcal{H}$ if there exists a function $m_{\mathcal{H}} : (0,1) \to \mathbb{N}$ such that:
- For every $\delta \in (0,1)$
- For every distribution $\mathcal{D}$ over $\mathcal{X}$
- For every labeling function $f : \mathcal{X} \to \{\pm 1\}$

if the realizable assumption holds with respect to $\mathcal{H}, \mathcal{D}, f$, then when running the learning algorithm on $m \geq m_{\mathcal{H}}(\delta)$ *i.i.d* samples generated by $\mathcal{D}$ and labeled by f, the algorithm returns a hypothesis h_S such that

$$\mathbb{P}\left(L_{\mathcal{D},f}(h_S) \leq \tfrac{1}{2} - \gamma\right) \geq 1 - \delta$$

- **הערה:** נשים לב שמתקיים

$$m_{H_{weak}}(\delta) = m_{H_{pac}}(0.5 - \gamma, \delta)$$

- **טענה:** אם אלגוריתם הוא לומד חלש אזי הוא גם למיד PAC.
- **נגדיר את הסיווג:** עבור $learners$ הבאים: $X_1 ... X_T \sim Ber(p)$ שמקבלים ערכים $\{-1, 1\}$ (כאשר 1 מגדיר חיזוי נכון, ו 1- שגיאה), ונגדיר $X = \Sigma_{i=1}^T X_i$, והפרדיקציה תהיה כך - $sign(X)$.
כלומר כל אחד מהם צודק בהסתברות p.
- **טענה:** בהינתן שהפרדיקציה הנכונה היא 1, ההסתברות של הוועדה לצדוק היא $P(X > 0)$.

- **טענה:** נרצה לחזום מלמטה את ההסתברות לתשובה נכונה ע"י $1-\exp\left(-\frac{T}{2p}\left(p-\frac{1}{2}\right)^2\right)$.
נשים לב כי ככל ש T גדל ההסתברות גדלה ונשאף ל 1.

- **התוחלת של X למ"מ ללא קורלציה בניהם:** ועבור X שהגדרנו ועבור מ"מ X_i שמתפלגים ברנולי עם $p>0.5$ מתקיים: $E(X)=2p-1$.

- **השונות של X למ"מ ללא קורלציה בניהם:**

$$var(X)=\frac{4p(1-p)}{T}$$

- **נשים לב:** התוחלת לא תלויה במספר ה $learners$ שיש לנו, אך השונות כן, והיא תקטן ככל ש T גדל.

- **הגדרה - קורלציה:** נגדיר קורלציה בין שני מ"מ

$$corr(X_i,X_j)_{i\neq j}=\rho$$

ונגדיר את השונות שלהם להיות: $var(X_i)=\sigma^2$.

- **נחשב את ה cov:** $cov(X_i,X_j)=\rho\cdot\sigma^2$.

- **נחשב את השונות של X עבור מ"מ עם קורלציה בניהם:**

$$var(X)=\rho\cdot\sigma^2+\frac{1}{T}(1-\rho)\cdot\sigma^2$$

כלומר - אם נקח $learners$ שדומים (ישבניהם קורלציה) אנו לא נלמד בצורה טובה כי ה var יהיה גבוה (כי אנו תלויים ב ρ).

8.1.1 האלגוריתם $ADABOOST$:

- **הרעיון:** כל $learner$ יקח את המסקנות של ה $learner$ הקודם ומשפר אותו.

- **קלט:** האלגוריתם מקבל סט דגימות ומשקולות על דגימות, בכל שלב נעדכן את המשקולות של הדגימות שטעינו עליהן להיות גבוהים יותר. כך כשהאלגוריתם יחשב את ה $loss$ הוא יתן משקל גבוה יותר לנקודות האלו.

- **הפלט:** עבור כל חיזוי h_t נכפול אותו במשקל שלו w_t, נחשב את הסכום שלהם ונחזיר את הסימן.

8.1.2 $Bagging$:

- **הרעיון:** יש לנו דאטה סט בגודל m, ואנו רוצים לעשות עליו מניפולציה כך שיהיו לנו כמה סטים של דאטה.
נגדיר סט חדש $S^{*j}=\left\{\left(x_i^{*j},y_i^{*j}\right)\right\}_{i=1}^m$ כך שכל דגימה $\left(x_i^{*j},y_i^{*j}\right)$ נדגמת מתוך הסט המקורי S עם חזרות. כך שנקבל B סטים שיש בניהם ובתוכם חפיפה.

- **כיצד נלמד:** נסכום את כל הפרדיקציות ונקח את הסימן.

$$h_{\text{bag}}(x) = sing\left(\sum_{j=1}^{B} h_s^{*j}(x)\right)$$

- **התיאוריה:** למעשה אנו אומרים כי אם נקח m ממש גדול, אזי ההתפלגות של הזוגות בסט S^* - $\hat{D_{S^*}}$, תהיה דומה להתפלגות בסט S המקורי - D_S.

8.2 הרצאה 8:

- אין הרצאה.

9 שבוע 9:

9.1 תרגול 9 - תרגול חזרה:

- **הערה:** בד"כ יש קשר ישיר בין כמות הפרמטרים שמגדירים לנו את מחלקת ההיפותזות לבין $VC\ dim$.

דוגמה: מחלקה שמוגדרת ע"י שני מלבנים, מוגדרת ע"י 8 פרמטרים (2 נקודות לכל מלבן, וכל נקודות מוגדרת ע" x, y). ולכן מימד VC שלה גדול שווה ל 8 .

9.2 הרצאה 9 - רגולריזציה (11.5):

9.2.1 עיקרון הרגולריזציה ורגרסיה לינארית:

- **סט האימון:** X הוא המרחב האוקלידי ה d מימדי.

- **רגולריזציה:** עיקרון המאפשר לנו ליצור משפחה רציפה של לומדים A_λ שמשתמשים באותה מחלקת היפותזות, עם פרמטר הרגולריזציה λ. ניתן להכיל את העיקרון על מסווגים ועוד, אנחנו נתמקד ברגרסיה.

- **הרעיון:** נניח שיש לנו אלגוריתם A_0שיודע לבחור היפותזה לפי העיקרון הבא - מינימזציה של פונקציית מחיר F. ניתן להפעיל את השיטה על כל אלגוריתם למידה שממנמם משהו - $loglikelihood, max\ margin, ERM$.

מה שאנו מנסים לעשות זה להקטין את ה var לא ע"י הקטנה מחלקת ההיפותזות, אלא ע"י הגבלת חופש הפעולה של האלגוריתם.

- **הפונקציה** $F(h)$: תיקרא $fidelity\ term$, ונרצה שהיא תהיה מינימלית.

- **נגדיר** ERM: עבור l פונקציית מחיר נמדוד כך -

$$L_S(h) = \frac{1}{m}\sum_{i=1}^{m} \ell\left(h\left(x_i\right), y_i\right)$$

- **נגדיר את בעיית האופטימיזציה מחדש:** אם H גדולה מידיי אנו חוששים מ $over\ fit$, לכן נלמד באופן אחר. נגדיר $\lambda \geq 0$ והלומד שלנו $A_\lambda : S \mapsto h_S$ יחפש את המינימום הבא

$$h_S := argmin_{h\in\mathcal{H}}\left[\mathcal{F}_s(h) + \lambda\mathcal{R}(h)\right]$$

למעשה ככל ש λ יגדל, האלגוריתם יתעלם מ F וינסה למנמם את R, ומכיוון ש R לא תלויה ב S אלא בהיפותזה h, ה $variance$ שלנו יקטן, וה $bias$ שלנו יגדל.

- **הפרמטר λ והפונקציה R:** הפונקציה R תגדיר את מורכבות האלגוריתם, והפרמטר λ יגדיר את העונש למורכבות של מודל.

- **פונקציה R:** נקראת $regularization\ term$, והיא תמדוד את רמת המורכבות של היפותזה h, וככל שהמחלקה יותר מורכבת R תגדל.
אנו נחפש את המינימום על $\mathcal{F}_s(h) + \lambda\mathcal{R}(h)$ כך מתנהל בניהם טריידאוף שמוגדר ע"י λ.

- **מספר אלגוריתמי הלמידה:** למחלקה יש $\aleph$ אלגוריתמי למידה.

9.2.2 גידול וגיזום עצי רגרסיה:

- **גידול העצים:** בדומה לעצי סיווג, רק שכאן נמדוד את הטעות כך:

$$argmin_{c\in\mathbb{R}}\sum_{i=1}^{k}\left(y_i - c\right)^2$$

- **גיזום העץ:** נרצה לגזום את העץ בכדי להפחית את ה var, נעשה זאת ע"י הורדת פיצולים וענפים.

- **כיצד נגזום:** נקח את ER של כל העץ - T. עבור N קופסאות, ועבור $\hat{y}_s$ הממוצע האמפירי של הנקודות ב y.

$$L_S(T) = \sum_{j=1}^{N}\sum_{i:x_j\in B_j}\left(y_i - \hat{y}_s\left(B_j\right)\right)^2$$

- **הגדרה - תת עץ:** נגדיר את העץ לאחר הגידול להיות T_0, נאמר כי עץ גזום $T \subset T_0$ אם T מתקבל מ T_0 ע"י הורדת קופסאות.

- **הפונקציה R בעצים:** נגדיר את הפונקציה R להיות הפונקציה שסופרת את מספר הקופסאות בעץ T (מספר העלים).

- **בעיית הרגולריזציה:** נעבור על מרחב כל תתי העצים של T_0 ונחפש את המינימום הבא -

$$\min_{T\subset T_0}\left[L_S(T) + \lambda\cdot|T|\right]$$

הרעיון הוא: כך נוכל לפצל את הדאטה ואם הוא רועש בחלק מסויים של העץ נגזום אותו, ונוריד את ה var.

- **אלגוריתמי רגרסיה מודרנית:** בשונה מרגרסיה לינארית -
לא נעלים את הפרמטר ההיסט w_0 (לא נכניס אותו לתוך המ"פ), אלא נשאיר אותו כמו שהוא נתון.
בנוסף נאפשר עבודה עם דאטה כך ש $d >> m$ (מספר הפיצ'רים גדול ממש ממספר הדגימות).

9.2.3 האלגוריתם $Best\ Subset$:

- **פונקציית ה $Loss$:**

$$L_s(\mathbf{w}) = \|w_0 1 + X\mathbf{w} - \mathbf{y}\|^2$$

- **נגדיר את בעיית האופטימיזציה:**

$$\begin{array}{ll} minimize_{w_0\in\mathbb{R}, w\in\mathbb{R}^d} & \|w_0 1 + xw - y\|^2 \\ \text{subject to} & \|w\|_0 \leq t \end{array}$$

עבור נורמת ה 0 מוגדרת באופן הבא: מספר הכניסות של הווקטור ששונות מ 0, (הדלילות של הווקטור).

$$\|v\|_0 = \#\left\{i \mid v_i \neq 0\right\}$$

כלומר: נסרוק על כל תתי הקבוצות בגודל t פיצ'רים (מתוך d סה"כ), ונפעיל על כל קבוצה את הרגרסיה הלינארית, ונבחר את הקבוצה עם $loss$ הקטן ביותר - כך נבחר את הפיצ'רים הטובים ביותר מבין כל הפיצ'רים.
החסרון: הבעיה היא $NP - hard$ כי יש לנו $\binom{t}{d}$ קבוצות. לכן נרצה למצוא בעיה קרובה אליה וקלה לפתון.

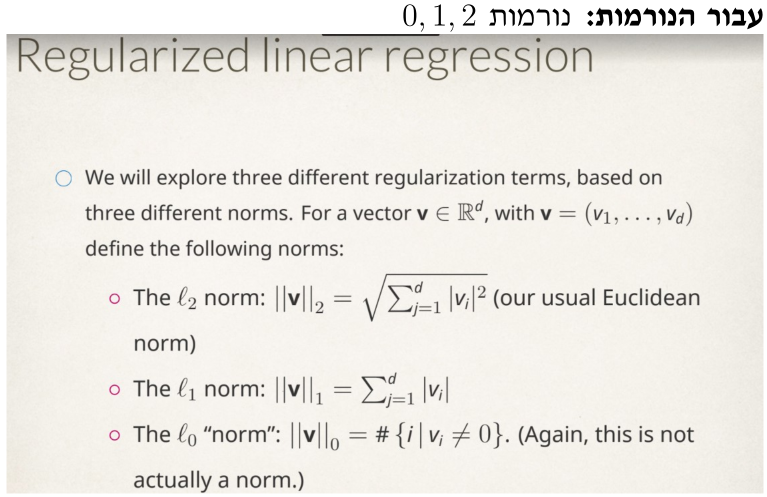
- **מכיוון שהבעיה קשה ננסה למצוא את בעיית האופטימיזציה הבאה:** ללא אילוץ על הנורמה

$$argmin_{w_0\in\mathbb{R}, w\in\mathbb{R}^d}\left[L_s\left(w_0, w\right) + \lambda R(\mathbf{w})\right]$$

הבעיה כעת קמורה ולכל t מתאים λ ולהיפך. כאשר $R(\mathbf{w}) = ||\mathbf{w}||$.
ועבור פונקציית L_S הבאה: $L_S\left(w_0, w\right) = \sum_{i=1}^{m}\left(w_0 + \langle w, x_i\rangle - y_i\right)^2$

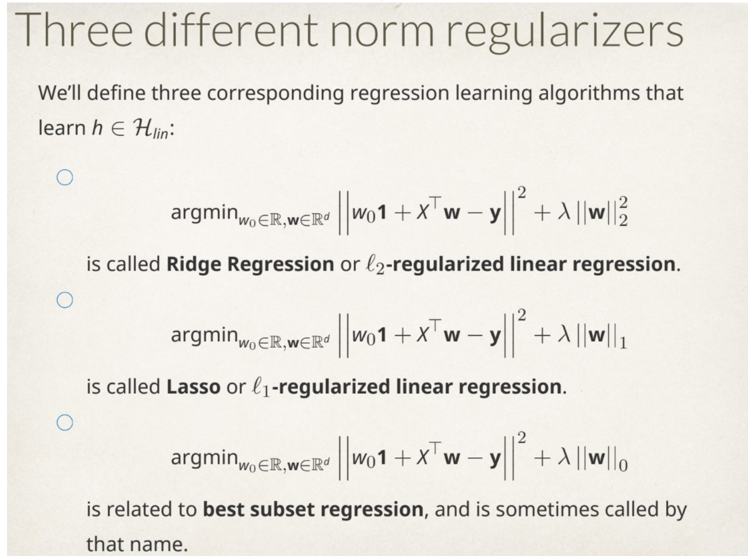
- **ניתן להגדיר שלשה אלגוריתמים עבור שלש נורמות שונות.**

עבור הנורמות: נורמות $0, 1, 2$



נגדיר שלשה אלגוריתמים:

הערה: יש טעות בנוסחת $argmin$, X צריך להיות רגיל ולא X^T



- **דמיון ושוני בין האלגוריתמים:**
  עבור $\lambda = 0$: כל האלגוריתמים יהיו עם רגרסיה לינארית.
  עבור $\lambda = \infty$: כולם יחזירו את אותה היפותזה - מוצע הנקודות על y, וה var יהיה 0 .
  עבור $0 < \lambda < \infty$: יש הבדל בין האלגוריתמים כפי שנפרט להלן.

- **באלגוריתם** l_0: λ יגדיר לנו בכמה פיצ'רים השתמשנו, בעיה זו היא גם $NP - hard$ אך הא מקרבת לבעיה המקורית, ואפשר להשתמש בה כשיש לנו מעט פיצ'רים.

- **באלגוריתם** $l_2 - Ridge$: בעיית אופטימיזציה קמורה, האלגוריתם ינסה למזער את המשקולות, וכאשר λ לא גדול מידיי שהשגיאה שלנו תשתפר, (עד טווח מסויים של λ, ואחכ היא תתחיל לעלות) ככל ש λ ישאף לאינסוף, המשקולות ישאפו ל 0.
  באופן שקול: כאשר נגזור את בעיית האופטימיזציה המקורית נגיע לכך כי הפתרון נמצא בפתרון מערכת המשוואות הבאה:
  $$X^\top y = \left(X^\top X + \lambda I\right) w$$
  כאן למעשה λ עוזר לנו לפתור את מערכת המשוואות ביותר קלות כי המטריצה $\left(X^\top X + \lambda I\right)$ תהיה הפיכה.
  ובאופן שקול: עבור Σ_i מטריצת הערכים הסינגולרים של X עם ערכים סינגולרים σ_i, אזי הפתרון האופטימלי הוא
  $$\hat{w}_\lambda^{\text{ridge}} = U^T \Sigma^\lambda V \mathbf{y}$$
  כאשר Σ^λ היא מטריצה שיש לה על האלכסון הראשי ערכים ששווים ל $\frac{\sigma_i}{\sigma_i^2+\lambda}$.
  כאן הפרמטר λ יעזור לנו במקרה ש $\sigma_i = 0$.

- **האלגוריתם** $l_1 - Lasso$: הוא ישלח חלק מהפיצ'רים ל - 0 (לא בהכרח שהם ישארו תמיד עם משקל 0). הפרמטר λ יגדיר לנו מי הפיצ'רים החשובים, ככל שנתקרב לאינסוף נוריד יותר פיצ'רים - ניתן להם משקל 0.
  כלומר הפרמטר λ יגדיר לנו איזה פיצ'רים להוריד. האלגוריתם הוא $interpetable$.

* **משפט - במקרה ש** $X^TX=I$ - $ortogonal\ desine$: כאשר הפצ'רים מאונכים אחד שני ו X מטריצה אורתוגונלית יתקיים עבור הפונקציות הבאות:

$$\eta_{\lambda}^{soft}(x)=\begin{cases}x-\lambda & x\geq\lambda\\ 0 & \lambda>x>-\lambda\\ x+\lambda & -\lambda\geq x\end{cases}$$

ו

$$\eta_{\lambda}^{\text{hard}}(x)=x\cdot 1_{|x|\geq\lambda}$$

מתקיים כי הפתרון עבור המקרה האורתונורמלי שווה להפעלת הפונקציות הנ"ל על כל קאורדינטה :

$$\begin{aligned}\hat{\mathbf{w}}_{\lambda}^{\text{ridge}} &= \hat{\mathbf{w}}^{LS}/(1+\lambda)\\ \hat{\mathbf{w}}_{\lambda}^{\text{lasso}} &= \eta_{\lambda}^{\text{soft}}\left(\hat{\mathbf{w}}^{LS}\right)\\ \hat{\mathbf{w}}_{\lambda}^{\text{subset}} &= \eta_{\sqrt{\lambda}}^{\text{hard}}\left(\hat{\mathbf{w}}^{LS}\right)\end{aligned}$$

* **רגולריזציה לרגרסיה לינארית:** נקראת l_1-regularized logistic regression, ומוגדרת ע"י:

$$argmin_{w\in\mathbb{R}^d}\left[\sum_{i=1}^{m}\left[\log\left(1+e^{w_0+\langle x_i,w\rangle}\right)-y_i\left(w_0+\langle x_i,w\rangle\right)\right]+\lambda\|w\|_1\right]$$

* **הספריה** $GLMNET$: ספריה שמממשת את l_1-regularized logistic regression ואת $lasso$.

9.2.4 כיצד נבחר את היפר הפרמטר λ, ונעריך מודל - $model\ selection$:

* **מה נרצה לדעת:** איזה אלגוריתם נבחר מבין כל האלגוריתמים, כיצד נדע מי מהם הטוב ביותר עבור הטסט סט. או כיצד נבחר את היפר הפרמטר (לדוגמה - עומק עץ, פרמטר k באלגוריתם KNN, פרמטר λ באלגוריתם $SOFT-SVM$).

* **הרעיון:** עבור n מחלקות היפותזות, נבחר לכל מחלקה H_i את הפונקציה h_i האופטימלית ($loss$ מינימלי), ונרצה לבחור מתוך כל ה h_i את הטובה ביותר. נבחר טסט סט שאף אחת מהן לא נחשפה אליו ונבחר את האחת שמחזירה לנו L_V- $loss$ על סט הולידציה הנמוך ביותר.

* **מה לא נעשה:** להריץ את כל האלגוריתמים ולבחור את האלגוריתם עם ה $Loss$ המינימלי, לא בהכרח יביא לנו את האלגוריתם הטוב ביותר, כי יכול להתרחש $over\ fit$.
לכן נשאף לאמן כל אחד מהמודלים על דאטה שונה וחדש, הבעיה היא שלא תמיד יש לנו מספיק דאטה כדי לאמן את האגוריתמים.

- **השיטה הנאיבית - k-fold Cross Validation :**
מקבל כקלט: סט אימון - S, אלגוריתם לומד - A, פרמטר שיגדיר את הפיצול - k, ופרמטרים - Θ.
נקח את ה $train\ Set$, ונפצל אותו ל k קבוצות, לדוגמה עבור $k=4$ - נאמן כל אלגוריתם על $3/4$ מהסט, ונבדוק אותו על $1/4$. ובכל פעם נבחר חלק אחר להיות ה $1/4$ של הטסט דאטה עליו נחשב את ה $loss$. לבסוף נקח ממוצע של השגיאות ונבחר את האלגוריתם עם השגיאה הממוצעת הנמוכה ביותר.
יתרון: ממוצע השגיאות מפחית לנו את ה var.
חסרונות: אם חלק מהדאטה יהיה "גרוע" אנו נאמן את המודלים על דאטה גרוע שלא יחזה לנו כמו שצריך - תהיה תלות (קורולציה גבוהה) בין המשתנים ולכן הממוצע לא תהיה שיטה אפקטיבית לשיערוך - var גבוה.
חסרון נוסף - אנו צריכים לאמן את המודל כמה פעמים.
בנוסף - אנו כל פעם מאמנים רק על חלק מהדאטה - סט האימון קטן יותר.

האימון הסופי: לאחר שבחרנו את האלגוריתם הטוב ביותר, נאמן אותו על כל הדאטה שיש לנו.

- **פסואודו קוד ל** $k-fold\ Cross\ Validation$:

Algorithm 10 Cross-Validation
```
procedure k-FOLD-CROSS-VALIDATION(S, k A_α)
    Randomly partition S to k disjoint subsets S = ⊎_{i=1}^k S_i.
    for i = 1,...,k do
        Train the model S except the i'th fold S \ {S_i}.
        Calculate the loss of the model on S_i functioning as a test set.
    end for
    return the estimated mean and standard deviation of the k losses obtained.
end procedure
```

- **התנהגות הדאטה:** ככל שנאמן את המודל על יותר דאטה טווח השגיאה יירד, אך החל משלב מסויים השיפור ל האלגוריתם ייפחת וטווח הטעות יקטן אך לא באופן משמעותי. לכן אין טעם תמיד לאמן על כמה שיותר דאטה, אלא על דאטה שיביא לנו את השיפור האפקטיבי בטעות.

- **איך נבחר את** k:
הבחירה של $k=2$ נקראת $split\ sample$.
עבור $k=m$, השיטה נקראת $leave\ one\ out$ ונצטרך להריץ את האלגוריתם m פעמים. החסרון - יהיה מלא רעש.
לכן נצטרך לבחור k בניהם.

- **שימוש ב** $Bootstrap$: ניתן לעשות הערכת מודל בשימוש באלגוריתם $Bootstrap$ כדי ליצור דאטה חדש.

9.2.5 כיצד נעריך ביצועים עתידיים $model\ evaluation$:

- **הערכת ביצועים:** לאחר שבחרנו את המודל ואת הפרמטר נרצה להעריך את הביצועים של כל h_i. נעשה זאת ע"י דגימת סט חדש שאף אחת מהן לא נחפשה עליו ונראה מי חזתה לנו טוב ביותר. נסתכל על h_i עם ה var הנמוך ביותר. למעשה ה var מגדיר לנו כמה נוכל להעריך טוב בהינתן דאטה חדש, כלומר כמה האלגוריתם למד לדאטה כללי ולא נקבע לפי דאטה ספציפי.

- **שימוש ב** $Bootstrap$: ניתן לעשות אבלואציה בשימוש באלגוריתם $Bootstrap$ כדי ליצור דאטה חדש.

10 שבוע 10

10.1 תרגול 10 ־ רגולריזציה (15.5):

- **הרעיון:** עד עכשיו ידענו באופן דטרמניסטי מהי מחלקת ההיפותזות וקבענו אותה מראש. העיקרון של רגולריזציה ־ לא נקבע מראש מהי מחלקת ההיפותזות, אך ניתן לו את ההיוריסטיקה הבאה: תתן עדיפות למודלים פשוטים, אך אם יש מודל ממש טוב (פונקציית $loss$ נמוכה) עם מורכבות גבוהה יותר תקח אותו.
נעשה את זה באופן הבא: נחפש את ההפסד המינימלי (F), אך נוסיף פרמטר ־ R שמגדיר עונש על מורכבות המודל. ואת המשקל למודל המורכב נגדיר באמצעות λ שנקבע מראש.

$$h_S := argmin_{h\in\mathcal{H}}\left[\mathcal{F}_s(h)+\lambda\mathcal{R}(h)\right]$$

- **האלגוריתם** $Ridge$: עם נורמה l_2, נגדיר את הפתרון ־ W להיות:

$$\hat{\mathbf{w}}_\lambda^{\text{ridge}} := \underset{\mathbf{w}}{argmin} RSS(S;\mathbf{w})+\lambda\|\mathbf{w}\|_2^2$$

עבור RSS ־ סכום המרחקים המרובעים $sum\left((y_{pred}-y_{true})^2\right)$.
כלומר הפונקציה R תהיה $\|\mathbf{w}\|_2^2$, והיא מודדת את מורכבות המודל לפי מספר הפיצ'רים בו.

- **באופן שקול על** $Ridge$: נרצה לפתור את הבעיה כבעיית רגרסיה לינארית עם חישוב המינימום על ה $loss$. ניתן להגדיר מטריצות חדשות ולחשב את בעיית רגרסיה לינארית באופן הבא:

Exercise 2.1 Let $\mathbf{X},\mathbf{y}$ be a design matrix and response vector. We will show that we can consider ridge regression as an ordinary least squares (OLS) problem over an augmented dataset. Denote the following:

$$\mathbf{X}_\lambda=\left[\frac{\mathbf{X}}{\sqrt{\lambda}I_d}\right]\in\mathbb{R}^{(m+d)\times(d)},\quad \mathbf{y}_\lambda=\left[\frac{\mathbf{y}}{0_d}\right]\in\mathbb{R}^{m+d}$$

Then the LS solution for $\mathbf{X}_\lambda,\mathbf{y}_\lambda$ is:

$$\hat{\mathbf{w}}_\lambda^{LS}=\left(\mathbf{X}_\lambda^\top\mathbf{X}_\lambda\right)^{-1}\mathbf{X}_\lambda^\top\mathbf{y}_\lambda=\mathbf{X}_\lambda^\dagger\mathbf{y}_\lambda$$

כלומר ־ במקום להתחשב בפרמטר הרגולזציה בפונקציית המינימום ולהביא גם אותו למינימום, אנו משנים את X,y ומכניים את העונש לסט. לאחר מכן אנו דורשים $loss$ כמו ברגרסיה לינארית.

- **טענה על** $Ridge$: האלגוריתם הזה יותר יציב נומרית עם הפרמטר λ, כי הוא מרחיק את הערכים העצמיים מ 0.

- **תכונות של נורמה** $q\leq 1$: תתקיים דלילות(ווקטור שרוב כניסותיו שוות ל 0) ־ נעודד פתרונות שהמשתנים מתאפסים בהם.

- **תכונות של נורמה** $q<1$: הפונקציה אינה פונקציה קמורה.

- **הערה:** ב $Ridge$ מתקיים כי הפונקציה אינה קמורה ויש דלילות.

- **האלגוריתם** $Lasso$: עם רגולריזציה l_1

$$\underset{\mathbf{w}}{argmin} f_{\ell_0}(\mathbf{w}) := \underset{\mathbf{w}}{argmin}\|\mathbf{y}-\mathbf{X}\mathbf{w}\|_2^2+\lambda\|\mathbf{w}\|_0$$

10.2 הרצאה 10 - *unsupervised learning* (18.5):

- **למידה שאינה מפוקחת:** אין לנו לייבלים - y_i אלא רק דגימות x_i.

- **1** *dimention reduction*: נרצה להוריד את המימד של X, להוריד פיצ'רים. לדוגמה - אם יש לנו כמה תמונות דומות, לא נצטרך לשמור את כל הפיקסלים של כולן, אלא רק תמונה אחת, ועבור כל תמונה נשמור את ההבדלים.
המטרה: נמפה כל נקודה x_i ששייכת ל R^d, לנקודה $W(x_i)$ ששייכת למרחב מימדי נמוך בהרבה מ d.
יתרונות: הבנה, יכולת לעשות ויזואליזציה, איחסון, חישוב ועוד.

- **2** *Clustering*: חלוקת המרחב ל k מחלקות.
דוגמה - נקבל תמונה עם אנשים ונצטרך להבדיל בין אנשים - לספור כמה אנשים שונים יש בתמונה ומי הם.

- **3** *Anomaly detection*: אלגוריתם שמקבל מידע כיצד מערכת מסויימת צריכה לתפקד, ולומד אותו. ברגע שהמערכת משנה את התנהגותה עקב תקיפה או תקלה האלגוריתם יידע להתריע שההתנהגות אינה תקינה. גם מבלי לדעת איפה התרחשה התקיפה או מה לא תקין במערכת.

10.2.1 הורדת מימד - *dimention reduction*:

- **הגדרה:** נתונות נקודות $x_1...x_m \in R^d$, נחפש העתקה לינארית $W : R^d \Rightarrow R^k$ שמורידה את המימד מ d ל k (נבחר את k מראש). נרצה שיתקיים $x_1...x_m$ יהיו דומים ל $W(x_1)...W(x_m)$.
כאשר W ה"ל הורדת המימד נקראת לינארית, כאשר ההעתקה לא לינארית הורדת המימד נקראת "הורדת מימד לא לינארית". אנו נתחיל עם הורדת מימד לינארית.

- **הנחה:** נניח כי הדאטה קרוב יותר להיות k מימדי - הוא יושב בתוך תת מרחב k מימדי, אך הוא כרגע מאוחסן בתוך מימד d.

- **הרעיון:** נקח את הדאטה שנתון לנו, נעשה מניפולציה על הפיצ'רים המקוריים כך שיחזרו לנו פיצ'רים חדשים שאיתם נוכל לסווג טוב יותר את הדאטה, אך הם לא יהיו קשורים לפיצ'רים המקוריים.
פורמלית: נשליך אורתוגונלית את הדאטה על תת המרחב, נמצא בסיס לת"מ (בסיס אורתונורמלי), ונתאר כל נקודה באמצעות קאורדינטות בת"מ. התמונה של ההעתקה תהיה ווקטורים ב R^k שהם קאורדינטות לפי בסיס בתוך ת"מ.

- **האלגוריתם** *PCA*: אלגוריתם למציאת בסיס אורתונורמלי הטוב ביותר לת"מ.
נבחר את k ונחפש ה"ל $W : R^d \Rightarrow R^k$, בנוסף נחפש העתקה הפוכה $U : R^k \Rightarrow R^d$ שתחזיר מ k ל d.
נאמר שהעתקות הלינאריות העתיקו טוב אם ההפרש בין הנקודה המקורית להפעלה של W ואז הפעלת Uעל הנקודה המקורית, מינימלי.

$$\sum_{i=1}^{m} \|x_i - UWx_i\|^2$$

- **הפתרון ל PCA:** נגדיר מטריצה ריבועית סימטרית אי שלילית $A=\sum_{i=1}^{m}\mathbf{x}_i\mathbf{x}_i^{\top}$. ותהי $U\in R^{m\cdot k}$ מטריצה שעבורה העמודות $u_1...u_n$ הם n הווקטורים העצמיים של A מסודרים בסדר יורד. נקח את k הווקטורים הראשונים $u_1...u_k$ ונגדיר את $W=U^T$.

  המטריצות U,W הן המטריצות שחיפשנו לפתרון בעיית PCA.

- **פתרון שקול:** נחפש רק מטריצה U ונמקסם את האיברים שעל האלכסון

$$\{\underset{u\in\mathbb{R}^{d,k}:U^{\top}U=I}{argmax}\ \{trace\left(U^{\top}\left(\sum_{i=1}^{m}\mathbf{x}_i\mathbf{x}_i^{\top}\right)U\right)$$

- **נכליל להעתקה אפינית:** נרצה לאפשר דאטה מכל סוג, ולא רק דאטה שתת המרחב שלו עובר בראשית. נגדיר

$$W(\mathbf{x})=\tilde{W}(\mathbf{x}-\mu)\text{ where }\mu\in\mathbb{R}^{d}\text{ and }\tilde{W}:\mathbb{R}^{d}\rightarrow\mathbb{R}^{k}linear$$

  כעת, המטריצה $A=sample\ cov$ תוגדר כך:

$$A=\sum_{i=1}^{m}\left(\mathbf{x}_i-\bar{x}\right)\left(\mathbf{x}_i-\bar{x}\right)^{\top}$$

  עבור $\bar{x}$ - ווקטור הממוצעים, שכל קאורדינטה בו מייצגת את הממוצע של הפיצ'ר ה i.

- **הגדרה פורמלית של PCA:** נגדיר מטריצת $sample\ cov$

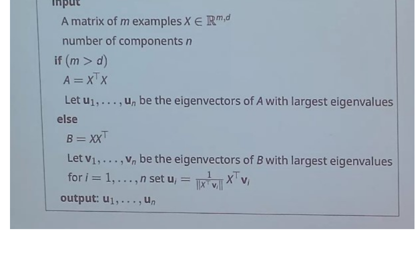
$$S=\frac{1}{m}\sum_{i=1}^{m}\left(\mathbf{x}_i-\bar{x}\right)\left(\mathbf{x}_i-\bar{x}\right)^{\top}$$

  נקח את k הווקטורים העצמיים הראשונים שלה, נגדיר אותם להיות המטריצה U, ונגדיר $W=U^T$ ונחפש פתרון לבעיית המנימיזציה הבאה:

$$argmin_{U,W}\sum_{i=1}^{m}\|x_i-UWx_i\|^2$$

- **הערה:** אחרי הפעלת ההעתקה W אנו נעבור לייצג את הדאטה בתור קאורדינטות בתת המרחב. אם לאחר מכן נפעיל את ההעתקה U לא נקבל את המרחב שהיה לנו, מקודם, אלא נקבל הטלה של המרחב על תת המרחב.

- **כיצד נמצא ו"ע עם חישוב נומרי יציב:** פסואודו קוד ללכסון יעיל.



- **משפט:** אם כל הנקודות יושבו ממש בתוך ת"מ V שגודלו k, אזי $dim(V) = k$.

- **כיצד נבחר את המימד k:** נרצה למצוא את המימד k שיביר לנו את סכום ריבועי ההפסד המינימלי.

נעשה זאת כך: נקח את המטריצה A ונלכסן אותה, נצייר את הע"ע בסדר יורד ב $scree\ plot$. נחליט מהי הנקודה שבה הם מפסיקים לרדת (מתייצבים), ונבחר אותה להיות ה k שלנו.

10.2.2 $Clustering$:

- **הרעיון:** נרצה לסווג את הדגימות ל k מחלקות, נזכור כי אין לנו את ווקטור הרספונס - y.

- **למה זה טוב:**

1: כשנקבל דאטה חדש נוכל לעשות עליו $Clustering$ ולהבין אם הוא מתפלג למחלקות בצורה מסויימת, כך נוכל ללמוד אותו טוב יותר.

2: אם נתונה לנו חלוקה של הדאטה לקבוצות, נוכל לבחון את החלוקה בלמידה בלתי מפוקחת ע" $Clustering$.

- **נגדיר את הבעיה כמותית:** נניח שיש לנו מטריקה על R^d, לדוגמה נורמה. נגדיר $Clustering$ כחלוקה של המרחב ל k קבוצות C זרות

$$\{\mathbf{x}_1, \ldots, \mathbf{x}_m\} = \biguplus_{j=1}^{k} C_j$$

ונגדיר פונקציית מחיר לחלוקה: נמצא מרכז לכל קבוצה C_j, נסכום את ריבועי המרחקים לכל נקודה בכל קבוצה מהמרכז μ_j שלה. אנו ננסה למזער את המרחק מהמרכז עבור כל קבוצה.

$$G(C_1, \ldots, C_k) = \min_{\mu_1, \ldots, \mu_k \in \mathbb{R}^d} \sum_{j=1}^{k} \sum_{\mathbf{x} \in C_j} d(\mathbf{x}, \mu_j)^2$$

איך נבחר את המרכז $\mu = centroid$:

$$\mu_j := \frac{1}{|C_j|} \sum_{x \in C_j} x$$

- **היוריסטיקת** $k - means\ Clustering$**:** היורסטיקה שתמצא לנו קבוצות ואת מרכזן.

הקלט: $x_1 ... x_m$ והיפר פרמטר k.

נבחר בהתחלה $\mu_1 ... \mu_k$ רנדומלים.

הקבוצה C_j יהיו הנקודות שהכי קרובות ל μ_j, נחשב את הממוצע ונעדכן את המרכז והקבוצות לפי הצורך.

תכונות: ה var יירד בכל איטרציה. האלגוריתם תמיד מכנס (בשאיפה לאינסוף). ההתכנסות תלוייה בהתחלה.

- **כיצד נבחר את k:** נסתכל על פונקציית המחיר, ונבחר את ה k בנקודה שבה הירידה מתייצבת ומפסיקה להתייצב.

10.2.3 :*Spectral Clustering*

- **מה נרצה להשיג:**
נרצה להתעלם ממרחקים גדולים (שימושי כשנרצה לסווג קבוצות מעגליות שנמצאות אחת בתוך השניה).
בנוסף נרצה לחלק רק לפי מרחקים בזוגות (שימושי במציאת קהילות ברשתות חברתיות).

- **הרעיון:** נרצה לחלץ מקבוצה מקומית, מידע על הדאטה הכללי. נעשה זאת עם אינטגרציה.
נסתכל רק על המרחקים בזוגות (בין נקודות).
נתעלם ממרחקים גדולים - אם המרחק גדול מידיי נסתכל עליו כאינסוף.
נוריד את הדאטה סט לתוך R^k ע"י שימוש בוקטורים עצמיים של הגרף (פירוק ספקטרלי (לכסון) של מטריצת המשקלים).

- **כיצד נתעלם ממרחקים גדולים:** נבנה גרף ממושקל סימטרי, עם m צמתים - כל דאטה תהיה צומת, והקשתות יגדירו את המרחק.

- **נגדיר מטריצת מרחקים A:** עבור $\varepsilon > 0$, המשקלים על הקשתות בגרף (מרחקים) יוגדרו כך

$$A_{i,j} = \exp\left(-\frac{\|x_i - x_j\|^2}{\varepsilon}\right)$$

A היא מטריצה סימטרית, ונקראת דעיכה גאוסיאנית משום שמרחק קטן יקבל ערך קרוב ל 1, ומרחק גדול יקבל ערך קרוב ל 0.

בנוסף נגדיר מטריצה אלכסונית D: מטריצת הדרגות, סכימה של השורות ב A.

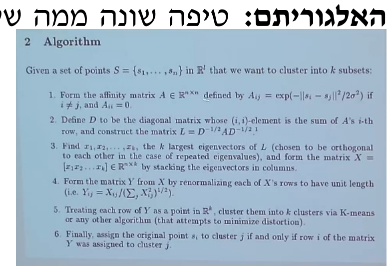
$$D_{i,i} = \sum_{j=1}^{m} A_{i,j}$$

ומטריצה L (סכום כל שורה במטריצה $= 1$):

$$L = D^{-1}A$$

- **כיצד נחלץ את הסיווג:** נמצא את הו"ע המובילים במטריצה L. כל ווקטור יכיל כמה קאורדינטות שערכן יהיה קרוב ל 1, וכל שאר הקאורדינטות ערכן יהיה קרוב ל 0. הסיווג יהיה מחלקה נפרדת עבור כל ווקטור בקאורדינטות שערכן קרוב ל 1.

- **האלגוריתם:** טיפה שונה ממה שלמדנו.



2 Algorithm

Given a set of points $S = \{s_1, \dots, s_n\}$ in $\mathbb{R}^l$ that we want to cluster into k subsets:

1. Form the affinity matrix $A \in \mathbb{R}^{n\times n}$ defined by $A_{ij} = \exp(-||s_i - s_j||^2/2\sigma^2)$ if $i \neq j$, and $A_{ii} = 0$.
2. Define D to be the diagonal matrix whose (i,i)-element is the sum of A's i-th row, and construct the matrix $L = D^{-1/2}AD^{-1/2}$.[1]
3. Find $x_1, x_2, \dots, x_k$, the k largest eigenvectors of L (chosen to be orthogonal to each other in the case of repeated eigenvalues), and form the matrix $X = [x_1 x_2 \dots x_k] \in \mathbb{R}^{n\times k}$ by stacking the eigenvectors in columns.
4. Form the matrix Y from X by renormalizing each of X's rows to have unit length (i.e. $Y_{ij} = X_{ij}/(\sum_j X_{ij}^2)^{1/2}$).
5. Treating each row of Y as a point in $\mathbb{R}^k$, cluster them into k clusters via K-means or any other algorithm (that attempts to minimize distortion).
6. Finally, assign the original point s_i to cluster j if and only if row i of the matrix Y was assigned to cluster j.

- **הערה:** לכסון המטריצה L מדמה אינטגרציה של המידע הלוקאלי, הו"ע הם מבנים גלובלים.

11 שבוע 11:

11.1 תרגול 11 - $unsupervised\ learning$ (22.5):

11.1.1 האלגוריתם PCA:

- **האלגוריתם:** אלגוריתם להורדת מימד מ R^d ל R^k כאשר $k << d$.

Algorithm 1 PCA

```
procedure PCA(X, k)                ▷ X The design matrix of m samples and d features
    Compute A ← Σ_{i=1}^{m} (x_i − x̄)(x_i − x̄)^⊤
    Let u_1, ..., u_k be the eigenvectors of A corresponding the largest eigenvalues.
    return u_1, ..., u_k
end procedure
```

הערה: בשורה 2 צריך לחלק את A ב m.

- **בתרגול:** נסתכל על האלגוריתם ממבט אחר - שימור ה var. מכיוון שאנו מורידים מימד אנו מאבדים מידע, לכן נרצה לאבד כמה שפחות מידע, כלומר למקסם את ה var.
כלומר, קיימת שקילות בין מציאת ההפסד המינימלי, לבין מציאת ה var המקסימלי.

- **משפט:** תהי $X \in R^{m \cdot d}$ מטריצה, ו S מטריצת השונויות. ההטלה של X על ת"מ ממימד k שמשמרת מקסימום var, היא המטריצה $U \in R^{d \cdot k}$ שהעמודות שלה הם k ו"ע שמתאימים לע"ע הגדולים ביותר של S.

- **טענה:** מטריצות אורתונורמלית משמרות נורמה מכיוון שהן רק משרות סיבוב.

- **באופן שקול:** נרצה למצוא מטריצה V כך שהיא אורתונורמלית.

$$\hat{\mathbf{v}}_i := \{\underset{\|\mathbf{v}\|=1 \wedge \mathbf{v} \perp \mathbf{v}_1,\ldots,\mathbf{v}_{i-1}}{argmax} \quad \mathbf{v}^\top S \mathbf{v}$$

- **אינטואיציה:** נסתכל על הציור הבא - אנו רוצים להטיל את הנקודה האדומה על הווקטור u_1. נשים לב לכמה דברים:
$c = ||x_i||$ והוא קבוע.
$b = ||x_i - V^T x_i||$ מייצג את המרחק בין הנקודה המקורית לבין הנקודה המוטלת - מה שראינו בהרצאה.
$a = V^T X$ מייצג את הגודל של ההטלה - מה שראינו בתרגול.
ממשפט פיתגרוס אנו יודעים כי מתקיים שוויון, ולכן מכיוון ש c קבוע, אם נוריד את b, c יעלה ולהיפך.

Figure 2: Link Between PCA Interpretations: Projection of x_i onto u_1 forming a right-angle triangle.

11.1.2 $Spectral\ Clustering$:

- **הרעיון:** נרצה לסווג מחלקות מורכבות יותר ע"י הגדרת מרחק באופן שונה.

- **בניית גרף שכנויות:** נגדיר משתנה $\varepsilon > 0$ כך שכל שתי דגימות שרחוקות יותר מ ε לא ייחשבו כשכנים.
נגדיר גרף כך: קודקודים - $V = \{x_1 ... x_m\}$, וצלעות - $E = \{\{i, j\} \mid ||x_i - x_j|| < \varepsilon\}$, כלומר - נמתח צלע רק אם

המרחק קטן מ ε.

נגדיר מטריצת שכנויות: מטריצה A תוגדר כך (היא מוגרת באופן דטרמיניסטי, בהרצאה ראינו הגדרה לא דטרמניסטית):

$$A_{ij} = \begin{cases} 1 & ||x_i - x_j|| < \varepsilon \ \wedge i \neq j \\ 0 & else \end{cases}$$

בנוסף נגדיר מטריצת דרגות: מטריצה אלכסונית D תייצג את הדרגה של כל קקדקוד ע"י סכימת השורות במטריצה A.

$$D_{ii} = \sum_{j=1}^{m} A_{i,j}$$

נגדיר מטריצת לפלסיאן L**:** מטריצה כך שסכום כל שורה במטריצה שווה ל 0 .

$$L = D - A$$

משיקולי יציבות נומרית נגדיר את הגרף המנורמל L_{sym}**:** ונשתמש בה במקום המטריצה L.

$$L_{sym} = D^{-0.5} L D^{-0.5}$$

$$(L_{sym})_{ij} = \begin{cases} 1 & i = j \wedge deg(i) \neq 0 \\ \frac{-1}{\sqrt{deg(v_i) \cdot deg(v_j)}} & i, j \ neighbors \\ 0 & else \end{cases}$$

- **תכונות של מטריצת לפלסיאן -** L**:**

1: היא מטריצת PSD - הע"ע שלה גדולים שווים 0 .

2: הע"ע הקטן ביותר שווה ל 0, והריבוי הגיאומטרי (מימד של מ"ע שמתאים לו"ע שמתאימים לע"ע 0) שלו שווה למספר רכיבי הקשירות בגרף G.

3: המרחב העצמי של ע"ע 0, נפרש ע"י הווקטורים שהם אינדיקטורים לרכיבי הקשירות (1 רק בקאורדינטות של קודקודים ששייכים לרכיב הקשירות C_j, והשאר אפסים).

11.2 הרצאה 11 - שיטות גרעין $Kernel$ (25.5):

- **הרעיון:** נרצה העתקה לינארית שמעבירה לנו את הווקטור X ממימד d למימד k **גבוה יותר**, לאחר מכן נפעיל מודל לינארי על הוקטור החדש. אנו מתבססים על ההנחה כי תופעות מורכבות הופכות פשוטות יותר לאחר טרנספורמציה (לדוגמה $poly\ fit$).

נבחר העתקה $\psi : \mathbb{R}^d \to \mathbb{R}^k \text{ (with } k > d)$, עבור $X \in R^d$ מתקיים כי $\psi(X) \in R^k$ עם קאורדינטות $\psi(x)_1, \ldots, \psi(x)_k$. נשתמש במודל הלינארי על הפיצ'רים **החדשים**.

- **$Bias\ Var$:** אנו מעשירים את מחלקת ההיפותזות לכן ה $Bias$ יירד, אך ה Var יעלה.

- **$Poly\ fit$:** בפולינומים, העלנו את דרגת המשתנה X והשתמשנו ברגרסיה לינארית. למעשה השתמשנו בשיטת $kernel$. נרצה להכליל ל $X \in R^n$.

- **עבור פולינומים עם $X \in R^d$:** ניצור העתקה ψ שתשלח את X למולטינום.
לדוגמה עבור $X \in R^2$: $\psi(x) = (1, x_1, x_2, x_1^2, x_2^2, x_1x_2)$ ולאחר מכן נפתור בעיית רגרסיה לינארית על הפיצ'רים שחזרו מההעתקה (עבור ווקטור משקולות $w \in R^6$).
במקרה הכללי: עבר פולינום $R^d \Rightarrow R$ נרצה להעביר אותו לפולינום מדרגה R^n עשה זאת כך:

$$p(x) = \sum_{a \in N^d : \sum a_i \leq n} W_a \prod_{i=1}^{d} x_i^{a_i}$$

נסכום על כל הווקטורים a שהסכום שלהם קטן מ n (דרגת הפולינום), את המכפלה של אברי הפולינום כפול המשקל W_a.
לאחר מכן נפתור את בעיית הרגרסיה לינארית הבאה: $\langle w, \psi(x) \rangle$ עבור $\psi : \mathbb{R}^d \to \mathbb{R}^k$.

- **מוטיבציות:**
1: נרצה להעשיר את מחלקת ההיפותזות, ייצוג במימד גבוה יותר יעזור לנו למצוא הפרדה לינארית.
2: יתרון חישובי - אפשר למצוא את ההיפותזה בלי להסתכל על R^k (המימד החדש) אלא רק על הפרשים בין m הדגימות.
3: ניתן לבנות מודל שמסתכל רק על מרחקים בין נקודות ולא על הפיצ'רים (שימושי לדאטה שאין לו פיצ'רים אלא מרחקים - דאטה לא אוקלידי).

- **מחלקות ההיפותזות:**
לרגרסיה לינארית:

$$\mathcal{H}_\psi = \left\{ \mathbf{x} \mapsto \langle \mathbf{w}, \psi(\mathbf{x}) \rangle \mid \mathbf{w} \in \mathbb{R}^k \right\}$$

לקלסיפיקציה:

$$\mathcal{H}_\psi = \left\{ \mathbf{x} \mapsto sign(\langle w, \psi(\mathbf{x}) \rangle) \mid \mathbf{w} \in \mathbb{R}^k \right\}$$

- **קרנל טריק - $The\ Kernel\ Trick$:** נגדיר את התבנית הבאה שתגדיר לנו כיצד ללמוד, כל בעיית למידה שניתן לכתוב לפי התבנית הזו יש לה פתרון יעיל לאיך למצוא את ההיפותזה ב $kernel$.

$$w_S = \underset{w}{argmin} f\left(\left\langle w, \psi\left(\mathbf{x}_1\right) \right\rangle, \ldots, \left\langle w, \psi\left(x_m\right) \right\rangle \right) + \lambda \|w\|^2$$

עבור $W \in R^k$, הנורמה האוקלידית ופונקציה $f : R^m \Rightarrow R$. נגדיר פונקציה f כרצונינו והאלגוריתם הלומד יבחר h_s לפי התבנית הנ"ל.
ה: הרבה מאלגוריתמי הלמידה שלמדנו עד כה נופלים לתוך התבנית הזו ($hard\ soft\ SVM, half\ space, logistic\ regression...$).

- **תבנית שקולה (ללמידה):** נרצה כעת למצוא ווקטור שימשקל לנו את הדגימות (לאחר הפעלת ψ) ולא את הפיצ'רים. נבחר ווקטור $\alpha \in \mathbb{R}^m$ (בעיה זו מעל $\mathbb{R}^m$, בשונה מבעיה קודמת שהיא מעל $\mathbb{R}^k$) כך ש

$$\alpha_S = \underset{\alpha \in \mathbb{R}^m}{argmin} \left(f(G\alpha) + \lambda \alpha^\top G \alpha \right)$$

כאשר המטריצה G (מטריצת גראהם) מוגדרת כך: $\mathrm{G}_{i,j} = \langle \psi(\mathbf{x}_i), \psi(\mathbf{x}_j) \rangle$.

משפט $Kernel\ Trick$: מתקיים כי

$$w_S = \sum_{i=1}^{m} (\alpha_S)_i \, \psi(x_i)$$

במילים אחרות - אין טעם להסתכל על כל R^k ועל ההעתקה ψ, אלא הפתרון נמצא בת"מ ב R^m. מה שנשאר זה לחשב את המטריצה G בצורה יעילה.

- **הגדרה - פונקציית קרנל:** פונקציה שמקבלת שני משתנים $K(\cdot, \cdot)$ שמושרית ע"י העתקה $\psi : \mathbb{R}^d \to \mathbb{R}^k$ כך

$$K(x, x') = \langle \psi(x), \psi(x') \rangle$$

למעשה פונקציית הקרנל מחזירה את הזווית בין שתי דגימות.

- **איך נעשה** $predict$ **עם הבעיה החדשה:** לאחר שמצאנו את α נוכל לעשות פרדיקציה באופן הבא, עבור נקודה x חדשה:

$$\langle w, \psi(\mathbf{x}) \rangle = \sum_{i=1}^{m} (\alpha_S)_i \, , K(x_i, x)$$

עבור רגרסיה לינארית זה החיזוי, ועבור קלסיפיקציה נקח את הסימן.

11.2.1 קרנלים מפורסמים:

- **מה אנו מחפשים:** אנו רוצים דרך מהירה יותר לחשב את המכפלה הפנימית $\langle \psi(\mathbf{x}), \psi(\mathbf{x}') \rangle$ כדי שנוכל ליצור את המטריצה G, ולחשב את פונקציית ה $kernel$ ולעשות פרדיקציה ביעילות. נביא כמה קרנלים שונים ואת הדרך לחשב עליהם ביעילות את המ"פ.

- **קרנל פולינומי -** $Plinomial\ kernel$: עבור פונקציה ψ שמוגדרת באופן הבא:

$$\psi(x)_a = \prod_{i=1}^{d} x_i^{ai}$$

ניתן לכתוב את המ"פ בצורה יעילה יותר כך

$$K(x, x') = \langle \psi(\mathbf{x}), \psi(\mathbf{x}') \rangle = (1 + \langle x, x' \rangle)^n$$

כך ניתן לפתור מבלי לדעת מהי הפונקציה ψ, בנוסף נוכל לפתור יותר מהר כי מ"פ של x, x' היא ב R^d, ולכן הסיבוכיות יורדת מ $O(d^n)$ (חישוב של מ"פ $(\psi(x)$), ל $O(d)$.

כעת, במקום ליצור מטריצת וונדרמונד מגודל $(d^m) \times m$, נפתור עם $kernel$ כך:

$$\min_{\alpha\in\mathbb{R}^m} \|G\alpha - y\|^2 + \lambda\alpha^\top G\alpha$$

- **הגדרה ־ מרחב הילברט:** מרחב ווקטורי אינסופי (R^n), כלומר ־ יש לו קבוצה בת"ל מכל גודל שנבחר.
- **קרנל גאוסיאני ־** $Gaussian\ Kernel\ (RBF)$ **עבור חד מימד:** נתחיל ממימד R ונעלה למימד R^∞ (הפונקציה ψ לוקחת אותנו מ R למרחב הילברט). נמפה לסדרה הבאה:

$$\psi(x) = \left(1, e^{-\frac{x^2}{2}}x, \frac{1}{\sqrt{2}}e^{-\frac{x^2}{2}}x^2, \ldots\right)$$

ובאופן כללי:

$$\psi(x)_n = \frac{1}{\sqrt{n!}}e^{-\frac{x^2}{2}}x^n$$

המ"פ שווה ל:

$$\langle\psi\left(\mathbf{x}\right), \psi\left(\mathbf{x}'\right)\rangle = e^{-\frac{\left(x-x'\right)^2}{2}}$$

באופן זה לא צריך לחשב מ"פ ב R^∞אלא נעבוד עם סקלרים והסיבוכיות היא $O(1)$.

- **קרנל גאוסיאני ־** $Gaussian\ Kernel\ (RBF)$ **עבור** d **מימדים:** נתחיל מ R^d ונעלה למימד R^∞ (הפונקציה ψ לוקחת אותנו מ R למרחב הילברט). נגדיר את הפונקציה ψ בכניסה ה a להיות:

$$\psi(x)_{\mathrm{a}} = \frac{1}{\sqrt{n!}}e^{-\frac{\|x\|^2}{2}}\prod_{i=1}^d x_i^{a_i}$$

והקרנל יהיה:

$$K\left(\mathbf{x}, \mathbf{x}'\right) = \langle\psi(\mathbf{x}), \psi\left(\mathbf{x}'\right)\rangle = e^{-\frac{\||x-\mathbf{x}'\|^2}{2}}$$

- **קרנלים נוסף:** הפעם כלל לא נציג את המ"פ אלא רק את פונקציית הקרנל:

$$1 : K\left(\mathbf{x}, \mathbf{x}'\right) = \left\|\mathbf{x} - \mathbf{x}'\right\|^2 \log\left(\left\|\mathbf{x} - \mathbf{x}'\right\|\right)$$
$$2 : K\left(\mathbf{x}, \mathbf{x}'\right) = \tanh\left(a\left\langle\mathbf{x}, \mathbf{x}'\right\rangle + b\right),\ \text{where } a, b \in \mathbb{R}$$

- **למה ־ התנאי של מרסר:** תהי פונקציית קרנל $K\left(\mathbf{x}, \mathbf{x}'\right)$ כך שהמטריצה G שמתקבלת היא PSD, אמ"מ היא מקיימת מ"פ במרחב הילברט (לכן אין צורך להציג את ψ ומ"פ).

12 שבוע 12:

12.1 תרגול 12 - $Kernels$ (29.5):

- ראינו ראינו.

12.2 הרצאה 12 - פרקטיקה:

12.2.1 ניתוח מקדים של הדאטה:

- **ניתוח הדאטה:** נרצה להבין מה הפיצ'ר מייצג, האם הוא קטגורי, רציף, בינארי. כל סוג של משתנה נטפל ונתייחס בדרך אחרת.
  הבנה של הפיצ'רים תעזור לנו לסנן מידע לא נכון שהופיע בטעות בדאטה.

- **הטלת ספקות בתיוגים:** כשנקבל מידע מתוייג לא נקח את התיוגים כנכונים באופן אבסולוטי, אלא נסתכל על דוגמאות וננסה להבין אם התיוגים אכן נכונים ביחס לדוגמאות. נוכל לתת לכמה אנשים לתייג את הדאטה, ולקחת את הדגימות שהרוב הסכימו עליהן.
  אם הלייבלים יהיו עם רעש כל אלגוריתם ייכשל, לכן נוודא תחילה תקינות של הדוגמאות.

- **מדד** $Cohens\ Kappa$: מדד שבא לבדוק אחוזי הסכמה בין מתייגים

$$\kappa = \frac{p_o - p_e}{1 - p_e}$$

  כאשר p_o מייצג את אחוז ההסכמה הצפוי. p_e מייצג תיוג רנדומלי.
  כלל ש k גבוה יותר, אנו יותר נסמוך על התיוג.

- **הטיה של לייבלים:** לעיתים מכיוון שאנשים מתייגים את הדאטה תהיה הטיה בתיוגים. בנוסף נצטרך לשים לב שהמודל לא יאומן על דוגמאות מסוג מסויים בלבד (אימון מודל עם תמונות של נשים במטבח, ואז בהינתן תמונה חדשה של גבר מבשל היא תתוייג כאישה).

- **חלוקת הדאטה:** כשנקבל דאטה חדש נצטרך להפריד אותו לסט אימון וטסט סט. נצטרך לדאוג שההתפלגות של חלוקת הדאטה תהיה דומה להתפלות המקורית של הדאטה.

12.2.2 פיתוח מודל:

- **שלבים בפיתוח מודל:**

$Preprocessing$ **- עיבוד מקדים של הדאטה:** נבדוק האם יש ערכים חסרים בפיצ'ר מסויים, ניצור פיצ'רים חדשים, ננקה את הדאטה. נבצע את השלב זה **רק** על סט האימון.

EDA: נעבור על הדאטה ונבדוק איזה פיצ'רים יהיו רלוונטים לתיוג הבעיה.

$Baseline$: נרצה לבדוק מהו המודל הבסיסי ביותר לפתירת הבעיה, כך נוכל להשוות למודל הסופי שלנו ולמדוד ביצועים.

בחירת מודל: המטרה היא לשפר את ה $baseline$ שלנו. נוכל לעשות זאת בעזרת $bagging, boosting, regularization, kernel$ ועוד.

- $imputation$**:** השלמת מידע כשיש לנו חוסר בפיצ'ר. ניתן להזין את החציון או הממוצע במקום הנקודה החסרה וכך לא לאבד את הדגימה.

13 שבוע 13:

13.1 תרגול 13 - $fiture\ selection$ וגרדיאנטים (7.6):

היוריסטיקות למציאת קבוצת הפיצ'רים הטובה ביותר.

13.1.1 האלגוריתם $Forward\ Stepwise\ Selection(FS)$:

- **האלגוריתם** $Forward\ Stepwise\ Selection(FS)$**:** אלגוריתם חמדן לפתרון הבעיה.

Algorithm 1 Forward Stepwise Selection

```
1: procedure FS-SELECTION(d)
2:     Denote M_0 the null model
3:     for k = 0,...,d−1 do
4:         Consider all models with the predictors as in M_k and one additional predictor of the
           remaining d−k predictors.
5:         Choose the best among these d−k models, and call it M_{k+1}
6:     end for
7:     Select a single best model from among M_0,...,M_d.
8: end procedure
```

- **כיצד נבחר את הפיצ'ר הטוב ביותר:** בשורה 5 באלגוריתם אנו צריכים לבחור פיצ'ר, נבחר כך:
$$R^2 := 1 - \frac{\text{Unexplained Variation}}{\text{Total Variation}} = 1 - \frac{\text{SSE}}{\text{SST}}$$
למעשה שיטה זו באה לאמוד את אחוז ה var של הדאטה אותו אנו יכולים להסביר.
ככל ש R^2קרוב ל 1 המודל טוב יותר.

- SSE**:** מוגדר להיות - $||y-\hat{y}||_2^2$. (כמו RSS)
SST**:** מוגדר להיות - $||y-\bar{y}\cdot 1||_2^2$, והוא גורם הנרמול. (המכפלה היא בווקטור של 1 -ים, ו $\bar{y}$ הוא הממוצע של y).
הערה: מתקיים $SSE \le SST$.

- **כיצד נבחר את המודל הטוב ביותר:** בשורה 7 אנו צריכים לבחור את המודל הטוב ביותר, נעשה זאת כך:
$$\text{Adjusted } R^2 := 1 - \frac{\frac{\text{RSS}}{m-d-1}}{\frac{m\cdot Var(y)}{m-1}} = 1 - \frac{m-1}{m-d-1}\left(1-R^2\right)$$
כך אנו מתחשבים גם בגודל המודל שנבחר.
הערה: מקסימום של $\text{Adjusted } R^2$ שקול למינימום על $\frac{RSS}{m-d-1}$.

- BIC**- קריטריון נוסף לבחירת המודל הטוב ביותר:**
$$\text{BIC} := \frac{\text{RSS} + \log(m)d\hat{\sigma}^2}{m\hat{\sigma}^2}$$

13.1.2 קמירות וגרדיאנטים:

- **הגדרה - פונקציה קמורה:** פונקציה $f : C \Rightarrow R$ תיקרא קמורה אם היא מקיימת את התכונות הבאות -
1: התחום שלה C קמור .
2:

$$\forall \mathbf{u}, \mathbf{v} \in C \text{ and every } \alpha \in [0,1] \text{ then } f(\alpha \mathbf{v} + (1-\alpha)\mathbf{u}) \leq \alpha f(\mathbf{v}) + (1-\alpha) f(\mathbf{u})$$

אינטואיציה: ניתן לחבר קו בין כל זוג נקודות של הפונקציה , כך שהקו נמצא **מעל** גרף הפונקציה.

- **תזכורת: גרדיאנט** - ווקטור הנגזרות החלקיות. **ההסיאן** - מטריצת הנגזרות השניות. (לא תמיד מתקיימים, אלא אם הפונקציה דפרינציאבילית).

- **הגדרה - פונקציה דפרנציאבילית:** פונקציה שקיים לה גרדיאנט בכל נקודה

- **משפט - קמירות מסדר ראשון (גרדיאנט):** פונקציה דיפרנציאבילית $f : R^n \Rightarrow R$ קמורה אמ"מ לכל $x, y \in R^n$ מתקיים

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^\top (\mathbf{y} - \mathbf{x})$$

אינטואיציה: אם המשיק לנקודה נמצא מתחת לגרף הפונקציה.

- **משפט - קמירות מסדר שני (הסיאן):** פונקציה דיפרנציאבילית פעמיים $f : R^n \Rightarrow R$ קמורה אמ"מ לכל $x \in R^n$ מתקיים כי מטריצת ההסיאן היא PSD: $\nabla^2 f(\mathbf{x}) \succeq 0$.

- **מטריצה היא PSD:** אם אחד מהתנאים הבאים מתקיים
1: אם ע"ע ≤ 0. **2:** לכל x - $x^T A x \geq 0$. **3:** $det(X) \geq 0$.

- **הערה:** הרכבה של פונקציות קמורות אינו בהכרח קמור. אך אם הפונקציה המורכבת מונוטונית עולה (נגזרת חיובית) - ההרכבה קמורה.

- **הגדרה - תת גרדיאנט:** נשתמש בו כשיש לנו פונקציה קמורה שאינה דיפרנציאבילית.
תהי פונקציה $f : R^d \Rightarrow R$, נאמר כי ווקטור $v \in R^d$ הוא תת גרדיאנט של f בנקודה x שייך לדומיין של f. אם לכל u בדומיין מתקיים:

$$f(\mathbf{u}) \geq f(\mathbf{x}) + \langle \mathbf{v}, \mathbf{u} - \mathbf{x} \rangle$$

אינטואיציה: הפונקציה אינה גזירה ב x, ולכן יש כמה משיקים לגרף הפונקציה בנקודה x, אך אם כל המשיקים נמצאים מתחת לגרף הפונקציה אזי היא קמורה על אף שאינה גזירה בכל התחום.

- **הגדרה - תת דיפרנציאל:** הסט של כל תתי הגרדיאנט של f בנקודה x נקראים תת דיפרנציאל של $f(x)$, ומסומנים ב $\partial f(\mathbf{x})$.
גודלה הוא : אם אין נגזרת - קבוצה ריקה. שווה ל 1 - אם היא דיפרנציאבילית. אם היא אינה דיפרנציאבילית - א.

- **טענה:** תת הדיפרנציאל היא קבוצה קמורה וסגורה.

- **למה:** פונקציה היא קמורה אמ"מ לכל נקודה תת הגרדיאנט אינה קבוצה ריקה.

13.2 הרצאה 13 - קמירות (8.6):

- **בעיות קמירות:** המודלים שלמדנו במהלך הקורס למעשה חיפשו פתרון לבעיות אופטימיזציה קמורות. נרצה להבין את התיאוריה מאחורי הלומדים.

- **הגדרה - קבוצה קמורה:** באופן אינטואיטיבי - קבוצה קמורה היא סט של נקודות, כך שכל שתי נקודות שנבחר, הקו שעובר בניהן נמצא בקבוצה.

  פורמלית: קבוצה $C \subseteq R^d$ קמורה אם לכל v, u ולכל $\alpha \in [0,1]$ מתקיים כי $\alpha v + (1-\alpha)u \in C$.

- **משפט - על מישור מפריד:** אם שתי קבוצות C, D לא ריקות, זרות וקמורות ב R^d קיים על מישור שמפריד ביניהם. (מישור כך שD מצידו האחד ו C מתידו האחר).

- **הגדרה - פונקציות קמורות:** פונקציה $f : C \Rightarrow R$ תיקרא קמורה אם לכל שתי נקודות על גרף הפונקציה, הקו המחבר בניהן יהיה מעל גרף הפונקציה.

  פורמלית: אם לכל v, u ולכל $\alpha \in [0,1]$ מתקיים כי :

$$f\left(\alpha v + (1-\alpha)u\right) \leq \alpha f\left(v\right) + (1-\alpha)f(u)$$

- **הגדרה - פונקציה קעורה:** פונקציה f תיקרא קעורה, אם $-f$ היא פונקציה קמורה.

- **הערה:** תחום של פונקציה קמורה חייב להיות קבוצה קמורה.

  טענה: אם פונקציה קמורה, אזי הקבוצה שנמצאת מעל גרף הפונקציה היא קבוצה קמורה.

- **תכונות של פונקציות קמורות:**

  1: סכום של פונקציות קמורות שמוכפלות ב $a_i > 0$ היא פונקציה קמורה.

  2: $max_i f_i$ עבור i פונקציות קמורות היא קמורה.

  3: מנימיזציה חלקית של פונקציה (רק לפי חלק מהמשתנים) היא קמורה.

  4: אם $f(x)$ קמורה, אזי גם עם העתרה אפינית $f(Ax+b)$ היא קמורה.

  5: הרכבה של פונקציות קמורות כך שהמורכבת מונוטונית עולה - היא פונקציה קמורה.

- **תכונות של פונקציות קמורות:**

  1 מינימום לוקאלי הוא גלובאלי: אם f קמורה ומצאנו מינימום לוקאלי, אזי הוא המינימום הגלובלי.

  2 קירוב מסדר ראשון: קירוב מסדר ראשון (לינארי - משיק לנקודה) של פונקציה קמורה בנקודה x, נמצא מתחת לגרף הפונקציה עבור כל נקודה.

13.2.1 אופטימיזציה קמורה:

- **הגדרה - בעיית אופטימיזציה:** בעיה מהצורה הבאה היא בעיית אופטימיזציה.

$$minimize_{x \in D} \quad f(x),\ subject\ to f_i(x) \leq b_i, \quad i = 1 \ldots n$$

- **בעיית אופטימיזציה קמורה:** אם הפונקציות f_i הן קמורות אזי הבעיה היא קמורה.
- **בעיית תכנות לינארי:** אם f_i לינארית, הבעיה היא בעיית תכנות לינארי.
- **בעיית תכנות ריבועי:** אם f תבנית ריבועית, והאילוצים f_i לינארים אבעיה היא בעיית תכנות ריבועי.
- תחום הבעיה: נגדיר את הדומיין של הפונקציה f להיות החיתוך של כל האילוצים

$$D = dom(f) \cap \left(\bigcap_{i=1}^{n} dom\left(f_i\right) \right)$$

- **נקודה פיזבילית:** הנקודה שמקיימת את כל האילוצים נקראת נקודה פיזבילית.
- **נקודה אופטימלית\ מינימייזר:** אם הנקודה פיזבילית וגם ממנממת את האובייקט.
- **טענה:** קבוצת הפתרונות לבעיית אופטימיזציה קמורה היא קבוצה קמורה.
- **אלגוריתמים לבעיות אופטימיזציה קמורה:** נרצה $solver$ ספציפי לבעיה ספציפית שתפתור אותן ביעילות, לכן למרות שיש לנו אלגוריתמים כללים נרצה אלגוריתמים ספציפיים. הם מחולקים לשניים -
שיטות מסדר ראשון יש להן גישה רק לגרדיאנט או תת הגרדיאנט של הפונקציה - $gradient\ descent$.
שיטות מסדר שני יש גישה להסיאן - קירוב מסדר שני של הפונקציה.
- **הגדרה - בעיית למידה קמורה:** בעיית למידה (H, Z, l) כאשר H מחלקת היפותזות, $Z = X \times Y$ ו l פונקציית לוס. נקראת בעיית למידה קמורה, אם מחלקת ההיפותזות היא קבוצה קמורה ופונקציית הלוס קמורה.
- **אבחנה:** בבעיית למידה קמורה אם אנו רוצים להשתמש ב ERM - $\min_{w\in\mathcal{H}} \frac{1}{m}\sum_{i=1}^{m} \ell\left(w, z_j\right)$ היא בעיית אופטימיזציה קמורה.
- **טענה:** לא כל בעיות הלמידה הקמורות הן למידות PAC.
- **הקשר בין למידות PAC לבעיות אופטימיזציה קמורות:** אם נוסיף את התנאים הבאים, נקבל קשר ביניהן.
1 - מחלקת ההיפותזות היא קבוצה חסומה (התנאי לא מתקיים בלינאריות).
2 - פונקצייה ההפסד היא הפונקציה הבאה (ליפשיץ): עבור סקלר ρ ועבור ווקטורים w_1, w_2 מתקיים

$$\left|f\left(w_1\right) - f\left(w_2\right)\right| \le \rho \left\|w_1 - w_2\right\|$$

- **הגדרה - בעיית למידה ליפשיץ קמורה:** בעיית למידה (H, Z, l) נקראת בעיית למידה ליפשיץ קמורה אם:
1 - אם H קמורה. **2** - הפונקציה l היא ליפשיץ לכל Z.
- **משפט:** בעיית למידה שמקיימת את התנאים הבאים - היא קמורה, H חסומה, ו l פונקציית ליפשיץ. אזי היא למידה PAC.

13.2.2 $Gradient\ Descent(GD)$:

- **הרעיון:** נרצה למצוא מינימום של פונקציה בעזרת הגרדיאנט (נגזרת של פונקציה בכמה משתנים). כך שאם נתקדם בכיוון הירידה (מינוס הגרדיאנט) בצעדים בגודל מתאים, מובטח לנו שנמצא את המינימום הגלובלי.

- **למה זה חשוב:**

  1: אם אנו לא יכולים להשתמש ב $loss_0$ ונרצה להחליף פונקציית הלוס.

  2: אם יש לנו כמות ענקית של דאטה שאי אפר לאחסן אותו על מחשב יחיד.

  3: אם הדאטה שלנו זורם באונליין והמערכת צריכה ללמוד להשתפר מדאטה חדש שנכנס.

  4: למידה עמוקה מבוססת על $Gradient\ Descent$.

- **תכונות של הגרדיאנט:**

  1: אם f דפרינציאבילית - הגרדיאנט יצביע על העלייה התלולה ביותר, ומינוס הגרדיאנט יצביע לכיוון הירידה התלולה ביותר.

  2: עבור $x \in \mathbb{R}^d$ נגדיר את ה $level\ set$ להיות:

$$L_x(f) = \{x' \mid f(x) = f(x')\}$$

  קבוצת כל x' שבהם f שווה לנקודת הבוחן שהגדרנו.

  3: אם f דפרינציאבילית - ו w משיק ל $level\ set$ אז $\langle \nabla f(x), w \rangle = 0$ כלומר הוא מאונך לגרדיאנט.

- **אפיון מסדר ראשון לאופטימליות:** לבעית אופטימיזציה קמורה, נקודה $x \in C$ תיקרא אופטימלית אמ"מ הגרדיאנט מאונך ל $x - y$.

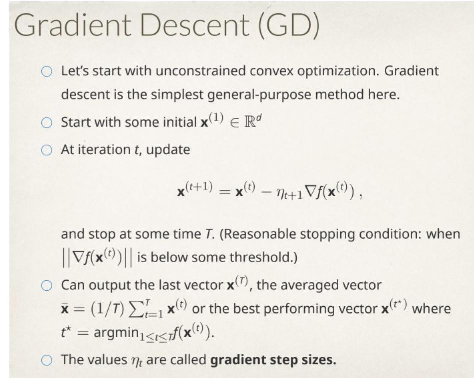
$$\langle \nabla f(x), y - x \rangle \geq 0 \quad \forall y \in C$$

  והנקודה x הזו היא הפתרון.

- **דוגמה - פתרון לבעיית תכנן ריבועי ללא אילוצים:** התעסקנו בבעיה כזו כשרצינו לפתור את רגרסיה לינארית, נרצה $Gradient\ Descent$ אם יהיה לנו דאטה גדול שאי אפשר להפעיל עליו svd.

האלגוריתם GD:

- **האלגוריתם GD:** אנו מחפשים את הצעד האופטימלי שנקח בכיוון שאליו אנו צריכים להתקדם. מצד אחד אנו לא רוצים לקחת צעדים קטנים ולהתקדם לאט, מצד שני אנו לא רוצים לקחת צעדים גדולים ולעקוף את הנקודה.
  נתחיל מבעיה ללא אילוצים ונוסיף אותם אח"כ.



אנו צריכים לקבוע: מהו T. מהו ערך ההחזרה (בפסואודו קוד יש שלש אופציות). מהו גודל צעד הגרדיאנט - η_t.

הערה: בהרצת האלגוריתם נצטרך למצוא את הגרדיאנט ולקודד אותו, כדי שיהיה לנו זמין בכל נקודה.

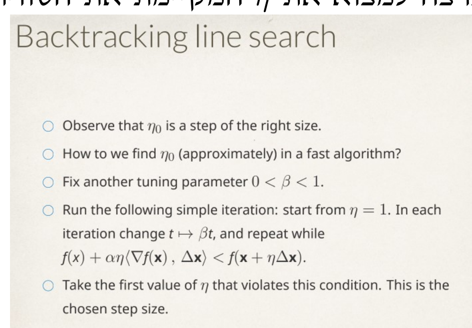
- **מהו x שאיתו אנו מתחילים:** מכיוון שאנו לא מניחים קיום של נגזרת שניה, נעבוד עם נגזרת ראשונה + הערכה (קירוב ריבועי נאיבי), נמצא את המינימום של הפונקציה הבאה, וזה יהיה $x^{(1)}$:

$$f(\mathbf{y}) \approx f(\mathbf{x}) + \langle \mathbf{y}-\mathbf{x}, \nabla f(\mathbf{x})\rangle + \frac{1}{2\eta}\|\mathbf{y}-\mathbf{x}\|^2$$

- **איך נבחר את גודל צעד הגרדיאנט - η_t:** נשתש בשיטה $Backtracking\ line\ search$ - בכל צעד נגדיר את גודל הצעד מחדש. נגדיר את כיוון ההתקדמות להיות $\Delta x = -\nabla f(x)$, ובכל שלב נזוז מ x ל $\mathrm{x}+\eta\Delta\mathrm{x}$.
אנו יודעים כי עבור $\eta_0 > 0$ כלשהי (שמסמנת את גודל הצעד), ועבור $0 < \alpha < 1$ מתקיים השוויון:

$$f(x) + \alpha\eta_0\langle\nabla f(\mathbf{x}), \Delta\mathbf{x}\rangle = f\left(\mathbf{x}+\eta_0\Delta\mathbf{x}\right)$$

נרצה למצוא את η המקיימת את השוויון. נעשה זאת כך:



- **פתרון לבעיה עם אילוצים - $Projected\ GD$:** למעשה אנו נעזה צעד בכיוון הגרדיאנט ונטיל לתוך הקבוצה הפיזיבילית.
נניח כי יש לנו בעיית אופטימיזציה קמורה מהצורה הבאה:

$$\min_{w} f(w) \text{ subject to } w \in C$$

נגדיר $projection\ operatior$ **על** C **כך:**

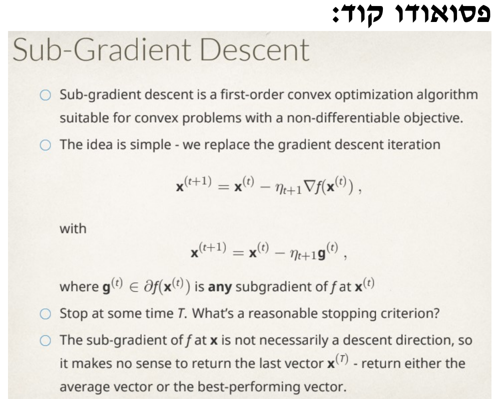
$$P_C(x) = argmax_{w\in c}\|x-w\|_2$$

הערה: הבעיה מוגדרת היטב מכיוון ש C (הקבוצה הפיזבילית) קבוצה קמורה.
כלומר: בכל צעד נצטרך למצוא את P_C שיטיל לנו את הצעד על הקבוצה.
כעת בכל איטרציה נבצע את הפעולה הבאה:

$$x^{(t+1)} = P_C\left(x^{(t)} - \eta_{t+1}\nabla f\left(x^{(t)}\right)\right)$$

13.2.3 $Sub - Gradient\ Descent(GD)$:

- **הרעיון:** נרצה שכל מה שעובד לנו עבור פונקציות דיפרנציאבליות יעבוד לנו גם עבור פונקציות קמורות שאינן דיפרנציאבליות.
- $Sub - Gradient\ Descent(GD)$: הוכחנו שאם הבעיה קמורה אזי יש לפחות תת גרדיאנט אחד, נבחר אחד מהם ונתקדם בכיוון שלו. כנראה שלא נתקרב הכי מהר, אך נתכנס למינימום.
- $Sub - Gradient\ optimality\ condition$: עבור פונקציה קמורה, $f(x^*)$ הוא מינימום אמ"מ 0 הוא תת גרדיאנט של $f(x^*)$.
- **פסואודו קוד:**



- **כיצד נבחר את הצעד:** נבחר את הצעד עבורו הטורים הבאים מקיימים -

$$\sum_{t=1}^{\infty}\eta_t^2 < \infty \text{ but } \sum_{t=1}^{\infty}\eta_t = \infty$$

14 שבוע 14 - $Gradient\ Descent(GD)$ (12.6):

14.1 תרגול 14:

- **פסואודו קוד לאלגוריתם** GD: האלגוריתם מקבל פונקציה, וסט של $\{\eta_t\}$, אנו קובעים את T.
הערה: ערך ההחזרה יכול להשתנות כפי ראינו בהרצאה 13.

```
Algorithm 1 Gradient Descent
procedure GRADIENT DESCENT(f, {η_t})
    Initialize x^(1) ∈ ℝ^d
    for t = 1,...,T do
        x^(t+1) ← x^(t) − η_{t+1} ∇f(x^(t))
    end for
    return x̄ = (1/T) Σ x^(t)
end procedure
```

- **האלגוריתם** GD **עם אלגוריתם** $Line\ Search$ **לבחירת הצעד -** η**:** האלגוריתם מקבל פרמטרים - α, β.

```
Algorithm 2 Gradient Descent With Backtracking Line Search
procedure LINE-SEARCH(x)
    Initialize η ← 1
    while f(x) + αη⟨∇f(x), Δx⟩ < f(x + ηΔx) do
        η ← βη
    end while
    return η
end procedure

procedure GRADIENT DESCENT(f)
    Initialize x^(1) ∈ ℝ^d
    for t = 1,...,T do
        η ← LINE-SEARCH(x^(t))
        x^(t+1) ← x^(t) − η∇f(x^(t))
    end for
    return x̄ = (1/T) Σ x^(t)
end procedure
```

- **האלגוריתם עבור** $Sub\ Gradient$**:**

```
Algorithm 3 Sub-Gradient Descent
procedure SUB-GRADIENT DESCENT(f)
    Initialize x^(1) ∈ ℝ^d
    for t = 1,...,T do
        η ← LINE-SEARCH(x^(t))
        Pick v^(t) ∈ ∂f(x^(t))
        x^(t+1) ← x^(t) − η_{t+1} v^(t)
    end for
    return x̄ = (1/T) Σ x^(t)
end procedure
```

14.2 הרצאה 14 - למידה עמוקה:

14.2.1 $Stochastic\ gradient (SGD)$:

- **הרעיון:** בניגוד ל GD, ניקח צעדים אקראיים, ונתבסס על כך כי בתוחלת אנו נתכנס לכיוון הנכון - המינימום. במקום לבחור את כל הסט, נבחר רק נקודה בודדת, או $btach$ - קבוצת נקודות ונעבוד עליהן.

- **הגדרת האיטרציות:** נגדיר את האיטרציות באופן הבא:

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \eta_{t+1}\mathbf{g}^{(t)}$$

- SGD עבור בעיות קמורות: עבור בעיה קמורה עם מחלקת היפותזות $H \subset R^d$ ופונקציית לוס $l(W, (x, y))$ וסט S, נגדיר $z = (x, y)$. תהי $L_s(W)$ שגיאת ההכללה של ההיפותזה W. נרצה למצוא את המינימום הבא: $\min_{\mathbf{w}\in\mathcal{H}} L_s(\mathbf{w})$ (למעשה ניתן להחליף את L_S בכל פונקציה).
יהיו $z_1...z_r$ מתפלגים אחיד וב"ת על S אזי:

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta_{t+1}\mathbf{g}^{(t)} \quad \text{where } \forall t.\mathbf{g}^{(t)} \in \partial\ell\left(\mathbf{w}^{(t)}, z_t\right)$$

וזאת האיטרציה שממנממת את $W \Rightarrow L_s(W)$.

- **ההבדלים בין GD ל SGD:** ב GD אנו מסתכלים על כל הסט ומשווים את הגרדיאנט $\nabla L_s\left(\mathbf{w}^{(t)}\right)$ בנקודה $\left(\mathbf{w}^{(t)}\right)$. לעומת זאת ב SGD - בכל איטרציה נבחר i רנדומלי ונשתמש ב $\nabla \ell\left(\mathbf{w}^{(t)},(\mathbf{x}_i, y_i)\right)$ כדי להעריך את $\nabla L_s\left(\mathbf{w}^{(t)}\right)$, וזהו אומד בלתי מוטה.

- **הורדת השונות בעזרת ממוצע:** נוכל לקחת $mini-batch$ - $B\subset\{1...m\}$ונשתמש ב

$$\frac{1}{|B|}\sum_{i\in B}\nabla \ell\left(\mathbf{w}^{(t)},(\mathbf{x}_i, y_i)\right)$$

כהערכה ל $\nabla L_s\left(\mathbf{w}^{(t)}\right)$.

- **כלל המעגל:** במקום לבחור נקודות שונות באופן רנדומלי עם חזרות, נוכל לקחת בכל פעם קבוצה בגודל B ולהוציא אותה מהמשחק, ולאחר מכן כשנסיים עם כל הדאטה נבחר שוב.

- **הגדרה - $epoch$:** מספר הצעדים שנדרשים כדי לכסות את כל הדאטה כאשר אנו עובדים בקבוצות בגודל B, כלומר - $\frac{m}{B}$.

14.2.2 רשתות נוירונים:

- **הרעיון:** הרכבת פונקציות, נניח כי הרצנו מודל של רגרסיה לוגיסטית וקיבלנו ווקטור משקולות w_1^1 וכעת אנו יכולים לעשות פרדיקציה. נעשה את אותו הדבר על k מודלים שונים ונקבל k ווקטורי w שונים. למעשה יש לנו פונקציה $f: R^d \Rightarrow R^k$. נגדיר את k וקטורי המשקולות שלנו במטריצה - W, ולהכניס אותה לפונקצייה האקטיביציה עם X. נוכל להמשיך ולעשות זאת עם מספר שכבות, כך שהשכבה האחרונה תהיה שכבת הפלט.

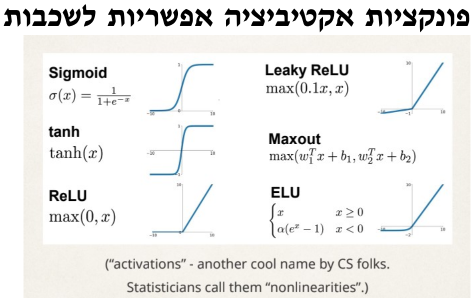
- **הגדרה - נוירון:** מוגדר ע"י משקולות - w, ופונקציית אקטיביציה ϕ - פונקציה לא לינארית (כשי שנוכל לחזות מחלקות מרכבות). נעשה מ"פ $\langle x, w\rangle$ ועליה נפעיל את ϕ.

- **הגדרה - מחלקת ההיפותזות:** מחלקת ההיפותזות של $feedforward\ neural\ nerwork$ עם שכבה מוסתרת אחת (למעט שכבת הקלט ושכבת הפלט) ופונקציית אקטיביציה σ, עבור קלסיפיקציה בינארית או רגרסיה היא:

$$\mathcal{H}=\left\{\phi\left(\left\langle\sigma\left(W_1^{\top}\mathbf{x}\right), w_2\right\rangle\right)\right\}$$

כאשר $\sigma: R \Rightarrow R$ פונקציית אקטיבציה, וגם ϕ פונקציית אקטיביציה.
כשנרצה קלסיפיקציה נבחר פונקציה שממפה לנו את R לקטע $[0,1]$, לבעיות רגרסיה נקח את הפלט.

- **פונקציות אקטיביציה אפשריות לשכבות המוסתרות:**



("activations" - another cool name by CS folks.
Statisticians call them "nonlinearities".)

- $multiple\ regrassion$: רגרסיה שבה נרצה למדל כמה מספרים רציפים ולא רק אחד (לדוגמה גובה ומשקל).

- **מידול של רגרסיה עם רשתות:** עבור קלט $o_2 = \sigma(W_1^T x)$ נחזיר את $\langle o_2, w_2 \rangle$.

- **מידול של קלסיפיקציה עם רשתות:** עבור קלסיפיקציה בינארית ועבור קלט $o_2 = \sigma(W_1^T x)$ נחזיר את

$$logit_{w_2}(o_2) = \frac{e^{\langle o_2, w_2 \rangle}}{1 + e^{\langle o_2, w_2 \rangle}}$$

- **הגדרה - קלסיפיקציה עם מולטי קלאס:** אנו כעת בהתפלגות מולטינום, והמודל מחזיר לנו את ההסתברות להיות שייך לכל מודל.

- **מידול של קלסיפיקציה מולטי קלאס עם רשתות:** עבור z_jהמכפלה הפנימית המיוצגת ע"י הפלט של הנוירון ה j, נגדיר $softmax$:

$$\phi : (z_1, \ldots, z_k) \mapsto \left(\frac{e^{z_1}}{\sum_\alpha e^{z_\alpha}}, \ldots \frac{e^{z_k}}{\sum_\alpha e^{z_\alpha}} \right)$$

כאשר $\phi : R^k \Rightarrow R^k$.

- **הגדרה -** $feedforward\ neural\ nerwork$ **עם** d **כניסות ו** t **שכבות:** עם פונקציית אקטיבציה σ, הרשת מוגדרת ע"י גרף מכוון ציקלי ופונקציית משקל לכל קשת (w) בגרף, כך שהאיבר ההתחלתי הוא $o_0 = x$, את השכבות הבאות נגדיר באופן רקורסיבי:

$$\mathbf{o}_t = \sigma\left(W_{t-1}^\top \cdot \mathbf{o}_{t-1}\right) \quad 1 \leq t \leq T.$$

- **דיאגרמת בלוקים:** נוכל לבנות את הרשת מבלוקים של רשתות עם מלא שכבות נסתרות, כך שכל בלוק מוזן מבלוק אחר גם לא באופן רציף.

- **גילוי מרעיש:** אם אנו משתמשים ב GD כדי למקסם את ה $likelihood$ של רשת נוירונים, נוכל לבנות אלגוריתם לומד מצויין על אף שהבעיה לא קמורה כלל. למעשה אלו נקודות מינימום מקומיות ולא גלובליות, אך מאל ווקטורי משקולות מקומיים יובילו להכללה טובה. בנוסף ה var יורד.

- **עיקרון הלמידה של רשת נוירונים:** המודל מורכב ממבנה אוקלידי של ווקטורי משקולות, ואנו נלמד לפי $max\ likelihood$. למעשה אנו רוצים לגזור את הפונקציה $loss$ שחזרה לנו מהשכבה האחרונה לפי משקל w מסויים, אך אנו צריכים להסתכל על כל הנוירונים שהמשקולת הזאת השפיעה עליהם (יכול להיות מספר עצום).
לכן כדי למקסם את ה $likelihood$ **בצורה יעילה יותר: 1** - נשתמש ב SGD (נסתכל כל פעם על תת קבוצה). **2** - כדי לחשב את הגרדיאנט בנקודה מסויימת (בעיה קשה כי יש מלא שכבות) נשתמש באלגוריתם נומרי $backprop$.

15 שבוע 15:

15.1 תרגול 15 - האלגוריתם $backprop$:

- **האלגוריתם** $backprop$**:** מכיוון שיש לנו מלא משקולות ונויירונים נרצה צורה יעילה יותר לגזירת המשתנים.
למעשה יש לנו מלא פונקציות שמורכבות אחת על השניה, לכן נרצה להשתמש ביעקוביאנים וכלל השרשרת.

- **הרעיון:** אנו לוקחים חישוב ארוך, מפצלים אותו לכמה חישובים פשוטים כך שכל חישוב מכיל חלקים מהחישוב הקודם, ולכן אין צורך לחשב נגזרת של כל איבר כמה פעמים.
למעשה בכל צעד קדימה בשכבה אנו נשמור את המשתנים שנצטרך בגזירה, כך לא נסתבך עם גזירות מורכבות.
בנוסף תחיל לגזור מהסוף.

- **נגדיר:** $a_i = w_{i-1} \cdot x$ ו $o_i = \phi(a_i)$, כלומר האיבר a הוא המכפלה לפני הפעלת האקטיבציה, ו o הא האיבר a לאחר שהפעלנו עליו אקטיביציה.

- **מה נרצה לגזור:** עבור רשת שמוגדרת באופן הבא:

$$\left(L_T \circ L_{T-1} \circ \ldots \circ L_1\right)(X)$$

נרצה לחשב את הנגזרת של כל הרשת לפי משקולת מסויימת w_i:

$$\frac{\partial}{\partial \mathbf{W}_{t-1}}\left(L_T \circ L_{T-1} \circ \ldots \circ L_1\right) = J_{\mathbf{a}_{t-1}}\left(L_t\right) \cdot \prod_{i=T}^{t+1} J_{\mathbf{o}_{i-1}}\left(L_i\right) = J_{\mathbf{a}_t}\left(\mathbf{o}_t\right) \cdot J_{\mathbf{W}_{t-1}}\left(\mathbf{a}_t\right) \cdot \prod_{i=T}^{t+1}\left[J_{\mathbf{a}_i}\left(\mathbf{o}_i\right) \cdot J_{\mathbf{o}_{i-1}}\left(\mathbf{a}_i\right)\right]$$

הערה: עבור משקולת w_i, היא לא משפיעה על כל המשקולות w_jכך ש $j < i$, ואין צורך לגזור אותן.

- **פסואודו קוד:** האלגוריתם מקבל דגימה בודדת.

Algorithm 1 Back-propagation

procedure BACK-PROPAGATION($\langle G, \{\mathbf{W}_t\}, \{\sigma_t\}, \phi\rangle, (\mathbf{x}, y)$)
  Denote $N := L_T \circ L_{T-1} \circ \ldots \circ L_1$
  FORWARD PASS
    Denote $\mathbf{o}_0 \leftarrow \mathbf{x}$
    for $t = 1, 2, \ldots, T$ **do**
      Compute pre-activations $\mathbf{a}_t \leftarrow \langle \mathbf{W}_{t-1}, \mathbf{o}_{t-1}\rangle$
      Compute activations $\mathbf{o}_t \leftarrow \sigma_t(\mathbf{a}_t)$
    end for

  BACKWARD PASS
    Set $\Delta_T = \phi'(\mathbf{o}_T)$
    for $t = T-1, T-2, \ldots, 1$ **do**
      Set derivation chain $\Delta_t \leftarrow \Delta_{t+1} \cdot J_{\mathbf{a}_t}(\mathbf{o}_t) \cdot \mathbf{W}_{t-1}$
      Set partial derivatives $\nabla_{\mathbf{W}_{t-1}} N \leftarrow \Delta_{t+1} \cdot J_{\mathbf{a}_t}(\mathbf{o}_t) \cdot \mathbf{o}_{t-1}$
    end for

  return ∇N
end procedure

תיקון: בחלק של $pre\ activation$ - $a_i = W_{t-1} \cdot o_{t-1}$ ולא מ"פ .