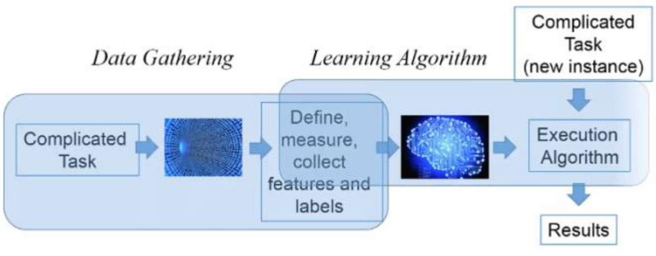
למידה חישובית ממידע: הרצאה 1 – Linear Regression



מושגים בלמידה חישובית ממידע:

- אוסף של דאטה, דגימות, טובות ורעות (חיוביות ושליליות) שיקראו samples או instances
- איסוף מידע אודות הדגימות הנ״ל, שיקרא features תכונות של הדגימות (חלקן רלוונטיות וחלקן אינן, לא בהכרח נדע מראש מה רלוונטי עבורנו ומה לא רלוונטי עבורנו לדעת)
- הפעלת <u>אלגוריתם הלמידה</u> שייצור אלגוריתם מבצע = שידע לקבל דגימות חדשות ולחשב עבורנו

סוגי למידה:

- רגרסיה: בהינתן דגימה $\{x_i, y_i\}$ נמצא פונקציה f כך ש $y=f(x)$
- קלסיפיקציה: בהינתן דגימה $\{x_i, y_i\}$ כאשר y_i הינו 0 או 1 בדאטה אימון, נקבע, עבור כל x חדש אם x שייך ל-c_0 או ל-c_1.
- Density Estimation
- Clustering

רגרסיה לינארית

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

דוגמה: בהינתן גודל של בית נרצה לקבוע מחיר עבורו. להלן דאטה אימון שלנו שמכיל 10 דגימות ו feature יחיד. מתוכו נרצה ללמוד פונקציה מכלילה. ההנחה היא שאנחנו רוצים מודל לינארי – ולכן נעשה רגרסיה לינארית – הדבר מגביל את מרחב ההיפותזות שלנו למרחב הפונקציות הלינאריות בלבד.

כיצד מייצגים את ההיפותזה h במרחב הפונקציות הלינאריות? על ידי משוואה לינארית:

$$y = \theta_0 + \theta_1 x$$

<u>בדוגמה שלנו יש פיצ׳ר בודד, מה עושים כאשר יש מספר פיצ׳רים?</u>

כאשר Xi = הוא וקטור הפיצ׳רים שלנו, עבור כל דגימה i.

$$\mathbf{x}^{(i)} = \left(x_1^{(i)}, x_2^{(i)}, \ldots, x_n^{(i)}\right)$$

למען הנוחות, נוסיף פיצ׳ר ״קבוע״ עבור כל $X_0(i) = 1$,i

עבור n פיצ׳רים, ההיפותזה הלינארית שלנו תהיה מיוצגת על ידי וקטור הפרמטרים טטה:

$$\boldsymbol{\theta} = \left(\theta_0, \theta_1, \theta_2, \ldots, \theta_n\right)$$

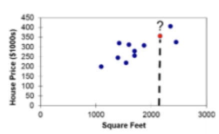
- We say that $\boldsymbol{\theta} = \left(\theta_0, \theta_1, \theta_2, \ldots, \theta_n\right)$ is the vector of parameters that defines our function (or our **model**)

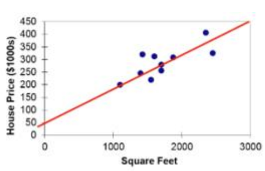
כך שמתקיים:

$$\boxed{y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n}$$

Scatter plot of House Price (y) vs House Size (x)

<u>Prediction</u>:
Given house of size x, what would be its price y = f(x)?





- We assume/hypothesize that the relationship between the observed and the independent/explained variable is linear and thereby conduct our search
- This is our **Hypotheses Space –** all linear functions

וקטור הפרמטרים טטה למעשה מגדיר את הפונקציה שלנו (המודל שלנו), וכך למעשה בממד גבוה תיראה הפונקציה שנרצה ללמוד, לכן עלינו ללמוד את הטטות. נרצה למצוא את הטטות הטובות ביותר – שיניב את המודל הטוב ביותר.

המכפלה הפנימית של טטה באיקס החדש (i) תניב לנו את מחיר הבית.

הדאטה אימון training data תעזור לנו להגיע לטטה הטובה ביותר. בעולם ״מושלם״ היינו רוצים למצוא טטה שמניבה טעות 0, כלומר ממש למצוא מודל שעובר בכל הנקודות, אלגוריתם למידה כזה נקרא consistent learner. זהו לא המקרה בעולם האמתי וכן לא בדוגמה שלנו. אבל עדיין נרצה את הטעות הקטנה ביותר.

<u>אז איך נמדוד את השגיאה שלנו?</u> הטעות בכל דגימה (instance) הינה:

$$\left(\theta_0 x_0^{(i)} + \theta_1 x_1^{(i)} + \theta_2 x_2^{(i)} + \cdots + \theta_n x_n^{(i)} - y^{(i)}\right) = \boldsymbol{\theta} \cdot \mathbf{x}^{(i)} - y^{(i)}$$

כעת נבצע ממוצע על כל דגימות האימון במטרה לקבל **את פונקציית העלות** שלנו Mean square error:

<u>cost function</u>:

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

אנו משתמשים בשגיאה הריבועית כדי שטעות בכיוונים שונים לא תתבטל, וגם זו פונקציה smoother מערך מוחלט (שאינה גזירה ב0).

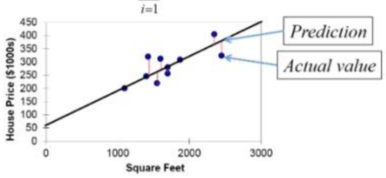
כך נוכל לחשב את הטעות עבור כל טטה וכך להעריך את איכות הטטה, על ידי חישוב השגיאה. ומכיוון שנרצה להגיע לטטה בעלת השגיאה המינימלית, **נרצה למצוא את הטטה שמניבה ערך מינימלי של הפונקציה J – פונקציית העלות**. הטטות הן המודל.

- Hence, our best hypothesis $\boldsymbol{\theta}^*$ would be the one that <u>minimizes the cost function</u>:

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

- How can we find it?

$$J(\boldsymbol{\theta}) = \frac{1}{2m}\sum_{i=1}^{m}\left(\theta_0 + \theta_1 x^{(i)} - y^{(i)}\right)^2$$



<u>השאלה, איך מביאים למינימום את פונקציית המחיר / העלות, שהיא בעלת מספר משתנים ?</u>

- הדרך הראשונה למצוא מינימום של פונקציה היא לגזור אותה ולהשוות אותה ל-0.
- הדרך השנייה : גרדיאנט דיסנט.

שתי השיטות יכולות להוביל אותנו למינימום לוקאלי, ולא גלובלי כפי שנרצה!

נזכיר שנגזרת חלקית מחושבת על ידי גזירה על פי משתנה אחד כך שכל השאר נותרים קבועים.

הנגזרת הכיוונית בכיוון u מוגדרת :

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

- The directional derivative in the direction of the vector u = (u_1,u_2) (a scalar!) can also be written as:

$$D_u f(x_1, x_2) = \nabla f \cdot \mathbf{\ddot{u}} = |\nabla f||\mathbf{\ddot{u}}|\cos\beta = |\nabla f|\cos\beta$$

- Where β is the angle between $\boldsymbol{u}$ and ∇f
- However, $\cos\beta \le 1$.
- Therefore:
  1. The greatest increase in the function happens in the direction of the gradient (<u>i.e. β =0</u>)
  2. The greatest decrease is in the direction $-\nabla f$ (i.e β =180°)

Define the **<u>GRADIENT of f:</u>**

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

הגרדיאנט של פונקציה f מוגדר להיות :

<u>משפט:</u> עבור כל כיוון u = (u1, u2) ולכל נקודה x = (x1, x2) מתקיים :

$$\boxed{D_u f(x) = \nabla f(x)\cdot u}$$

כלומר הנגזרת הכיוונית בכיוון u בנקודה x שווה למכפלה הפנימית של הגרדיאנט של הפונקציה f בנקודה x עם וקטור הכיוון u.

במטרה להגיע למינימום של פונקציה נתחיל בנקודה מסוימת ונלך נגד הגרדיאנט. תהליך זה נקרא **<u>גרדיאנט דיסנט</u>**.

גדיאנט דיסנט לא מבטיח מינימום לוקאלי! ומתאים גם לאלגוריתמים שאינם מניבים פונקציה שהיא בהכרח לינארית.

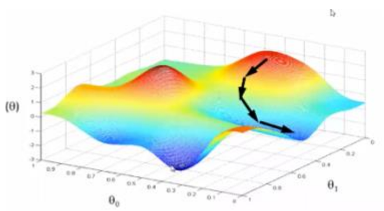
אלגוריתם גרדיאנט דיסנט Gradient Descent

- נתחיל עם ערך רנדומלי כלשהו של טטה $\boldsymbol{\theta} = (\theta_0, \theta_1, \theta_2, \ldots, \theta_n)$
- <u>עד שנגיע למינימום נבצע את הצעד הבא : (הליכה נגד הגרדיאנט)</u>

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

  עבור כל j, נעדכן את טטה j להיות :

  כלומר נעדכן את טטה j להיות טטה j עם "צעד קטן" נגד הגרדיאנט.
- אלפא היא פרמטר של האלגוריתם שנקרא learning rate גודל צעד העידכון שמוגדר.



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

  For all j,
  Update $\theta_j := \theta_j - \alpha\frac{\partial}{\partial\theta_j}J(\boldsymbol{\theta})$
- $\alpha > 0$ is a parameter of the algorithm called the <u>learning rate</u>
- Updates are simultaneous (in all $n+1$ directions)
- In the general case this process can still be trapped in local minima!

כאשר חישוב הגרדיאנט של פונקציית העלות הינו :

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

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

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

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

האלגוריתם הסופי להפעלת גרדיאנט דיסנט על פונקציית העלות (השגיאה של טטה):

- מתחילים מערך התחלתי של טטה
- חוזרים על הצעד עד תנאי העצירה : מעדכנים טטהj חדש להיות טטה j הנוכחי פחות הlearning rate (אלפא) כפול הנגזרת החלקית של פונקציית העלות ביחס לטטה j, מחושב על הטטה הנוכחי.
- נזכור ש- x0(i) = 1

- Initialize $\boldsymbol{\theta} = (\theta_0, \theta_1, \theta_2, \ldots, \theta_n)$
- Repeat until you reach a minimum (or stop cdn):
  - For all 0≤j ≤n,

    $$\theta_j := \theta_j - \alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})\cdot x_j^{(i)}$$
- In words: set the new θ_j to the current θ_j minus the learning rate (α) times the partial derivative of the error function with respect to θ_j , computed at the current θ.
- Also remember that $x_0^{(i)}$=1

אם המטריצה X הייתה ריבועית, היינו משתמשים באינברס שלה.

סודו אינברס:

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

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

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

$$\boxed{\vec{\boldsymbol{\theta}} = pinv(\mathbf{X})\overrightarrow{\mathbf{Y}} \quad \textbf{where} \quad pinv(\mathbf{X}) = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T}$$

- The n×m matrix $\boldsymbol{pinv(X)} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T$ is called the **pseudo inverse** of **X** (which is mxn)
- If **X** is square it is just its inverse.

הערות לגבי סודו אינברס:

- **יתכן מצב בו למכפלה לא יהיה אינברס, ניתן לפתור זאת עם פעולות אלגבריות אשר לא נתעמק בהן בקורס.**
- **הפינב לא יעבוד לכל פונקציית עלות j, גרדיאנט דסנט הוא כללי יותר.**
- **גרדיאנט דיסנט מאפשר פרלליזציה.**

נשתמש בטכניקה של רגרסיה גם עבור פונקציות פולינומיאליות:

$$\boxed{X\cdot\theta = y}$$

$$\begin{pmatrix} x_0^{(1)} & x_1^{(1)} & \cdots & x_n^{(1)} \\ x_0^{(2)} & x_1^{(2)} & \cdots & x_n^{(2)} \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ x_0^{(m)} & x_1^{(m)} & \cdots & x_n^{(m)} \end{pmatrix} \begin{pmatrix} \theta_0 \\ \vdots \\ \theta_n \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \\ \\ y_m \end{pmatrix}$$

- If X is square and non singular we can write $\vec{\boldsymbol{\theta}} = X^{-1}y$
- Most of the time X is overdetermined ($m \gg n$).

- $\mathbf{X}^T\mathbf{X}$ can still be singular (or not full rank) and not have an inverse. This can be resolved with some more algebra.
- This pinv technique doesn't work for all error functions J. Gradient descent is more general.
- Gradinet descent allows parallelization.

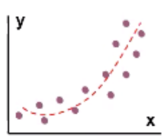
Polynomial Regression

- We can expand our feature space by using functions of the original features.
- For example, if we want to use a cubic function feature space we can define:

$$x_0 = 1, x_1 = x, x_2 = x^2, x_3 = x^3$$

then use regular regression and in essence we are learning the function

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



IDC