סיכום למידת מכונה - תשפ"ו 2026

30 באפריל 2026

גיא יער־און

הסיכום נכתב במהלך סמסטר ב' תשפ"ו (2026), ומבוסס על הרצאותיהם של פרופסור גל צצ'יק, ד"ר אורי שוהם וד"ר אורן גליקמן, לכן יתכן שנפלו טעויות בסיכום, כך שהשימוש בו על אחריותכם בלבד.

תוכן עניינים

1 הרצאה 1 . . . 2
1.1 רקע מתמטי לקורס . . . 2
1.2 הגרדיאנט ($Gradient$) . . . 6
2 הרצאה 2 . . . 8
2.1 למידה מונחית (Supervised learning) . . . 8
2.1.1 איך נוכל להפחית overfitting? . . . 11
2.2 תאוריית ההחלטה של בייס - Bayes decision theory . . . 11
2.3 The Bias-Variance Trade-off . . . 14
2.4 $Overfitting$ ו$Underfitting$ . . . 16
2.5 Cross Validation . . . 17
3 הרצאה 3 . . . 18
3.1 Parameter estimation (הערכת פרמטרים) . . . 18
3.2 Maximum Likelihood (ML) . . . 18
3.3 Maximum A Posteriori (MAP) (הסתברות פוסטריורית מקסימלית) . . . 19
3.4 Bayesian Estimation (הערכה בייאסיאנית) . . . 20
3.5 Loss-Based View . . . 21
3.6 סיכום . . . 22
3.7 PAC Learning . . . 23
3.7.1 מטרת העל של תהליך הלמידה . . . 23
3.7.2 Realizability . . . 24
3.7.3 Empirical Risk Minimization (ERM) . . . 24
3.7.4 Version Space and Bad Hypotheses . . . 24
3.7.5 אי שוויון הופדינג . . . 27
3.7.6 הטרייד אוף בין השונות לביאס . . . 27
4 הרצאה 4 . . . 27
4.1 מבוא . . . 27
4.2 רגרסיה לינארית: הגדרת הבעיה . . . 27
4.3 הגדרה מטריציונית . . . 28
4.4 מציאת הפתרון האופטימלי . . . 28
4.5 $\hat{y}$ הוא הטלה אורתוגונלית ובפרט ממזער את השגיאה . . . 30
4.6 רגרסיה פולינומיאלית . . . 30
4.7 Ridge Regression . . . 31
4.8 בפרספקטיבה של SVD . . . 32
4.9 Binary classification . . . 33
4.10 Linear-threshold neuron . . . 33
4.11 The perceptron algorithm (אלגוריתם הפרספטרון) . . . 34

4.12 Gradient Descent 35
4.13 GD במימד גבוה $\mathbb{R}^d$ 35
4.14 Stochastic Gradient Descent (SGD) 36
4.15 הטריידאוף בין GD לSGD 36
4.16 Linear separability 37
4.17 Correct Classification - הצלחה בסיווג 37
4.18 דוגמאות 38
4.19 אלגוריתם לאימון ניורון בודד באמצעות SGD ו Hinge loss finction 38
4.20 רגרסיה לוגיסטית 39
4.21 מדוע רגרסיה לוגיסטית נחשבת למסווג לינארי? 41
5 הרצאה 5 42
6 הרצאה 6 42
7 הרצאה 7 42
8 הרצאה 8 42
9 הרצאה 9 42
10 הרצאה 10 42
11 הרצאה 11 42

1 הרצאה 1

1.1 רקע מתמטי לקורס

הגדרה 1. תהי $A \in \mathbb{R}^{d\times d}$. הוקטורים $0 \neq v \in \mathbb{R}^d$ כך שהפעלת A עליהם לא תשנה את הכיוון שלהם נקראים **וקטורים עצמיים**. כלומר, מתקיים $Av = \lambda v$, כך ש$\lambda \in \mathbb{C}^d$ נקרא **ערך עצמי** ול$\frac{v}{||v||}$ נקרא וקטור עצמי המתאים לλ.

הערה 2. תמיד ננרמל את הוקטורים העצמיים.

הערה 3. בפועל, וקטור עצמי הוא וקטור שהפעלת המטריצה עליו רק משנה את גודל הוקטור, ולא את הכיוון. המטריצה משנה את הגודל בהתאם לλ, הערך העצמי.

הערה 4. באופן כללי: כאשר מטריצה A פועלת על וקטור v, מתקבל הוקטור Av. A משנה את v באופן לינארי: סיבוב + הזזה.

משפט 5. תהי $A \in \mathbb{R}^{d\times d}$ סימטרית. ויהי λ ע"ע של A. אזי, λ בהכרח ממשי.

הערה 6. לכל מטריצה $A \in \mathbb{R}^{d\times d}$ ישנם d ע"ע מעל $\mathbb{C}$ (הם לא בהכרח שונים זה מזה, ויכולים להיות מרוכבים).

הגדרה 7. המכפלה הסקלרית הסטנדרטית בהינתן $v_1 = (x_1, ..., x_n), v_2 = (y_1, ..., y_n) \in \mathbb{R}^n$ מוגדרת כך:

$$< v_1, v_2 >= v_1^T v_2 = \sum_{i=1}^{n} x_i y_i$$

הגדרה 8. יהיו שני וקטורים $v, u \in \mathbb{R}^n$.

א. נאמר כי u, v הם **אורתוגונליים** אם, ורק אם $< v, u >= v^T u = 0$

ב. נאמר כי u, v הם **אורתונורמליים** אם הם אורתוגונליים וגם $\forall v : ||v|| = \sqrt{v^T v} = \sqrt{\sum_{i=1}^{n} v_i^2} = 1$

משפט 9. תהי $A \in \mathbb{R}^{d\times d}$ סימטרית, ויהיו $0 \neq v_1, v_2 \in \mathbb{R}^d$ ו"ע השייכים לע"ע המתאימים $\lambda_1 \neq \lambda_2$. אזי, v_1 וv_2 אורתוגונליים (ולכן אורתונורמליים) זה לזה.

הוכחה: מתקיים $Av_1 = \lambda_1 v_1 \wedge Av_2 = \lambda_2 v_2$ ולכן:

$$\lambda_2 v_1^T v_2 = v_1^T \lambda_2 v_2 =_{Av_2=\lambda_2 v_2} v_1^T A v_2 =_{A=A^T} v_1^T A^T v_2 = (Av_1)^T v_2 = (\lambda_1 v_1)^T v_2 = v_1^T \lambda_1 v_2$$

$$\implies (\lambda_1 - \lambda_2) v_1^T v_2 = 0$$

כיוון ש$\lambda_1 \neq \lambda_2$ הרי ש$\lambda_1 - \lambda_2 \neq 0$ ולכן $v_1^T v_2 = 0$, כלומר $< v_1, v_2 >= 0$, ולכן v_1, v_2 אורתגונליים. כנדרש.

■

משפט 10. תהי $A \in \mathbb{R}^{d\times d}$ סימטרית. אזי,
א. A לכסינה.
ב. קייס לה בסיס אורתונורמלי של ו"ע של $\mathbb{R}^d$.

משפט 11. תהי $A \in \mathbb{R}^{d\times d}$ סימטרית בעלת הע"ע $\{\lambda_i\}_{i=1}^n$ כך ש$\forall 1 \le i \le n : \lambda_i \ge 0$, אזי לכל וקטור $z \in \mathbb{R}^d$ מתקיים $z^T Az \ge 0$.
הוכחה: A סימטרית ולכן קיים לה בסיס אורתונורמלי של וקטורים עצמיים (לפי משפט לעיל), נסמנו $\{v_1, ..., v_d\}$. מכאן, כל וקטור z ניתן לכתוב כצירוף לינארי של הו"ע כך: $z = \sum_{i=1}^d c_i v_i$, באשר $\{c_i\}_{i=1}^d$ המקדמים. כעת:

$$Az = A \times \sum_{i=1}^d c_i v_i = \sum_{i=1}^d c_i \times Av_i = \sum_{i=1}^d c_i \lambda_i v_i$$

כעת, נעבור לביטוי $z^T Az$:

$$z^T Az = (\sum_{i=1}^d c_i v_i)^T (\sum_{j=1}^d c_j \lambda_j v_j) \Longrightarrow \sum_{i=1}^d \sum_{j=1}^d c_i c_j \lambda_j (v_i^T v_j)$$

לפי הגדרת בסיס אורתנורמלי, אם $i \ne j$ הרי ש$v_i^T v_j = 0$ ואחרת $v_i^T v_i = 1$. (נורמלי). כלומר, כל האיברים בהם $i \ne j$ מתאפסים ונשאר עם האיברים בהם $i = j$, ולכן:

$$\Longrightarrow \sum_{i=1}^d c_i^2 \lambda_i \ge 0$$

שכן, $\{\lambda_i\}_{i=1}^n \ge 0$, ולכן סה"כ קיבלנו $z^T Az \ge 0$, כנדרש.

■

הגדרה 12. תהי $A \in \mathbb{R}^{d\times d}$. אזי,
1. נאמר כי A היא PSD (Positive Semi Definite) אם לכל $0 \ne z \in \mathbb{R}^d$ מתקיים $z^T Az \ge 0$
2. נאמר כי A היא PD (Positive Definite) אם לכל $0 \ne z \in \mathbb{R}^d$ מתקיים $z^T Az > 0$

משפט 13. תהי $A \in \mathbb{R}^{d\times d}$ סימטרית. אזי, כל הע"ע של A הם אי שליליים אמ"מ A היא PSD. בדומה, כל הע"ע של A הם חיוביים ממש אמ"מ A היא PD.
הוכחה:
$\Longleftarrow$ לפי משפט 10.
$\Longrightarrow$ יהי λ ע"ע עם ו"ע תואם לו v המקיים $||v|| = 1$ (מנורמל). אזי,

$$0 \le v^T Av = v^T \lambda v = \lambda v^T v = \lambda ||v||^2 = \lambda \cdot 1 = \lambda$$

כנדרש.

■

ñî .14÷מסקנה ממשפט 13: מטריצת PSD היא גם סימטרית, וגם כל הערכים העצמיים שלה אי שליליים.

äøãâää 15. תהי מטריצה $A \in \mathbb{R}^{n\times n}$. נאמר כי A **אורתוגונלית** אם $A^T A = AA^T = I$.

16. aÿòà¨ בפועל, במטריצה אורתגונלית הכוונה כי כל זוג שורות ועמודות במטריצה הם אורתוגונליים זה לזה ואורתנורמליים (מאונכים, בגודל וקטור יחידה).

הערה 17. אם נפעיל וקטור על מטריצה אורתוגונלית וגם PSD - מה יקרה? נזכר כי הע"ע של מטריצה אורתוגונלית הם ± 1. כיוון שהיא PSD, אזי רק 1 הוא ערך עצמי. לכן לא יקרה כלום.

משפט 18. (**כלל כייס**). יהיו A, B זוג מאורעות. אזי, $Pr(A|B) = \frac{Pr(B|A)P(A)}{Pr(B)}$

הגדרה 19. יהיו 2 משתנים מקריים, X וY. כך שX מקבל ערכים $x_1, ..., x_k$ עם ההסתברות $Pr(X = x_i)$ וY מקבל ערכים $y_1, ..., y_k$ עם ההסתברות $Pr(Y = y_i)$. אזי, נגדיר את ההתפלגות המשותפת שלהם $Pr(X = x_i, Y = y_i)$. נוכל להכליל זאת עבור d משתנים מקריים $\{X_i\}_{i=1}^d$, כך שכל X_i מקבל ערכים $X_i^1, ..., X_i^{k_i}$ **וההתפלגות המשותפת** שלהם הינה:

$$Pr(X_1 = x_1, ..., X_d = x_d)$$

כמו כן, נוכל להגדירו כמשתנה מקרי יחיד רב ממדי, וקטור $X = (X_1, ..., X_d)$ ולשאול מתי הוא מקבל את הערך $(x_1, ..., x_d)$.

הגדרה 20. נגדיר את התוחלת של משתנה מקרי רב ממדי להיות וקטור רב ממדי של התוחלות כך:

$$\mu = \mathbb{E}[X] = (\mathbb{E}[X_1], ..., \mathbb{E}[X_d])$$

הגדרה 21. יהיו X, Y זוג משתנים מקריים. נגדיר את ה$Covariance$ שלהם להיות השונות המשותפת שלהם, בצורה הבאה:

$$Cov(X,Y) = \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])] = \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y]$$

מדד זה בודק כמה ישנו קשר בניהם, בפרט אם $Cov(X,Y) > 0$ אזי ישנו קשר חיובי בין השניים (עולים יחד), אם $Cov(X,Y) < 0$ אזי ישנו קשר שלילי בין השניים (אחד עולה, השני יורד) ואם $Cov(X,Y) = 0$ אזי אין קשר לינארי בניהם.

משפט 22. כל ההבאים מתקיימים:
א. $Cov(X,X) = Var(X)$
ב. $Cov(X,Y) = Cov(Y,X)$
ג. $Var(X+Y) = Var(X) + Var(Y) + 2Cov(X,Y)$
ד. אם X, Y ב"ת אזי הם בלתי מתואמים.

הגדרה 23. נגדיר את מטריצת ה$Covariance$ באופן הבא:

$$\begin{pmatrix} Cov(X,X) & Cov(X,Y) \\ Cov(Y,X) & Cov(Y,Y) \end{pmatrix}$$

ונבחין כי היא מטריצה סימטרית.

הגדרה 24. יהי $X = (X_1, ..., X_d)$ משתנה מקרי רב ממדי, נגדיר את מטריצת ה$Covariance$ כך:

$$\Sigma = Cov(X) = \begin{pmatrix} Cov(X_1,X_1) & \cdots & \cdots & Cov(X_1,X_d) \\ & Cov(X_2,X_2) & & \\ \vdots & & \ddots & \vdots \\ Cov(X_1,X_{d-1}) & & & \\ Cov(X_1,X_d) & \cdots & & Cov(X_d,X_d) \end{pmatrix} = \mathbb{E}[(X - \mathbb{E}[X])(X - \mathbb{E}[X])^T]$$

הערה 25. לכל $i, j \in [n]$ מתקיים כי $\Sigma_{ij} = Cov(X_i, X_j)$ כאשר בתא זה של המטריצה ישנו ערך ה$Covariance$ בין שני המשתנים.

הערה 26. המטריצה Σ היא מטריצה סימטרית, לכן היא לכסינה אורתוגונלית, כמו כן, המטריצה היא PSD ולכן כל הע"ע שלה הם אי שליליים לפי משפט שראינו קודם.

משפט 27. מטריצת Σ היא PSD, כלומר לכל וקטור $z \in \mathbb{R}^d$ שונה מאפס מתקיים $z^T \Sigma z \geq 0$.
הוכחה:

$$z^T \Sigma z =_{def} z^T \mathbb{E}[(X-\mu)(X-\mu)^T] z$$

כעת, כיוון שz הוא וקטור קבוע, הרי מתקיים עבור קבוע $\mathbb{E}[aX] = a\mathbb{E}[X]$ ותכונה זו נכונה גם עבור וקטור קבוע, ולכן נוכל להכניסו:

$$= \mathbb{E}[z^T (X-\mu)(X-\mu)^T z]$$

נבחין כי $z^T(X-\mu)$ הוא מכפלה סקלרית (וקטור שורה, כפול וקטור עמודה), לכן תוצאתו סקלר. הביטוי השני, הוא בדיוק טרנספוז של סקלר זה. כיוון שטרנספוז של סקלר שווה לסקלר עצמו, הרי שנוכל לכתוב כי:

$$= \mathbb{E}[(z^T(X-\mu))(z^T(X-\mu))^T] = \mathbb{E}[(z^T(X-\mu))^2] \geq 0$$

שהרי המעבר האחרון נובע מכך שאם מ"מ הוא אי שלילי, גם התוחלת שלו היא אי שלילית. כנדרש.

■

הגדרה 28. יהי $X \sim N(\mu, \sigma^2)$ משתנה מקרי גאוסיאני (נורמלי), אזי פונקציית הצפיפות שלו מקיימת:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\cdot(\frac{x-\mu}{\sigma})^2}$$

הגדרה 29. יהי X וקטור רב ממדי, כך ש$X \in \mathbb{R}^d$, המתפלג גאוסיאנית $X \sim N(\mu, \Sigma)$, אזי פונקציית הצפיפות שלו מקיימת:

$$f(x) = \frac{1}{(2\pi)^{\frac{d}{2}}\sqrt{det(\Sigma)}} e^{-\frac{1}{2}\cdot(x-\mu)^T\,\cdot\,\Sigma^{-1}\,\times\,(x-\mu)}$$

באשר x הוא וקטור הקלט, μ הוא וקטור התוחלת, Σ היא מטריצת ה$Covariance$ (קובעת את "צורת" הפעמון), $det(\Sigma)$ היא הדטרמיננטה של המטריצה. הביטוי במעריך מחליף את הריבוע בנוסחה המקורית, והוא מודד כמה רחוק x מהמרכז μ, אך הוא מתחשב ב$Covariance$. מרחק זה נקרא Mahalanobis Distance.

הערה 30. מאוד חשוב לשים לב - משתנה רב ממדי מתפלג נורמלית כאשר Σ מייצגת את השונות, נבחין כי למשתנה רב ממדי אין שונות כערך מספרי בודד, אלא מטריצת $Covariance$ שעל האלכסון מייצגת את השנויות של הערכים $\{X_i\}_{i=1}^d$.

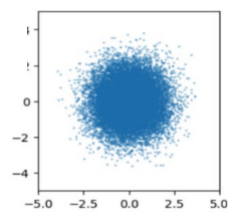
ישנם כמה מקרים פרטיים שנרצה לדון בהם:

1. **מקרה ראשון - $\Sigma = I$:** במקרה זה יתקיים,

$$f(x) = \frac{1}{(2\pi)^{\frac{d}{2}}} e^{-\frac{1}{2}\cdot||(X-\mu)||^2}$$

כאן כל השנויות שוות לאחד, וכן כל זוג מימדים $x_i \neq x_j$ בלתי מתואמים, כלומר $Cov(X_i, X_j) = 0$. לבסוף, קיבלנו כי ההסתברות של הוקטור הנ"ל תלויה אך ורק במרחק הגאומטרי הישיר שלה מהמרכז (μ). יש צפיפות גבוהה מאוד סביב התוחלת, ואם נחפש את כל הנקודות שיש להן את אותה הסתברות נקבל את המשוואה $||X-\mu||^2 = c$ באשר במרחב דו ממדי מדובר במעגל, ובמרחב רב ממדי על כדור.

לדוגמה: נקודות שנלקחות מתוך התפלגות המקיימת $\Sigma = I$, ענן הנקודות נראה כמו עיגול סימטרי, הסיבה היא כי על האלכסון יש רק 1 ולכן השונות של x שווה בדיוק לשונות של y: הפיזור לגובה שווה לפיזור לרוחב. כיוון שהערכים מחוץ לאלכסון הם אפס, אין קשר בין המשתנים, לכן הם בלתי מתואמים ולכן ענן" הנקודות לא נוטה באלכסון. כמו כן, רוב הנקודות קרובות ל$(0,0)$, כיוון שכאן מתקיים $\mu = (0,0)$.



2. **מקרה שני - Σ היא מטריצה אלכסונית:** כלומר: $\Sigma = \begin{pmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ \vdots & \sigma_2^2 & & \vdots \\ 0 & & \ddots & 0 \\ 0 & \cdots & 0 & \sigma_d^2 \end{pmatrix}$. נפתח אלגברית מעט:

$$f(x) = \frac{1}{(2\pi)^{\frac{d}{2}}\sqrt{det(\Sigma)}} e^{-\frac{1}{2}\cdot(x-\mu)^T\,\cdot\,\Sigma^{-1}\,\times\,(x-\mu)} = \frac{1}{(2\pi)^{\frac{d}{2}}\sqrt{\prod_{i=1}^d \sigma_i^2}} e^{-\frac{1}{2}\cdot(x-\mu)^T\,\cdot\,\begin{pmatrix} \frac{1}{\sigma_1{}^2} & 0 & \cdots & 0 \\ \vdots & \frac{1}{\sigma_2{}^2} & & \vdots \\ 0 & & \ddots & 0 \\ 0 & \cdots & 0 & \frac{1}{\sigma_d{}^2} \end{pmatrix} \times (x-\mu)}$$

$$= \prod_{i=1}^d \frac{1}{\sqrt{2\pi\sigma_i^2}} \cdot exp(-\frac{(x_i-\mu_i)^2}{2\sigma_i^2})$$

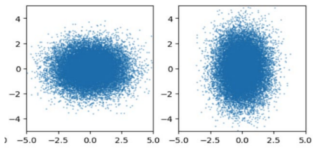
נתעכב להסביר את החלק הכי פחות טריוואלי, המעריך. הרי $x-\mu$ הוא וקטור, נוכל לסמן את ערכיו על מנת שיהיו לנו חיים קלים ע"י $\{x_i-\mu_i\}_{i=1}^d$. מכאן, $\begin{pmatrix} \frac{1}{\sigma_1 2} & 0 & \cdots & 0 \\ \vdots & \frac{1}{\sigma_2 2} & & \vdots \\ 0 & & \ddots & 0 \\ 0 & \cdots & 0 & \frac{1}{\sigma_d 2} \end{pmatrix} \times (x-\mu) = \begin{pmatrix} \frac{x_1-\mu_1}{\sigma_1^2} \\ \vdots \\ \vdots \\ \frac{x_d-\mu_d}{\sigma_d^2} \end{pmatrix}$. ומכאן, מכפילים ב$(X-\mu)^T$ שהרי מדובר בוקטור שורה, שמקבל את הערכים $\{x_i-\mu_i\}_{i=1}^d$, מכאן מדובר במכפלה סקלרית:

$$(x_1-\mu_1,.....,x_d-\mu_d) \times \begin{pmatrix} \frac{x_1-\mu_1}{\sigma_1^2} \\ \vdots \\ \vdots \\ \frac{x_d-\mu_d}{\sigma_d^2} \end{pmatrix} = \prod_{i=1}^{d} \frac{(x_i-\mu_i)^2}{\sigma_i^2}$$

לאחר מכפלה ב$-\frac{1}{2}$, מקבלים את הדרוש.

שוב, תהיה צפיפות סביב μ גבוהה מאוד, אך הפעם בדו מימד מדובר בצורת אליפסה (ולא במעגל) וברב מימד צורות מעניינות אף יותר ככל שהמימד ימשיך לגדול.

דוגמה: כאן, אכן מדובר במטריצה אלכסונית. בדוגמה משמאל נראה כי שונות x היא 2, והשונות בציר הy היא 1 ולכן ציר הx נראה מתוח יותר לצדדים וקיבלנו אליפסה אופקית. בדוגמה מימין, זה בדיוק הפוך, קיבלנו אליפסה אנכית כיוון שהשונות בציר הy גדולה מהשונות בציר הx.



$\Sigma = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$ $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$

משפט 31. יהי $X=(X_1,...,X_d)$ משתנה מקרי רב ממדי גאוסיאני $X \sim N(\mu,\Sigma)$. אזי, $X_i|X_j$ גאוסיאן (כלומר, גם אם יודעים ערך של אחד מ$\{X_i\}_{i=1}^d$ אזי עדיין המשתנה הרב ממדי ממימד קטן יותר הוא גאוסיאן, וכן $\int_{-\infty}^{\infty} f_X(x_1,...,x_d)dx_1$ גאוסיאן גם כן.

משפט 32. אם מטריצת ה$Covariance$ Σ היא מטריצה אלכסונית (כלומר לכל $i \neq j$ מתקיים $Cov(X_i,X_j)=0$) אזי $\{X_i\}_{i=1}^n$ הם ב"ת.

הערה 33. לא כטענה אך לשים לב: כל תזוזה לינארית (כולל הכפלה במטריצה וכו') של גאוסיאן היא גאוסיאן.

1.2 הגרדיאנט $(Gradient)$

הערה 34. מהו הגרדיאנט? מדובר בוקטור, ש"זוכר" את כל הנגזרות החלקיות של פונקציה בכמה משתנים. במקום להסתכל רק על כיוון אחד, הגרדיאנט מייצג את השיפוע של הפונקציה בכל הכיוונים בבת אחת. הוא אומר לנו באיזה כיוון הפונקציה עולה הכי מהר ובאיזו עוצמה. ופורמלית:

הגדרה 35. תהי פונקציה $f(x_1,x_2,.....,x_d)$ המוגדרת בתחום $f:\mathbb{R}^d \to \mathbb{R}$. נגדיר את הגרדיאנט של f כך:

$$\nabla f(x) = [\frac{\vartheta f}{\vartheta x_1},.....,\frac{\vartheta f}{\vartheta x_d}]$$

הגדרה 36. תהי f פונקציה, נגדיר את **קצב השינוי של הפונקציה** f בנקודה x לאורך כיוון u **כנגזרת הכיוונית** של f בכיוון u (כאשר u הוא וקטור יחידה כלומר $||u||=1$) כך:

$$D_u f(x) = lim_{h\to 0}\frac{f(x+hu)-f(x)}{h}$$

משפט 37. תהי $f:\mathbb{R}^d\to\mathbb{R}$ פונקציה גזירה בנקודה $x\in\mathbb{R}^d$. אזי, קצב השינוי של הפונקציה f בנקודה x בכיוון וקטור יחידה u הוא מקסימלי כאשר u בכיוון הגרדיאנט.

הוכחה: תהי f פונקציה גזירה, אזי הקירוב עבורה מסדר ראשון הינו:

$$f(x+hu)\approx f(x)+\nabla f(x)\cdot(hu)+\varepsilon(h)$$

באשר $\varepsilon(h)$ מסמלת שארית ששואפת לאפס באשר $h\to 0$. מכאן, נציב בהגדרת הנגזרת הכיוונית:

$$D_uf(x)=lim_{h\to 0}\frac{f(x)+\nabla f(x)\cdot(hu)+\varepsilon(h)-f(x)}{h}=lim_{h\to 0}\frac{\nabla f(x)\cdot(hu)+\varepsilon(h)}{h}=$$

$$lim_{h\to 0}(\varepsilon+\nabla f(x)\cdot u)=\nabla f(x)\cdot u$$

מסקנה: קצב השינוי של הפונקציה בכיוון u הוא המכפלה הסקלרית $\nabla f(x)\cdot u$. נעזר באי שוויון שוורץ: $a\cdot b\le||a||\cdot||b||$ ונקבל:

$$D_uf(x)=\nabla f(x)\cdot u\le||\nabla f(x)||\cdot||u||$$

אנו מניחים כי u וקטור יחידה והרי $||u||=1$ ומכאן:

$$D_uf(x)\le||\nabla f(x)||$$

כלומר, הערך המקסימלי של $D_uf(x)$ חסום ע"י $\nabla f(x)$ ויש שוויון לפי א"ש שוורץ רק כאשר הוקטורים תלויים זה בזה, מכאן: הפונקציה משתנה הכי מהר בכיוון $\nabla f(x)$.

■

משפט 38. נגדיר את הפונקציה $f(x)=x^Tx=||x||^2$. אזי, $\nabla f(x)=2x$.

הוכחה: נבחין כי אם נסמן $x=(x_1,....,x_d)$ הרי שמתקיים לכל $1\le i\le d$ כי:

$$\frac{\vartheta}{\vartheta x_i}x^Tx=\frac{\vartheta}{\vartheta x_i}(\sum_{j=1}^{d}x_j^2)=2x_i$$

ולכן, $\nabla f(x)=(2x_1,....,2x_d)=2x$.

■

הגדרה 39. מטריצת הסיאן $(Hassian)$ היא מטריצה ריבועית שמגדירה את הנגזרות החלקיות מסדר שני. כלומר,

$$H_f=\nabla^2f=\begin{pmatrix}\frac{\vartheta^2f}{\vartheta^2x_1} & & & \frac{\vartheta^2f}{\vartheta x_1x_d}\\ \cdot & & & \\ \cdot & & & \\ \frac{\vartheta^2f}{\vartheta x_dx_1} & \dots & \dots & \frac{\vartheta^2f}{\vartheta x_d^2}\end{pmatrix}$$

משפט 40. נגדיר את התבנית הריבועית $f(x)=\frac{1}{2}x^TAx$ כאשר A היא מטריצה סימטרית ממימד d. אזי, $\nabla^2f=A$.

הוכחה: בשלב ראשון נראה כי:

$$\frac{\vartheta f}{\vartheta x_k}(\frac{1}{2}\cdot\sum_{i=1}^{n}\sum_{j=1}^{n}a_{ij}x_ix_j)=$$

$$\frac{\vartheta f}{\vartheta x_k}(\frac{1}{2}(\sum_{i,j\neq k} a_{ij}x_ix_j + \sum_{i=k,j\neq k} a_{kj}x_kx_j + \sum_{j=k,i\neq k} a_{ik}x_ix_k + a_{kk}x_k^2)) =$$

$$\frac{1}{2}(\sum_{j\neq k} a_{kj}x_j + \sum_{i\neq k} a_{ik}x_i + 2a_{kk}x_k) = \frac{1}{2}(\sum_{i=1}^{d} a_{ik}x_i + \sum_{j=1}^{d} a_{ki}x_i) =_* \sum_{i=1}^{d} a_{ik}x_i = (Ax)_k$$

המעבר $*$ נכון כיוון שA סימטרית ולכן $a_{ik} = a_{ki}$. מכאן,

$$\nabla f = \begin{bmatrix} (Ax)_1 \\ (Ax)_2 \\ .. \\ .. \\ (Ax)_d \end{bmatrix} = Ax$$

ומכאן, $H_f = \nabla^2 f = \frac{\vartheta f}{\vartheta x}(\nabla f(x)) = \frac{\vartheta f}{\vartheta x}(Ax) = A$.

■

הגדרה 41. פונקציה היא קמורה אם הקטע שמחבר כל זוג נקודות על גרף הפונקציה נמצא מעל או על הגרף עצמו. פורמלית, לכל $0 \leq t \leq 1$ ולכל x, y:

$$f(tx + (1-t)y) \leq tf(x) + (1-t)f(y)$$

הערה 42. אנחנו נעדיף פונקציות הפסד קמורות, כיוון שבהן כל מינימום מקומי הוא בהכרח מינימום גלובלי. כלומר - אם מצאת נקודה בה השיפוע הוא אפס (נקודת מינימום מקומית), היא בהכרח תהיה הנקודה הנמוכה ביותר בכל הפונקציה.

משפט 43. תהי f פונקציה גזירה פעמיים בתחום קמור. f היא פונקצייה קמורה אם ורק אם מטריצת ההסיאן שלה $H_f(x)$ היא PSD.

2 הרצאה 2

2.1 למידה מונחית (Supervised learning)

המטרה שלנו תהיה לנסות ללמוד איזושהי "חוקיות" מהעולם. בשביל שנוכל ללמוד חוקיות - אנו זקוקים למידע, $data$.

נתוני האימון - (Training Data): בהינתן n זוגות של דגימות מהצורה $\{(x_i, y_i)\}_{i=1}^n$, כך ש$\forall 1 \leq i \leq n : x_i \in X, y_i \in Y$, כך ש$x_i$ הוא הקלט שלקוח מעולם קלטים X וכן y_i הוא הפלט (או התווית) מתוך עולם של פלטים שנקרא Y. לדוגמה: x_i הוא תמונה, וy_i יכול להיות התיאור "חתול". ישנה הנחת יסוד לפיה קיימת התפלגות כלשהי $P(X,Y)$ אינה ידועה לנו שממנה מגיעים זוגות אלו. התפלגות זו מייצגת פונקציה $f : X \to Y$ שאותה נרצה למצוא.

המטרה: נרצה למצוא פונקציה $h : X \to Y$ שתקרא היפותזה, היא תהיה ההיפותזה שלנו: היא "תחקה" את הפונקציה הנסתרת f בצורה הכי טובה שאפשר. לשם כך נגדיר את המדד הבא -

הגדרה 44. מדד ההפסד ($Loss$): נגדיר מדד הפסד ונסמנו $\ell(h(x), y)$ - מדד זה מודד חוסר התאמה בין מה שחזינו $(h(x))$ לבין המציאות (y). בפועל, נחפש פונקציה h שתביא למינימום את הExpected Loss - הממוצע המשוקלל של הטעויות שלנו על פני כל המקרים האפשריים לפי ההתפלגות $P(x,y)$ ומוגדר כך:

$$\mathbb{E}_{P(X,Y)}[\ell(h(x), y)] = \sum_{(x,y)} \ell(h(x), y) \cdot p(x, y)$$

הגדרה 45. (מרחב ההיפותזות): נסמן ב$\mathcal{H}$ את מרחב ההיפותזות - מדובר במשפחה מוגדרת של פונקציות שממנה אנחנו "בוחרים" את פונקציית ההפסד. כל פונקציה $h \in \mathcal{H}$ במרחב זה מאופיינת ע"י פרמטרים שנקראים w, אנו מחפשים $h \in \mathcal{H}$ הטובה ביותר שקיימת במרחב ההיפותזות.

את תהליך הלמידה נוכל לחלק לשני חלקים מרכזיים, נפרדים ומשלימים:

1. **Training Phase** (**שלב האימון**): זהו השלב שבו המחשב "לומד". אנו נותנים לאלגוריתם הלמידה את נתוני האימון שלנו - זוגות $\{(x_i, y_i)\}_{i=1}^n$. כפלט, האלגוריתם מוציא מודל אותו נסמן בh_w. המשמעות היא שהw שנוסף לh מייצג את הפרמטרים שהאלגוריתם מצא כדי שהפונקציה h תתאים בצורה הטובה ביותר לנתונים.
2. **inference Phase** (**שלב ההסקה/יישום**): זהו השלב שבו אנחנו משתמשים במה שלמדנו על נתונים חדשים שלא ראינו קודם לכן. אנו לוקחים את המודל הנלמד h_w ומפעילים אותו על קלט חדש x, אנו חוזים את הy המתאים למעשה ע"י חישוב פשוט של הפונקציה $h_w(x)$.

הערה 46. בעיות של למידה מונחית מחולקות לרוב לפי סוג הפלט (Y). כאשר סוג הפלט הוא משתנה קטגורי בדיד נקרא לבעיה מסוג Classification (בעיה מסוג סיווג). כאשר סוג הפלט הוא משתנה רציף נקרא לבעיה מסוג Regression (רגרסיה).

לצורך דוגמה, נוכל לבחור באופן ספציפי את מרחב ההיפותזות $\mathcal{H}$ להיות פולינומים מדרגה d. פורמלית, נגדיר את $h_w(x)$ כסכום ממושקל של חזקות x, כך:

$$h_w(x) = w_0 + w_1 x + w_2 x^2 + ... + w_d x^d$$

כאשר w־ים (המשקולות) הם הפרמטרים של המודל שאנו מחפשים בשלב האימון.
למה שנרצה לבחור את משפחת הפולינומים? מדובר בסט עשיר מאוד - שיכול לקרב כמעט כל פונקציה שהיא רציפה בצורה טובה. כלומר, הקירוב יהיה טוב. נוכל למצוא פרמטרים $\{w_i\}_{i=1}^d$ שיהיו קרובים מספיק לכלל.
כעת, לאחר שבידינו (למשל) פונקציה כנ"ל $h_w(x)$. כיצד נוכל לבדוק שהיא אכן מתאימה לנתונים שבידינו?

הגדרה 47. נגדיר מדד הפסד ($Loss$) בשם **Squared Loss** (**הפסד ריבועי**) שיעבוד בצורה הבאה: עבור דגימה בודדת (x_i, y_i) נגדיר את **ההפסד האמפירי** (**Empirical Loss**) ע"י הפונקציה:

$$\ell(w, x_i, y_i) = (h_w(x_i) - y_i)^2$$

ההפרש בניהם מועלה בריבוע על מנת שכל טעות תהיה חיובית, ועל מנת לתת קנס" על טעויות גבוהות יותר.
נגדיר את **מודל הטעות הכולל** (**The Overall Error**) כממוצע הטעויות הריבועיות, ונסמן פונקציה זו ב$Err(w)$ (או בMSE):

$$Err(w) = \frac{1}{n}\sum_{i=1}^{n}(h_w(x_i) - y_i)^2$$

כיצד נמדוד איכות של פתרון? נרצה לבדוק כמה המודל (ההיפותזה $\mathcal{H}$) מתאימה לנתונים. ראשית נתבונן במקרה פשוט, בו המודל הוא מדרגה $d = 0$. דהיינו, $h_w(x) = w_0$.
נראה כי הפרמטר של המודל הוא יחיד w_0. נרצה לפתור את בעיית האופטימיזציה הבאה:

$$min_{w_0}\{Err(w_0)\} = min_{w_0}\{\frac{1}{n}\sum_{i=1}^{n}(w_0 - y_i)^2\}$$

נראה שמדובר בפונקציה ריבועית בw_0, ונחפש את המינימום שלה. לכן נגזור לפי w_0:

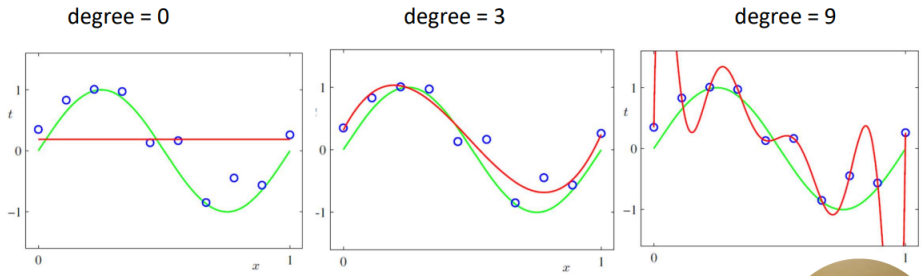
$$\frac{\vartheta Err(w)}{\vartheta w_0} = \frac{1}{n}\sum_{i=1}^{n} 2(w_0 - y_i) \Longrightarrow$$

נשווה לאפס, ונקבל: $\sum_{i=1}^{n}(w_0 - y_i) = 0$, כלומר קיבלנו $n \cdot w_0 - \sum_{i=1}^{n} y_i = 0$ ולכן:

$$w_0 = \frac{\sum_{i=1}^{n} y_i}{n}$$

כלומר, קיבלנו כי **הערך האופטימלי עבור מודל מדרגה** 0 **הוא הממוצע האריתמטי של כל** $\{y_i\}_{i=1}^n$.

נראה, כי בסבירות גבוהה שמודל מדרגה 0 לא יביא למינימום את הLoss של המודל. לכן, מה שנרצה הוא להמשיך ולהעלות את חזקת הפולינום. בצורה הבאה:



נבחין כי פולינום מדרגה 0 מביא שגיאה שונה מפולינום מדרגה גבוהה מדי (כמו 9). פולינום מדרגה 0 מביא פשוט ערך קבוע - סוג השגיאות שהוא עושה הן קבועות, נראה כי לכל x משמאל הוא חוזה ערך נמוך מדי, ולכל x מימין הוא חוזה ערך גדול מדי. ישנה הטיה ($Bias$).

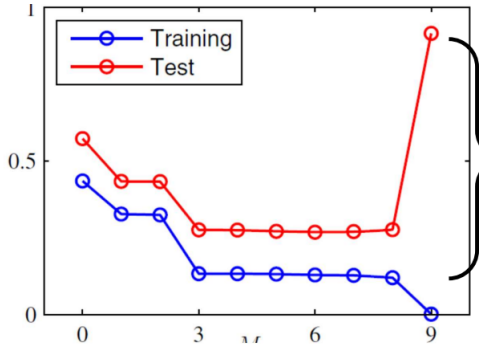
עם זאת, השגיאות במעלה גבוהה הרבה יותר מעניינות - המודל מביא שגיאה מזערית. אך למה אינטואטיבית הוא מרגיש לנו מודל "לא טוב"? כי אנחנו רואים שהקו האדום שונה מהקו הירוק. נבחין כי אנחנו נקבל במקרה זה שונות גבוהה מאוד! יתרה מזאת - השגיאות שהמודל הזה עושה תלויות ברעש של הדגימה (כמה "רעש" נכנס לנתונים). מה הבעיה? המודל הנ"ל למד יותר מדי את הרעש מהנתונים - הוא משנן את הרעש שיש בנתונים, הוא מסוגל לחזות את הנתונים שהוא כבר ראה באופן מושלם, אבל זה נובע מכך שהוא שינן ($Overfitting$) את הרעש ולא רק את המידע עצמו. זו תופעה מעניינת מאוד - שחושפת לנו את ההבדל בין שגיאת האימון לבין סוג אחר של שגיאה - **Generalization Error:** "**טעות ההכללה**" שמוגדרת כך:

$$Err_{GEN}(w) = \mathbb{E}_{P(x,y)}[\ell(h_w; x, y]$$

אנחנו הרי לא מעוניינים להקטין את הLoss על סט הנקודות עליהם אנחנו "מתאמנים", אלא נרצה להקטין את תוחלת הLoss על כלל ערכי (x, y) מתוך ההתפלגות. הרי - המטרה שלנו היא שהוא יצליח לחזות היטב לא רק את הנקודות שאנחנו נתנו לו, אלא כל זוג נתונים שהוא יקבל בעתיד מתוך $\mathcal{H}$. כלומר, אם נסכם:

אנו לא מכירים את $P(x, y)$ ולכן לא יכולנו למזער את כלל האוכלוסייה - זה שניסינו למזער את נתוני האימון (הערכים שידועים לנו) מוביל לשתי בעיות מרכזיות:

1. **ביצועים ירודים -** המודל עלול ללמוד "רעש" או תכונות ספציפיות שקיימות רק בn הדגימות שהבאנו לו ולא קיימות בחוקיות האמיתית של העולם.
2. **אופטימיות יתר -** טעות האימון תהיה תמיד נמוכה יותר מטעות ההכללה, אנו חושבים שהמודל שלנו מצויין כיוון שהוא חוזה מעולה את נתוני האימון אך הוא עלול להכשל על נתונים חדשים. כלומר הבעיה שלנו היא שהביצועים שלנו גרועים - ואנחנו לא יודעים זאת!



נתבונן בדוגמה הבאה בהמשך לדוגמה על הפולינומים:

נראה כי ישנן 2 סוגי טעויות. טעות המבחן מסומנת באדום, ושגיאת האימון בכחול. בעוד שטעות האימון תמשיך תמיד לרדת ככל שהמודל נעשה מורכב יותר (במקרה זה, עם הפולינומים!) , טעות המבחן תתחיל לעלות בשלב מסויים - העליה הזו בטעות המבחן היא הסימן לכך שהמודל הפסיק ללמוד חוקיות כללית ופשוט התחיל לשנן את נתוני האימון. כלומר: טעות המבחן עוזרת לנו לזהות $overfitting$.

הערה 48. אם נניח כי למודל היו 10 נקודות: $\{(x_i, y_i)_{i=1}^{10}\}$, אזי לפולינום מדרגה 9 ישנן 10 משקולות: כלומר, יש לנו 10 נקודות ועלינו למצוא 10 מקדמים - לכן אפשר למצוא ממש את הפולינום עצמו, ולכן נקבל בגרף כי שגיאת האימון כאשר דרגת הפולינום 9 הוא אפס, כי הצלחנו לחזות את הפולינום בדיוק.

$Test$: כפי שאמרנו, נרצה שהמודל שלנו ידע להתמודד עם נתונים שלא ראה. זהו שלב המבחן שלו. פעמים רבות נחזיק data test, דטה שמיועד למען בדיקה של המודל. אנחנו משתמשים במבחן על מנת להעריך כיצד המודל יתפקד במציאות.

2.1.1 איך נוכל להפחית overfitting?

נציע כמה וכמה פתרונות.

1. להוסיף עוד $data$**:** כשיש לנו מעט דגימות, "רעש" מקרי יכול להראות כמו חוקיות אמיתית. ככל שנוסיף יותר דוגמאות הרעש יתחיל להתקזז והחוקיות האמיתית תבלוט יותר. יתרה מזאת - אם נתן למודל 3 נקודות הפולינום יוכל להתפתל בניהן באלפי דרכים מוזרות. אם נתן לו 1000 נקודות, הוא יהיה חייב "להתיישר" לפי המגמה הכללית של כולן כדי לא לצבור שגיאה גדולה מדי.

2. $Regularization$**:** כאשר אנו מבצעים $overfitting$ המודל מנסה לעבור בדיוק דרך כל נקודת רעש בנתוני האימון. כדי לעשות את הזיג־זגים האדירים הללו המקדמים של הפולינום הופכים להיות גדולים מאוד בערכם המוחלט. לכן, הרעיון הוא להוסיף מעין קנס למודל על כך שמקדמיו גבוהים מדי. אנו משנים את פונקציית המטרה שלנו כך שהיא לא רק תנסה למזער את הטעות, אלא גם תדאג לשמור על המקדמים קטנים. כך:

$$Err(w) = \sum_{i=1}^{n}(h_w(x_i) - y_i)^2 + \frac{\lambda}{2} \cdot ||w||^2$$

החלק השני הוא למעשה $||w||^2 = w^T w = \sum_{i=1}^{d} w_i^2$. החלק הנ"ל מעוניין שהמקדמים יהיו כמה שיותר קרובים לאפס. תפקיד הλ הוא להיות "הבורר": אם λ קטן מאוד, אנו כמעט ולא קונסים את המודל - הוא ימשיך לעשות $overfitting$ על מנת להוריד את ה$Loss$. אם λ גדול מאוד - הקנס הופך להיות ממש דומיננטי: המודל יפחד לקבוע מקדמים גבוהים ויעדיף לקבל טעות אימון קצת גבוהה יותר, העיקר שהמקדמים ישארו קטנים.

2.2 תאוריית ההחלטה של בייס - Bayes decision theory

לרוב בלמידת מכונה, נרצה לנחש את הקשר בין הקלט לפלט, x וy מתוך נתונים ידועים. נניח לרגע ויש לנו ידע אלוהי - ובו אנו יודעים בדיוק את פונקציית ההסתברות המשותפת של x וy. כלומר אנו יודעים את $\mathcal{D} = P(x, y)$. בהינתן ידע מלא על הבעיה, מהי האסטרטגיה האופטימלית לפעולה? כלומר, איך נחליט החלטות שיביאו לכמה שפחות טעויות.

ואפילו מקרה מעניין יותר: אנו יודעים את ההתפלגות - אך איננו יודעים את הפרמטרים של ההתפלגות. למשל, אנו יודעים כי הנתונים שלנו מתפלגים נורמלית אך לא יודעים את μ, σ^2. לכן: נרצה לבצע הערכה (estimate) של הפרמטרים.

איך לא, נתחיל בדוגמה. דילמה: אתה שוקל האם לקחת היום מטרייה, או שלא. אתה לא יודע בוודאות אם ירד או שלא ירד היום גשם! **זוהי דילמה.** מצד אחד, גשם+אין מטרייה: אתה תרטב ותהיה מעוצבן. מצד שני, אין גשם+ יש מטרייה: אתה תהיה מעוצבן קצת (כי סחבת מטרייה לחינם). אך, נניח וקיבלת שבריר של מידע. הסתכלת דרך החלון וראית עננים שחורים - כעת החלטת: בעקבות התצפית, אקח מטרייה.

כמובן שמטריות לא מעניינות אותנו. נרצה להכליל דוגמה זו ל**הסקה סטטיסטית**. בהינתן $data$, דגימה אמפירית, נרצה להכליל ממנה ולהסיק על "העולם". נפרמל זאת:

הגדרה 49. על מנת לתאר את תאוריית ההחלטה של בייס נגדיר:

1. **world states:** "מצבי עולם", מצבים $\{\omega_i\}_{i=1}^n$ שהם התרחישים האפשריים במציאות. מצבים אלו זרים, כלומר לכל $i \neq j \in [n]$ מתקיים $\omega_i \neq \omega_j$ וכן אם Ω הוא מרחב שלנו אזי $\Omega = \bigcup_{i=1}^n \omega_i$.
2. **תצפיות אמפיריות:** $S_n = \{x_1, ..., x_n\}$. דגימות אמפיריות אשר אספנו מהשטח. n מייצג את מס' הדגימות.
3. **מודל הסתברותי:** יש לנו ידע מלא על ההתפלגויות. בפרט, $P(\omega)$ - ההסתברות לפני שראינו בכלל את הנתונים (למשל: מהי ההסתברות שירד גשם ביום זה), וכן $P(S_n|\omega)$ - הסיכוי לראות את התצפיות הללו, באשר שהעולם נמצא במצב מסויים (למשל: מה הסיכוי לראות עננים שחורים אם אכן ירד גשם?).
4. **פעולות אפשריות:** קבוצה $A = \{\alpha_1, ..., \alpha_k\}$ של החלטות אפשריות שאנו יכולים לקבל. (קודם לכן בדוגמה, α_1 היא לקחת מטריה וα_2 היא שלא לקחת).
5. **עלויות:** נסמן $\Lambda = \{\lambda(\alpha_k|\omega_j)\}$ - המדד שקובע כמה נשלם על ההחלטה שלנו. כלומר, המחיר של נקיטת פעולה α_k כאשר המצב האמיתי בעולם הוא w_j. למשל: $\lambda(\overline{Umbrella}|rain)$ יהיה בעל מחיר גדול מאוד: העצבים שנרטבים ואין מטרייה.

משפט 50. יהי ω מצב כלשהו בעולם Ω שאינו ידוע. תהי x תצפית מהשטח. אזי, לפי חוק בייס מתקיים:

$$\underbrace{Pr(\omega|x)}_{Posterior} = \frac{Pr(x|\omega)}{Pr(x)} \cdot \underbrace{Pr(\omega)}_{Prior\ (belief)}$$

הסבר הנוסחה:

- $Pr(w)$: ה$Prior$, ה"אמונה", זהו הידע הקודם. זוהי ההסתברות של המצב ω עוד לפני שראינו את הנתונים.

- $Pr(x|\omega)$: זו ה$Likelihood$, ה"נראות". כלומר, בהינתן שאנו יודעים כי העולם במצב ω מה הסיכוי לקבל x.
- $Pr(x)$: מדובר בראייה. זוהי ההסתברות לראות את התצפית x ללא קשר למצב העולם כרגע.
- $Pr(\omega|x)$: מדובר ביעד הסופי, ה$Posterior$: ההסתברות בדיעבד. זוהי ההסתברות כי מצב העולם הוא ω לאחר שכבר ראינו את התצפית x. כלומר, אחרי שראית את התצפית (העננים), מה הסיכוי שבאמת יהיה גשם?

הערה 51. זהו המקום שבו אנחנו מגלים שנוסחה בהסתברות כל כך תמימה כל כך חשובה - אנו למעשה לוקחים ידע קודם, משלבים אותו עם נתונים חדשים ($Likelihood$) ומקבלים מסקנה מעודכנת.

מהי המטרה שלנו? מטרתנו לתכנן אסטרטגיה, שתסומן כפונקציה $\alpha(S_n)$. כקלט: היא מקבלת את התצפיות שלנו, וכפלט היא קובעת איזו פעולה α_i באשר $i \in [k]$ עלינו לבצע. הקריטריון להצלחה - היא תביא למינימום של עלויות/הפסדים. כלומר - נרצה כמה שפחות נזקים בממוצע בהתחשב במחירי הטעות λ.

הגדרה 52. הסיכון המותנה (Conditional risk) מסומן כ$R(\alpha_k|x)$ הוא העלות המצופה שלנו אם נבחר בפעולה α_k לאחר שראינו את התצפית x. מתקיים:

$$R(\alpha_k|x) = \mathbb{E}_{P(\omega|x)}[\lambda(\alpha_k|\omega)] = \sum_{\omega_j \in \Omega} \lambda(\alpha_k|\omega_j) Pr(\omega_j|x)$$

הסבר: מדובר סה"כ במעבר על כל מצבי העולם, חישוב מחיר הפעולה בהינתן המצב הספציפי הזה, ומכפילים בהסתברות של מצב העולם הזה בהינתן שאנחנו יודעים את התצפית x.

נבחין כי בנוסחה מוזכר $Pr(\omega_j|x)$. כיצד נחשבו? כלומר - בהינתן התצפית x, מהי ההסתברות למצב העולם ω_j? נוכל למצוא את ההסתברות באמצעות חוק בייס ונוסחת ההסתברות השלמה:

$$Pr(\omega_j|x) = \frac{Pr(x|\omega_j)}{Pr(x)} \cdot Pr(\omega_j) = \frac{Pr(x|\omega_j) \cdot Pr(\omega_j)}{\sum_{\omega_i \in \Sigma} Pr(x|\omega_i) \cdot Pr(\omega_i)}$$

כעת, יש לנו מס' סיכון לכל פעולה α_k אפשרית. כיצד נבחר את האסטרטגיה האופטימלית? ובכן, נגדיר:

$$\alpha^*(S_n) = argmin_\alpha R(\alpha|x)$$

כלומר, מעבר על כל הα האפשריות ונקח את הα שנתנה את התוצאה המינימלית עבור R (מספר הסיכון) - כיוון שנרצה כמה שפחות עלות.

הגדרה 53. (**Overall risk**). אסטרטגיה $\alpha(S_n)$ היא החוק שקובע איזו פעולה נבחרה עבור כל תצפית x_i אפשרית. נרצה להגדיר את העלות הממוצעת של חוק זה על פני כל התצפיות:

$$R(\alpha(S_n)) = \sum_{x_i \in S_n} R(\alpha(x_i)|x_i) \cdot Pr(x_i)$$

הערה 54. אסטרטגיה היא פונקציה, כך שבהינתן x_i היא אומרת מראש - בצע $\alpha(x_i)$. לכן בהגדרה הקודמת, רצינו לחשב את הסיכון של האסטרטגיה. לכל פעולה ישנה אסטרטגיה שנבחרה בפונקציית האסטרטגיה, $\alpha(x_i)$. נרצה לבדוק את הסיכון של אותה פעולה נבחרת, כלומר $R(\alpha(x_i)|x_i)$, ולשקלל זאת בהסתברות לקבל את אותו הקלט. כך נחשב למעשה את הסיכון המשוקלל של האסטרטגיה.

כעת נציג כמה וכמה דוגמאות חשובות:

דוגמה 55. (Zero One cost) נניח כי אנחנו רוצים לנחש נכון את מצבי העולם, וכל טעות עולה לנו אותו דבר. נניח כי הפעולות $\{\alpha_i\}_{i=1}^k$ הן הניחושים שלנו לגבי מצב העולם האמיתי $\{\omega_i\}_{i=1}^k$. (לדוגמה: מצבי העולם הם התמונה היא תפוז, והתמונה היא תפוח. α_1 זה ניחוש שהתמונה תפוז וα_2 ניחוש שהתמונה תפוח. ישנה התאמה על ניחוש מצב העולם האמיתי). כמו כן, נשלם 1 על כל טעות ו0 אם צדקנו. לשם כך, נשתמש בפונקציה דלתא δ_{kj} כך שמקיימת:

$$\delta_{kj} := \left\{ \begin{array}{ll} 1 & k = j \\ 0 & k \neq j \end{array} \right\}$$

לכן, נגדיר את העלות בצורה הבאה: $\lambda(\alpha_k|\omega_j) = 1 - \delta_{kj}$ (הרי אם הניחוש α_k מתאים למצב ω_j אזי $k = j$ ובמקרה זה $\delta_{kj} = 1$ ולכן $\lambda(\alpha_k|\omega_j) = 1 - 1 = 0$, אחרת נקבל כי $\lambda(\alpha_k|\omega_j) = 1$. הרי שאם צדקנו בניחוש, לא נרצה לשלם על כך אך אם טעינו נשלם בדיוק אחד).

כעת, נחשב את הסיכון המותנה R:

$$R(\alpha_k|x) = \sum_{j=1}^{K} \lambda(\alpha_k|\omega_j) Pr(\omega_j|x)$$

נבחין כי העלות היא 0 שצדקנו ו1 שטעינו, לכן הגורמים מתאפסים בכל המקרים בהם צדקנו. כלומר הΣ הנ"ל היא סכום המקרים בהם טעינו.

$$R(\alpha_k|x) = \sum_{j \neq k} Pr(\omega_j|x)$$

אך אנחנו יודעים כי סכום ההסתברויות הללו, של כל מצבי העולם הוא 1. כלומר, $\sum_{j\neq k} Pr(\omega_j|x) + Pr(\omega_k|x) = 1$ ולכן:

$$R(\alpha_k|x) = 1 - Pr(\omega_k|x)$$

כעת, נרצה $\alpha^*(S_n) = argmin\{R(\alpha_k|x)\}$. על מנת ש$1 - Pr(\omega_k|x)$ יהיה הכי קטן, נרצה כי $Pr(\omega_k|x)$ יהיה גדול ככל הניתן. כלומר, להמר על מצב העולם שיש לו את ההסתברות הגבוהה ביותר בהינתן התצפיות שראינו.

לסיכום: כשכל טעות עולה לנו אותו המחיר בדיוק, הדרך הכי זולה במקרה זה היא להמר על הסוס הכי חזק (זה עם ההסתברות הכי גבוהה). והערה נוספת: למעשה הדרך שלנו לשלם הכי פחות היא באמת אכן $1 - Pr(\omega_k|x)$. אנחנו תמיד נשלם 1, פרט למקרה בו צדקנו, ואז נשלם פחות. הדרך הכי טובה שלנו לשלם פחות ־ להמר על זה שנהיה צודקים לגביו בהסתברות הגבוהה ביותר.

■

דוגמה 56. כעת, נניח כי העלויות לא בהכרח 0 ו1. ננתח מקרה עם 2 מצבי עולם ו2 פעולות. נתבונן במטריצת עלויות מגודל 2×2. נסמן $\lambda_{kj} = \lambda(\alpha_k|\omega_j)$. הסיכון לכל פעולה מוגדר כך:

$$R(\alpha_1|x) = \lambda_{11} Pr(\omega_1|x) + \lambda_{12} Pr(\omega_2|x)$$

$$R(\alpha_2|x) = \lambda_{21} Pr(\omega_1|x) + \lambda_{22} Pr(\omega_2|x)$$

כעת, נשאל את עצמנו מתי נבחר בα_1? כאשר הסיכון שלה קטן יותר. דהיינו,

$$\lambda_{11} Pr(\omega_1|x) + \lambda_{12} Pr(\omega_2|x) < \lambda_{21} Pr(\omega_1|x) + \lambda_{22} Pr(\omega_2|x)$$

$$(\lambda_{12} - \lambda_{22}) Pr(\omega_2|x) < (\lambda_{21} - \lambda_{11}) Pr(\omega_1|x)$$

$$\frac{Pr(\omega_1|x)}{Pr(\omega_2|x)} > \frac{\lambda_{12} - \lambda_{22}}{\lambda_{21} - \lambda_{11}}$$

כעת, נעזר בחוק בייס:

$$\frac{\frac{Pr(x|\omega_1)}{Pr(x)} \cdot Pr(\omega_1)}{\frac{Pr(x|\omega_2)}{Pr(x)} \cdot Pr(\omega_2)} > \frac{\lambda_{12} - \lambda_{22}}{\lambda_{21} - \lambda_{11}}$$

$$\underbrace{\frac{Pr(x|\omega_1)}{Pr(x|\omega_2)}}_{\text{Likelihood ratio}} > \underbrace{\frac{\lambda_{12} - \lambda_{22}}{\lambda_{21} - \lambda_{11}} \cdot \frac{Pr(\omega_2)}{Pr(\omega_1)}}_{\text{Decision boundary}}$$

מאגף שמאל, קיבלנו את הLikelihood ration: יחס בין הסתברויות, שעונה על השאלה: פי כמה התצפית שראינו x מתאימה למצב 1 יותר מאשר למצב 2?

מצד ימין, קיבלנו את הDecision boundry: מדובר בערך מספרי שקובע את הרף ומורכב מ2 חלקים:

1. יחס ה$Priors$: כמה מצב 2 נפוץ יותר לעומת מצב 1 ביומיום.
2. יחס העלויות: כמה יותר כואב לטעות בסוג אחד של טעות לעומת השני.

המספר הזה הוא הגבול - אם הראיות מהשטח (לייקהוד רטיו) חזקות יותר מערך זה - נבחר בα_1. אחרת: נבחר בα_2.

במקום לחשב הסתברויות מסובכות בכל שלב, המודל הנ"ל אומר: תשווה בין שתי האפשרויות, תראה אם הראיה מספיק חזקה לך על מנת לעבור את הסף.

■

עד כה התעסקנו במצב בו יש לנו תצפית אחת, x. ומה אם יש לנו n תצפיות? כלומר תצפיות $\{x_i\}_{i=1}^n$? במצב זה:

$$\underbrace{\frac{Pr(x_1,....,x_n|\omega_1)}{Pr(x_1,....,x_n|\omega_2)}}_{\text{Likelihood ratio}} > \underbrace{\frac{\lambda_{12}-\lambda_{22}}{\lambda_{21}-\lambda_{11}}\cdot\frac{Pr(\omega_2)}{Pr(\omega_1)}}_{\text{Decision boundary}} = \Theta$$

כעת, נניח כי התצפיות שלנו $\{x_i\}_{i=1}^n$ הן בלתי תלויות זו בזו (דרישה הגיונית), בהינתן למצב העולם ω_i. אזי, לפי חוקי הסתברות יתקיים כעת:

$$Pr(x_1,...,x_n|\omega_1)=\prod_{i=1}^n Pr(x_i|\omega_1), Pr(x_1,...,x_n|\omega_2)=\prod_{i=1}^n Pr(x_i|\omega_2)$$

ולכן אי השוויון יהפוך להיות: $\frac{\prod_{i=1}^n Pr(x_i|\omega_1)}{\prod_{i=1}^n Pr(x_i|\omega_2)} > \Theta$. נבחין כי בעולם של הסתברויות, להכפיל הרבה הסתברויות זה לא רעיון חכם במיוחד. למה לא? כי יתכן וההסתברויות הנ"ל מאוד קטנות וקרובות לאפס, מכפלתן זו בזו תהיה עוד יותר קטנה וקרובה לאפס, ומבחינת המחשב זה ייצור מספרים אפסיים שהוא לא ידע כיצד להתמודד איתם. לכן, נרצה להעזר במתמטיקה על מנת למנוע מצב כזה: נשתמש בlog שהיא פונקציה מונוטונית עולה ולכן לא תשפיע על אי השוויון. נגדיר $\Theta' = log(\Theta)$ ויתקיים:

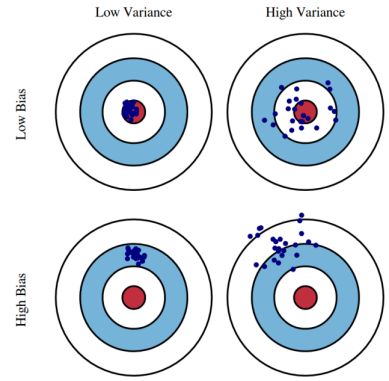
$$log(\frac{\prod_{i=1}^n Pr(x_i|\omega_1)}{\prod_{i=1}^n Pr(x_i|\omega_2)}) > \Theta' \iff \sum_{i=1}^n log(\frac{Pr(x_i|\omega_1)}{Pr(x_i|\omega_2)}) > \Theta'$$

מעבר למשמעות המתמטית, במקום מבחן אחד עכשיו יש לנו צבירת ראיות. לכל תצפית x_i כך ש$i \in [n]$ נחשב את הLog-Likelihood ratio שלה. אם התצפית תומכת בω_1 אזי הlog יהיה חיובי ואחרת שלילי (לפי חוקי log, הרי תמיכה משמעותה $\frac{Pr(x_i|\omega_1)}{Pr(x_i|\omega_2)} > 1$ ובמקרה זה הlog חיובי ואי תמיכה משמעותה $\frac{Pr(x_i|\omega_1)}{Pr(x_i|\omega_2)} < 1$ ובמקרה זה הlog שלילי). אנו סוכמים את הראיות, לכן נבחר את α_1 אם סכום התצפיות הכולל היה גדול מהגבול. בדומה, על α_2.

הערה 57. כפי שראינו בדוגמאות קודמות, הנחנו כי אנחנו יודעים את התפלגות הנתונים. תאוריית בייס מניחה שיש לנו ידע אלוהי על התפלגויות הנתונים. בפועל, בחיים האמיתיים אף אחד לא נותן לנו את התפלגות הנתונים. אז מה נעשה בקורס? הקורס ילמד אותנו כיצד לגשר בין הגישות.

2.3 The Bias-Variance Trade-off

Bias היא הטייה, זו שגיאה שנובעת מהנחות יסוד פשטניות מדי של המודל. שונות זו השגיאה שנובעת מרגישות יתר של המודל לשינויים בנתוני האימון. ננסה להבין מה הקשר בין השניים, באמצעות התמונה הבאה:



נחלק למקרים:

1. **שונות והטיה נמוכים** (**שמאל למעלה**): המודל פוגע בדיוק במרכז, ובכל פעם שניתן לו סט חדש של נתונים יפגע בדיוק באותו מקום - עקבי, זהו היעד שלנו.
2. **שונות נמוכה, הטיה גבוהה** (**שמאל למטה**): המודל מאוד עקבי (תמיד פוגע באותה נקודה!) כי השונות נמוכה, אבל הנקודה הנ"ל רחוקה מהמרכז: כי יש הטיה. המודל טועה בגלל הנחה שגויה, אך טועה בצורה עקבית.
3. **שונות גבוהה, הטיה נמוכה** (**ימין למעלה**): בממוצע, הנקודות נמצאות יחסית קרובות למרכז כי ההטייה נמוכה אך הן מפוזרות זו מזו בגלל שהשונות גבוהה. המודל מבין את הכיוון הכללי - אך נורא מפוזר.
4. **שונות והטיה גבוהים** (**ימין למטה**): המצב הכי גרוע. גם הטיה גבוהה - ולכן רחוק מהיעד, וגם הנתונים מפוזרים כי השונות גבוהה. המודל טועה גם בממוצע וגם לא עקבי.

נרצה לפרמל את המושגים עליהם דנו כאן, ולפתח קשר מתמטי.

הגדרה 58. יהיו $S_n = \{(x_i, y_i)_{i=1}^n\}$ נתוני האימון שלנו, המגיעים מעולם $\mathcal{D}$ כך שהדגימות ב"ת זו בזו. נרצה להעריך את y (שלא ידוע) באמצעות פונקציית היפותזה $\mathcal{H}$: $h(x) = h(x; S_n)$ שנבחין שתלויה בנתונים עליה התאמנה. מטרתנו לחזות את y עבור נק' מבחן חדשה $(x, y) \sim \mathcal{D}$ שלא נמצאת בנתוני האימון שלנו. נגדיר את הMSE (Mean Squared Error) כתוחלת של ריבוע ההפרש בין הערך האמיתי לחזוי:

$$MSE = \mathbb{E}_{(x,y)\sim\mathcal{D}}[(y - h(x))^2]$$

מטרתנו תהיה להבין מדוע המודל שלנו טועה - האם זה בגלל חוסר גמישות (Bias) או מרגישות יתר לנתונים (Variance). הMSE מתפרק לשניהם -

- $Bias$: המידה שבה החיזוי של המודל מתאים לפונקציית המטרה בממוצע.
- $Variance$: המידה שבה החיזור עבור נקודה בודדת משתנה בממוצע.

לפני כן, נגדיר שני סוגי תוחלות:

1. תוחלת על נתונים: תוחלת ע"פ כל הנקודות מהעולם $\mathcal{D}$. כאשר נבדוק טעות מבחן, נשתמש בתוחלת זו.

$$\mathbb{E}_{(x,y)}[\cdot] = \mathbb{E}_{(x,y)\sim\mathcal{D}}[\cdot]$$

2. תוחלת על סטי אימון: תוחלת מעל סטי האימון. התוחלת הנ"ל מדגימה בממוצע כיצד המודל מתנהג בממוצע כשמחליפים לו את הדטה עליו התאמן.

$$\mathbb{E}_{S_n}[\cdot] = \mathbb{E}_{S_n\sim\mathcal{D}^n}[\cdot]$$

משפט 59. בהינתן התפלגות נתונים $\mathcal{D}$, מדגם אימון $S_n \sim \mathcal{D}^n$, והיפותזה $h(x, S_n)$ השגיאה הריבועית הממוצעת (MSE) ניתנת לפירוק לשני רכיבים חיוביים:

$$MSE = Bias^2 + Variance$$

באשר:
$Bias = \mathbb{E}_{(x,y)}[(y - \mathbb{E}_{S_n}[h(x)])^2]$ וכן $Variance = \mathbb{E}_{(x,y)}[\mathbb{E}_{S_n}[(h(x) - \mathbb{E}_{S_n}[h(x)])^2]]$

הוכחה:

$$\mathbb{E}_{(x,y)\sim\mathcal{D}}[(y - h(x))^2] = \mathbb{E}_{(x,y)}[(y - \mathbb{E}_{S_n}[h(x)] + \mathbb{E}_{S_n}[h(x)] - h(x))^2] =$$

$$\mathbb{E}_{(x,y)}[(y - \mathbb{E}_{S_n}[h(x)])^2] + \mathbb{E}_{(x,y)}[(\mathbb{E}_{S_n}[h(x)] - h(x))^2] + 2\mathbb{E}_{(x,y)}[(\mathbb{E}_{S_n}[h(x)] - h(x)) \cdot (y - \mathbb{E}_{S_n}[h(x)])]$$

נרצה לנתח את הביטוי $2\mathbb{E}_{(x,y)}[(\mathbb{E}_{S_n}[h(x)] - h(x)) \cdot (y - \mathbb{E}_{S_n}[h(x)])]$:

$$2\mathbb{E}_{(x,y)}[(\mathbb{E}_{S_n}[h(x)] - h(x)) \cdot (y - \mathbb{E}_{S_n}[h(x)])] =$$

$$2\mathbb{E}_{S_n,(x,y)}[(y - \mathbb{E}_{S_n}[h(x)])(\mathbb{E}_{S_n}[h(x)] - h(x))] = 2\mathbb{E}_{(x,y)}[\mathbb{E}_{S_n}[(y - \mathbb{E}_{S_n}[h(x)])(\mathbb{E}_{S_n}[h(x)] - h(x))]]$$

$$= 2\mathbb{E}_{(x,y)}[(y - \mathbb{E}_{S_n}[h(x)])[(\mathbb{E}_{S_n}[\mathbb{E}_{S_n}[h(x)] - h(x))]] =$$

$$2\mathbb{E}_{(x,y)}[(y - \mathbb{E}_{S_n}[h(x)])[\mathbb{E}_{S_n}[h(x)] - \mathbb{E}_{S_n}[h(x))]] = 0$$

מכאן קיבלנו:

$$\mathbb{MSE} = \mathbb{E}_{(x,y)}[(y - \mathbb{E}_{S_n}[h(x)])^2] + \mathbb{E}_{(x,y)}[(\mathbb{E}_{S_n}[h(x)] - h(x))^2]$$

■

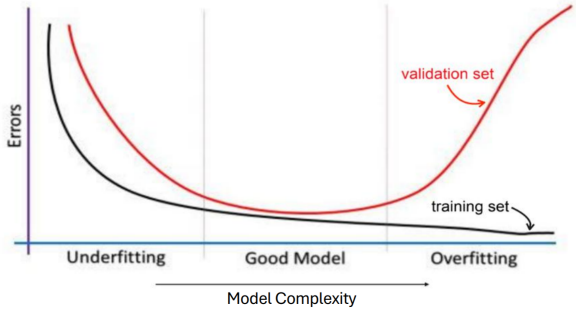
הערה 60. למעשה, $Bias$ הוא כמה המודל $h(x)$ חוזה טוב את המטרה בתוחלת. אם יש $Bias$ גבוה - המודל עקום מהיסוד.
כיצד נקטין את ה$Bias$? צריך להפוך את המודל לגמיש יותר או לחלופין למורכב יותר. כלומר: נרצה ש$h(x)$ תבחר מסט רחב יותר של פונקציות - למשל פולינומי מדרגות גבוהות ולא רק לינארי.
כיצד נקטין את השונות? אפשרות אחת היא להגדיל את הדטה של האימון (ככה נעריך טוב יותר את $h(x)$), **אפשרות אחרת** היא להגביל מראש את המודל לקבוצה קטנה יותר של פונקציות - הרי אם נכריח אותו לבחור מתוך סט קטן (למשל מתוך קווים לינאריים בלבד) הוא יהיה הרבה יותר "קשיח" : ההגיון הוא כי מודל פשוט הוא מודל עקבי - קו ישר לא יכול להתפתל כדי לעבור דרך רעש בנתונים ולכן אם נחליף את נתוני האימון הקו הישר שנקבל יהיה דומה לקודם. התוצאה היא שהשונות תרד משמעותית אך ההטייה עלולה לעלות.

2.4 $Underfitting$ ו$Overfitting$

כיצד בפועל נראה אימון של מודל? נחלק לשלבים:

Training loop (**לולאת האימון**): זהו השלב שבו המודל לומד. המודל מעדכן את הפרמטרים שלו, כדי למזער את השגיאה על סט האימון S_n. המטרה בשלב זה היא להקטין את ה$Bias$. ככל שנרוץ יותר פעמים על נתוני האימון, המודל יתאים את עצמו אליהם יותר.
Evaluate the loss on the validation set: זהו שלב ה"בקרה" שקורה תוך כדי האימון. מדוע זה נחוץ? סט האימון לא יכול להגיד לנו אם התחלנו לעשות overfitting, ולכן אנחנו בודקים את הביצועים על נתונים שהמודל לא התאמן עליהם ישירות. אם השגיאה באימון יורדת אבל השגיאה בוולדיאציה מתחילה לעלות: זהו סימן מובהק לכך שהמודל התחיל ללמוד רעש ועלינו לעצור.
Evaluate on Test Set: זהו שלב המבחן הסופי שקורה רק פעם אחת בסוף התהליך. סט המבחן חייב להיות טהור - אסור לנו לקבל החלטות על המודל כמו לשנות פרמטרים בהתבסס על סט המבחן - זה לרמות את עצמנו.

כיוון שאנחנו כל הזמן עושים בקרה - שגיאת האימון שלי תמיד ירדה, כי אני מתקן את עצמי תוך כדי הבקרה. אך השגיאה של ה$validation$ לא בהכרח תמיד תרד! כמו בדוגמה בתמונה מטה. אם נראה מצב שהשגיאה של ה$validation$ עולה: אנחנו נבין שיש כאן סכנה לoverfitting. אם ראינו מצב כזה - צריך לעצור ולתת למודל שוב להתאמן. לבסוף, כשנראה ששגיאת הואלידציה מתיישרת עם שגיאת האימון (שיורדת), נוכל לעבור לשלב המבחן הסופי.

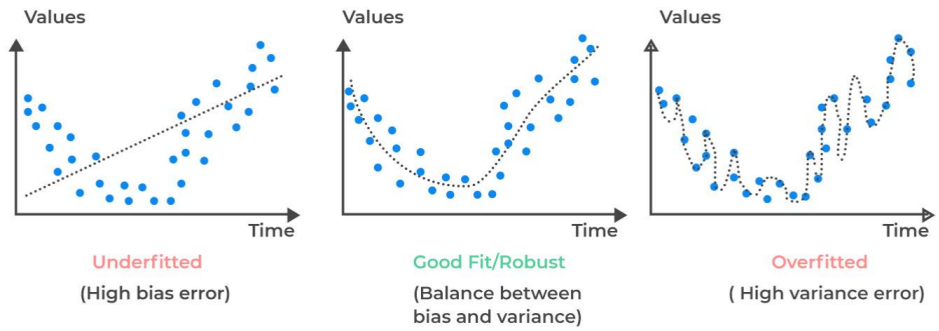


הגדרה 61. Underfitting הוא מצב של תת התאמה. זהו מצב שבו המודל פשוט מדי בכדי לתפוס את המבנה של הנתונים. סימנים לכך:
1. שגיאה גבוהה בסט האימון וגם בסט ההערכה.
2. מצב של $Bias$ גבוה, המודל לא מצליח ללמוד אפילו את הנתונים שהוא ראה: לכן ברור שלא יצליח לחזות חדשים.
למשל, נסיון להתאים קו לינארי לנתונים שמתנהגים בצורה ריבועית של פרבולה.
מדוע זה קורה? המודל פשוט מדי. **איך לתקן?** להגדיל את מורכבות המודל או לאפטמז יותר טוב.

הגדרה 62. Good fit היא התאמה טובה, זהו המצב אליו אנחנו שואפים בתהליך האימון. סימנים לכך:
1. שגיאה נמוכה בסט האימון וגם בסט ההערכה.
2. מצאנו איזון בין השונות להטיה: ה$Bias$ נמוך מספיק בשביל להבין את החוקיות, והשונות נמוכה מספיק בשביל לא להיות מושפעים מהרעש.
המודל מבצע הכללה (Generalization) טובה.

הגדרה 63. overfitting הוא מצב שבו המודל שינן את נתוני האימון במקום ללמוד את החוקיות שלהם. סימנים לכך:
1. שגיאה נמוכה מאוד בסט האימון, אך שגיאה גבוהה בסט ההערכה.
2. מצב של שונות גבוהה מאוד - המודל גמיש מדי וחודר לרעש של מדגם ספציפי. פוגע בנתוני האימון אך מתרסק על נתונים חדשים.
מדוע זה קורה? סט נתונים קטן מדי או מודל מורכב. **כיצד לתקן?** לקחת מודל פחות מורכב, לקחת דטה־סט גדול יותר.

הערה 64. בתמונה, דוגמאות לכל אחד מהמקרים:



אם נרצה לתאר מהם שונות והטיה, נוכל לומר כך: Bias מתאר למעשה כמה הנתונים שלי רחוקים מהמצב האמיתי (y), בעוד Variance מתאר כמה בכל שלב ושלב החיזויים שלי יהיו שונים זה מזה. כעת נרצה לשאול:
מהו Bias גבוה? בממוצע, אנחנו רחוקים מהחיזוי האמיתי. המודל לא הבין את מורכבות הדטה.
מהו Variance גבוה? המודל הופך להיות רגיש מדי לכל תנודה בנתוני האימון - במקום ללמוד חוקיות כללית שמשותפת לכל הנתונים בעולם הוא מתחיל להתייחס לכל רעש או סטיות מקריות בסט האימון הספציפי. מדוע לפונקציה מורכבת יש שונות גבוהה? המודל בונה פונקציה מורכבת - שעוברת בדיוק דרך כל נקודה: אם יש נקודה אחת שהיא טעות או רעש - המודל יתעקם כדי להגיע אליה. ההתעקמויות האלו הם יוצרים וריאנס גבוה: המודל איבד את יכולת ההכללה כי בנה את עצמו בהתאם לנתונים.

2.5 Cross Validation

כפי שראינו, אי אפשר להשתמש בTest set על מנת לקבל החלטות על המודל לכן צריך את סט הValidation. תפקידו: עזרה בבחירת הגדרות המודל (אלו הHyper-parameters): מס' הפרמטרים, בחירת אלגוריתם אופטימיזציה, קבועי רגולציה λ וכו'.
מה הבעיה שנוצרת מכך? אם הdataset שלנו קטן,
אם נפריש חלק משמעותי ממנו לValidation ישאר קצת מאוד דטה עבור האימון עצמו. (זה יעלה את הBias כי המודל לא למד מספיק)
אם נפריש חלק קטן ממנו לValidation הערכת הביצועים שלנו תהיה תלויה מאוד בנקודות הספציפיות שנפלו בסט הוולידציה. בחירה מקרית מזלזלת או מחמירה מדי של סט וולידציה תשפיע על כל החלטות המדגם.

לשם כך, נשתמש בשיטה בשם Cross Validation:
1. נקח את כל הנתונים שלנו ונחלקם לk קבוצות שוות בגודלן. (לרוב משתמשים $k = 5, 10$)
2. מריצים אימון למשך k פעמים: בכל איטרציה $1 \leq i \leq k$:
א. Training set: משתמשים ב$k-1$ מהחלקים כדי לאמן את המודל.
ב. Validation set: משתמשים בחלק הi שנותר בצד כדי לבדוק את הביצועים.
המשמעות: בכל פעם חלק אחר של הנתונים משמש כ"בוחן".
3. בסוף הלולאה יש לנו k ערכי שגיאה (אחד לכל קבוצה) ונחשב את הממוצע שלהם: $Error = \frac{1}{k}\sum_{i=1}^{k} Error_i$. הממוצע הנ"ל הוא הPerformance Estimation שלנו, הרבה יותר יציב מהערכה על סט וולידציה בודד כי הוא ראה את כל הנתונים בתפקיד הוולידציה.
4. בחר את הפרמטרים עם הperformance הגבוה ביותר. אנחנו למעשה חוזרים על התהליך הנ"ל כולו עבור קבוצות שונות של היפר פרמטרים (פולינומים דרגה 2, דרגה 3 וכו'...). נבחר את הפרמטרים שהשיגו את ממוצע השגיאות הנמוך ביותר.

הערה 65. אם בחרנו בשיטה זו, אנחנו קודם נבצע cross validation, ואז נעבור לשלב האימון - ללא ולדיציה, עם הפרמטרים שנבחרו, ולבסוף את שלב המבחן. לעומת זאת: אם לא בחרנו בשיטה זו - הוולדיציה קורית במקביל לאימון, ולבסוף מתבצע המבחן.

הערה 66. כיצד בחירת k משפיעה על ההטייה והשונות?

נבחין כי ככל שk יהיה גדול יותר, ה$Bias$ יהיה קטן יותר: כיוון שככל שk גדול, כך סט האימון בכל קבוצה גדול יותר וקרוב יותר לגודל N המקורי, מודל שמאומן על יותר נתונים מייצג טוב יותר את המודל הסופי, ולכן השגיאה שמודדים בCV היא הערכה קרובה יותר.

כמו כן, ככל שK גדול יותר ־ ה$Variance$ יהיה גדול יותר: כיוון שככל שk גדל, סטי האימון בכל איטרציה הופכים להיות כמעט זהים (**לצורך ההמחשה:** אם $N = 100$ וכן $k = 2$, אזי החפיפה תהיה אפסית שכן אפשר לחלק ל2 קבוצות בגודל 50. אך אם $N = 100$ ו$k = 10$, במקרה זה בכל איטרציה יהיו 90 דגימות ־ ויהיה הרבה יותר חפיפה בין הדגימות במקרה זה), כאשר המדגמים עליהם נריץ את K האיטרציות מאוד דומים ־ הממוצע שייחושב בסוף יהיה רגיש יותר לנתונים חריגים ולכן השונות תגדל.

מסקנה: שוב ישנו טרייד־אוף בין השניים.

3 הרצאה 3

3.1 Parameter estimation (**הערכת פרמטרים**)

תחת ההנחה של Bayes Decision Theory, התייחסנו למצב בו כל הנתונים הסטטיסטיים ידועים לנו. כלומר, $\forall i\ Pr(\omega_i)$ היה ידוע לנו ה$Priors$, וגם הצפיפות המותנית $Pr(x|\omega_i)$. אם הנתונים הללו אכן בידינו ־ אנו יכולים לבנות מסווג אופטימלי (Bayes Optimal classifier) שממזער את טעות הסיווג בצורה מתמטית. אך ־ בעולם האמיתי זה לא ככה. אנחנו לא באמת יודעים את ההתפלגויות האמיתיות. יש לנו אינטואיציה כיצד הדברים אמורים להראות ־ ויש לנו סט אימון. במקום להניח שאנחנו יודעים את הסטטיסטיקה של העולם, אנחנו נשתמש בדגימות שבידינו על מנת להעריך את אותן הסתברויות.

למה שנעבור לגישה זו? למה שלא פשוט נעבור פונקציה פונקציה ונחשב אותן ונבדוק אם היא מתאימה? ניסיון להעריך פונקציה כזו קשה מהסיבות הבאות:

- צריך כמות עצומה של דגימות שלרוב אין לנו
- קללת המימד: ככל שלx יש יותר מאפיינים (מימד גבוה יותר) המרחב הופך לריק יותר, ונצטרך כמות דגימות שגדלה אקספוננציאלית.

לכן ־ **אנחנו מעט נרמה. נניח מראש כי אנחנו יודעים את הצורה של הפונקציה: אך אנחנו לא יודעים את ערך הפרמטרים שלה.** למשל, אם נניח כי $Pr(x|\omega_i) \sim N(\mu_i, \Sigma_i)$ נצטרך רק ללמוד מהדטה את Σ_i, μ_i.

3.2 Maximum Likelihood (ML)

- התפיסה: הפרמטר Θ קבוע אך לא ידוע. אין התפלגות ־ מדובר במס' שצריך למצוא.
- התוצר: הערכה נקודתית $\hat{\Theta}_{ML}$ של הפרמטר.
- הפרשנות: נבחר את Θ שיתן את הסיכוי הגבוה ביותר לראות את הדטה שבאמת ראינו. כלומר ־ אם זה מה שקרה בפועל: איזה Θ הכי סביר שהוביל לזה לקרות?

יהיו הנתונים $\mathcal{D} = S_n = \{x_1, ..., x_n\}$. מייצגים את קבוצת הdataset. כל x_i היא דגימה בודדת. אנו מניחים כי הנתונים נוצרו ע"י תהליך סטטיסטי שיש לו פרמטר לא ידוע θ. אנחנו נבחר את הθ שגורם לדטה שראינו להיות הכי סביר (Most Likely). כלומר ־ אם נסתכל על כל הθ האפשריים בעולם, נשאל איזה מהם נותן את ההסתברות הגבוהה ביותר להיות $x_1, ...x_n$?

נגדיר:

$$\hat{\theta}_{ML} = argmax_\theta\ Pr(S_n|\theta)$$

כאומד לθ לפי ML, כאשר $Pr(S_n|\theta)$ היא פונקציית הנראות. היא מודדת כמה הנתונים סבירים ביחס לפרמטר מסויים θ. כיוון שהדגימות iid (מגיעות מהתפלגות זהה וב"ת) הרי שמתקיים:

$$Pr(S_n|\theta) = \prod_{i=1}^{n} Pr(x_i|\theta)$$

כיוון שlog פונקצייה מוטונטית עולה יתקיים:

$$\hat{\theta}_{ML} = argmax_\theta\ Pr(S_n|\theta) = argmax_\theta\ \sum_{i=1}^{n} Pr(x_i|\theta)$$

דוגמה 67. מודל התפלגות בינומית. נניח כי אנחנו מטילים מטבע כך ש: $Pr(H) = p, Pr(T) = 1 - p$. לפי נוסחה, ההסתברות לk הצלחות בהינתן n נסיונות ניתנת לפי:

$$Pr(X = k|p) = \binom{n}{k} p^k (1-p)^{n-k}$$

נתייחס ל(n,k) כקבועים ונאמוד את ערך הפרמטר p כך שימקסם את ההסתברות. נפעיל log:

$$ln(\ell(p)) = ln(\binom{n}{k}) + kln(p) + (n-k)ln(1-p)$$

כעת נגזור ונשווה לאפס:

$$\nabla_p = \frac{k}{p} - \frac{n-k}{1-p} = 0 \Longrightarrow \frac{k}{p} = \frac{n-k}{1-p} \Longrightarrow k - kp = pn - kp \Longrightarrow \hat{p}_{ML} = \frac{k}{n}$$

כלומר ערך הפרמטר p שימקסם את ההסתברות יהיה השכיחות היחסית של k בהינתן הדטה n.

■

3.3 Maximum A Posteriori (MAP) (הסתברות פוסטריורית מקסימלית)

- התפיסה: הפרמטר Θ הוא משתנה אקראי, יש לנו ידע מוקדם עליו $(Prior)$ לפני שראינו את הנתונים.
  - התוצר: הערכה נקודתית $\hat{\Theta}_{ML}$ של הפרמטר.
  - הפרשנות: נבחר את Θ הכי סביר אחרי $(Post)$ ששקללנו את הדטה עם הידע המוקדם שלנו. פשרה בין מה שחשבנו מראש ־ למה שהנתונים מראים.

הMap למעשה הוא השכיח של התפלגות ה$Posterior$. נגדיר:

$$\hat{\theta}_{MAP} = argmax_\theta \ Pr(\theta|\mathcal{D}) \propto argmax_\theta \ Pr(\mathcal{D}|\theta)Pr(\theta)$$

נזכר כי $Pr(\theta|\mathcal{D}) = \frac{Pr(\mathcal{D}|\theta)Pr(\theta)}{Pr(\mathcal{D})}$, כיוון ש$Pr(\mathcal{D})$ הוא ערך קבוע שאיננו תלוי בθ, אנחנו יכולים לומר שפורפורצינלית $(\propto)$ אם נרצה למקסם את $Pr(\theta|\mathcal{D})$ זה שקול למקסם אך ורק את המונה. כלומר ־ הMAP הוא הנקודה שבה הדטה $\times$ הידע המוקדם ־ הכי גבוה.

- הערה: אם ה$Prior$ שלנו היא פונקציה אחידה (כלומר $Pr(\theta) = c$ עבור c כלשהו) אזי $MAP = MLE$, שכן במקרה זה נוכל גם לומר כי:

$$\hat{\theta}_{MAP} = argmax_\theta \ Pr(\theta|\mathcal{D}) \propto argmax_\theta \ Pr(\mathcal{D}|\theta)Pr(\theta) \propto argmax_\theta \ Pr(D|\theta) = \hat{\theta}_{ML}$$

לכן אם נסכם, הMAP בניגוד לMLE גם מייחס חשיבות לערך ה$Prior$, וכן הוא פחות רגיש לרעשים והוא מתקן את הMLE לכיוון ה$prior$.

דוגמה 68. נדון שוב בהטלת מטבע. נסמן θ כהסתברות לH. $\mathcal{D}$ הוא הדטה שראינו בפועל: k פעמים H וכן $n-k$ פעמים T. נניח כי הידע המוקדם שלנו על המטבע מתנהג לפי התפלגות בטא, כאשר:

$$Pr(\theta) \propto \theta^{\alpha-1}(1-\theta)^{\beta-1}$$

(מדוע פורפורצינולי? ישנם קבועים נוספים בהתפלגות בטה, אך הם לא רלוונטים כשנרצה לחשב את המקסימום).
אם כן, **אנו יודעים שני ערכים**:

- נראות ־ $Likelihood$: מהדטה נתון כי ראינו k ראשים ו$n-k$ פלי, לכן כיוון שיש אי תלות :

$$Pr(D|\theta) = \theta^k(1-\theta)^{n-k}$$

- ידע מוקדם ־ $Prior$: לפי הגדרת פונקציית בטה:

$$Pr(\theta) \propto \theta^{\alpha-1}(1-\theta)^{\beta-1}$$

מכאן:

$$\hat{\theta}_{MAP} \propto argmax_\theta \; Pr(\mathcal{D}|\theta)Pr(\theta) \propto \theta^k(1-\theta)^{n-k} \cdot \theta^{\alpha-1}(1-\theta)^{\beta-1} = \theta^{\alpha-1+k}(1-\theta)^{n-k+\beta-1}$$

נפעיל ln:

$$(\alpha-1+k)ln(\theta) + (n-k+\beta-1)ln(1-\theta) \approx xln(\theta) + yln(1-\theta)$$

נגזור:

$$\frac{x}{\theta} - \frac{y}{1-\theta} = 0 \Longrightarrow \theta y = x - x\theta \Longrightarrow \theta = \frac{x}{x+y} = \frac{\alpha+k-1}{n+\beta+\alpha-2}$$

וסה"כ קיבלנו כי $\hat{\theta}_{MAP} = \frac{\alpha+k-1}{n+\beta+\alpha-2}$

■

דוגמה 69. נניח כי $\{x_i\}_{i=1}^n$ הם זמני המתנה לתור. נניח $i.i.d$. אם כן, ידוע כי: $\forall i \in [n] : Pr(x_i|\theta) = \theta e^{-\theta x_i}$. לפי הגדרת הנראות,

$$Pr(\mathcal{D}|\theta) = \prod_{i=1}^n \theta e^{-\theta x_i} = \theta^n e^{-\theta \sum_{i=1}^n x_i}$$

נניח כי הידע המוקדם, ה$prior$ הוא $Pr(\theta) = e^{-\theta}$. מכאן, לפי הנוסחה:

$$Pr(\theta|\mathcal{D}) \propto \theta^n e^{-\theta \sum_{i=1}^n x_i} \cdot e^{-\theta} = \theta^n e^{-\theta(\sum_{i=1}^n x_i + 1)}$$

מכאן, נפעיל ln ונגזור לפי θ. נקבל כי: $\hat{\theta}_{MAP} = \frac{n}{\sum_{i=1}^n x_i + 1}$.
לצורך ההשוואה, אם היינו פועלים לפי MLE היינו מקבלים $\hat{\theta}_{ML} = \frac{n}{\sum_{i=1}^n x_i}$. מה ההבדל? ה$+1$, הוא הנציג של ה$prior$, אם ראינו מעט מאוד דטה, כלומר n קטן יחסית, ה1 הזה ישפיע מאוד לטובת ה$prior$. ככל שנוסיף עוד דטה - ה$+1$ יהפוך לזניח ויתקיים $\hat{\theta}_{ML} \approx \hat{\theta}_{MAP}$.

■

3.4 Bayesian Estimation (הערכה בייאסיאנית)

- התפיסה: הפרמטר Θ הוא משתנה אקראי.
- התוצר: התפלגות פוסטריורית מלאה $Pr(\theta|D)$, לא מקבלים ערך אחד אלא פונקצייה שאומרת מה ההסתברות לכל ערך אפשרי של θ.
- הפרשנות: לא מהמרים על ערך אחד של θ. מחשבים ומעדכנים את כל האי וודאות שלנו לגבי הפרמטר. אנחנו לא מבצעים כאן אופטימיזציה של הפרמטר - אלא אינטגרציה / חישוב של התפלגות מלאה.

כעת נתייחס לθ כמשתנה אקראי, ולא כערך קבוע. הרעיון כאן הוא כתהליך של עדכון. אנו מביאים איתנו ידע קודם. $Pr(\theta)$ זו ההתפלגות המוקדמת - היא מייצגת את מה שאנחנו חושבים על הפרמטר לפני שראינו ולו דגימה אחת מהדטה. (למשל: אני חושב שהמטבע הוגן אזי $pr(H) = \frac{1}{2}$). בעת שמגיע דטה $\mathcal{D}$, נשתמש בחוק בייס לעדכן את האמונה שלנו - המטרה למצוא את ה posterior (ההסתברות בדיעבד). באופן הבא:

$$Pr(\theta|\mathcal{D}) = \frac{Pr(\mathcal{D}|\theta)Pr(\theta)}{Pr(\mathcal{D})}$$

כאן, בניגוד לMLE שם חיפשנו רק נקודת אחת, אנחנו מחשבים את כל הפונקציה $p(\theta|\mathcal{D})$.

דוגמה 70. (לא מההרצאה). בידינו מטבע. איננו יודעים את ערך הפרמטר. נניח כי ה$prior$ שלנו הוא $Pr(\theta) = 6\theta(1-\theta)$. נניח כי ראינו דטה $\mathcal{D}$ של הטלה אחת H. אם כן, $Pr(\mathcal{D}|\theta) = \theta$ שכן θ מעריך את ההסתברות לקבל H.
אם כן, כעת נחשב:

$$Pr(\mathcal{D}|\theta)Pr(\theta) = \theta \cdot 6\theta(1-\theta) = 6\theta^2(1-\theta)$$

זו פונקציה חדשה שמתאימה לדטה שראינו. נבחין כי על מנת שיהיה מדובר בפונקציית הסתברות - השטחה מתחתה חייב להיות שווה לאחד. אך,

$$\int_0^1 6\theta^2(1-\theta) = \int_0^1 6\theta^2 - 6\theta^3 = [{}_0^1 2\theta^3 - 1.5\theta^4] = 0.5$$

זו לא פונקציית הסתברות תקינה. אך, לפי בייס: צריך לחלק ב$Pr(\mathcal{D})$. והרי, $Pr(\mathcal{D}) = \int P(\mathcal{D}|\theta)Pr(\theta)d\theta = 0.5$ לפי נוסחת ההסתברות השלמה. מכאן, נקבל כי:

$$Pr(\theta|\mathcal{D}) = \frac{6\theta^2(1-\theta)}{0.5} = 12\theta^2(1-\theta)$$

זו פונקציית ההסתברות החדשה, לאחר שראינו את ערך הפרמטר.

■

3.5 Loss-Based View

כעת, מהשאלה: "מה הערך הסביר ביותר?", נעבור לשאלה: "מי הערך שיגרום לי להכי פחות נזק אם אטעה?".

נניח $P(x|\theta)$ כמודל ההסתברותי, $Pr(\theta)$ כ$prior$ וכן $\mathcal{D}$ אלו הנתונים שראינו. כמו כן, תהי $\lambda(\hat{\theta}, \theta^*)$ פונקציית ההפסד. היא מקבלת את הפרמטר שניבאנו $\hat{\theta}$ ואת האמיתי θ^* ומחזירה מס' (קנס). נרצה למצוא אומד $\hat{\theta}$ שיביא למינימום את ההפסד הצפוי. למה צפוי? כי איננו יודעים את θ^*.

מה בידינו? ביידנו ה$posterior$ ($Pr(\theta|\mathcal{D})$) שאומר לנו כמה כל θ הוא סביר בהינתן הנתונים. לכן, נעשה ממוצע (תוחלת) של הנזק ע"פ כל האפשרויות של θ. כך:

$$\hat{\theta} = argmin_\theta \ \mathbb{E}_{\theta|\mathcal{D}}[\lambda(\hat{\theta}, \theta^*)]$$

על מנת לחשב את תוחלת זו, נעזר באינטגרל:

$$\hat{\theta} = argmin_\theta \int \lambda(\hat{\theta}, \theta^*)Pr(\theta|\mathcal{D})d\theta$$

בעברית - עבור על כל ערך של θ אפשרי בעולם, בדוק כמה נזק הוא יגרום לך בעת שסיפקת לו $\hat{\theta}$ שאמור למזער את הנזק, הכפל בסיכוי שזה אכן הθ האמיתי. סכום את הכל: מי שינצח הוא זה שיתן את הערך הנמוך ביותר.

דוגמה 71. הפסד ריבועי.

נניח כי בידינו פונקציית הפסד ריבועי, $\lambda(\hat{\theta}, \theta^*) = (\hat{\theta} - \theta^*)^2$. נרצה למצוא:

$$\hat{\theta}_{SE} = argmin_\theta \int (\hat{\theta} - \theta^*)^2 Pr(\theta|\mathcal{D})d\theta$$

נגזור את הביטוי לפי $\hat{\theta}$. אם כן:

$$\frac{d}{d\hat{\theta}} \int (\hat{\theta} - \theta)^2 Pr(\theta|\mathcal{D})d\theta = 0 \Longrightarrow \int 2(\hat{\theta} - \theta)Pr(\theta|\mathcal{D})d\theta = 0$$

$$\Longrightarrow \int 2\hat{\theta}Pr(\theta|\mathcal{D})d\theta - \int 2\theta Pr(\theta|\mathcal{D})d\theta = 0 \Longrightarrow \int \hat{\theta}Pr(\theta|\mathcal{D})d\theta = \int \theta Pr(\theta|\mathcal{D})d\theta$$

כעת נבחין, בתוך האינטגרל המשתנה שרץ הוא θ, לכן ניתן להוציא את $\hat{\theta}$. מכאן:

$$\hat{\theta} \int Pr(\theta|\mathcal{D})d\theta = \int \theta Pr(\theta|\mathcal{D})d\theta$$

לפי הגדרה, $\int Pr(\theta|\mathcal{D})d\theta = 1$ שהרי זהו סכום הסתברויות ה$Posterior$. ערך זה שווה אחד מהגדרת פונקציית הסתברות. ולכן נקבל:

$$\hat{\theta} = \int \theta Pr(\theta|\mathcal{D})d\theta = \mathbb{E}_{\theta|\mathcal{D}}[\theta]$$

עד כה, בMAP חיפשנו את הערך הסביר ביותר ביחס ל$priors$ ־ זהו ה$MODE$. אך, ראינו כי אם פונקציית ההפסד ריבועית, דווקא נרצה את הממוצע/ תוחלת ־ לא השיא. אינוטואטיבית, במצב של הענשה ריבועית נעדיף להיות בממוצע על מנת "לחטוף כמה שפחות".

■

דוגמה 72. בניגוד לדוגמה הקודמת, נראה כיצד MAP דווקא הוא הכי טוב כשעובדים עם פונקציית Zero to one. אם כן:

$$\hat{\theta_{0-1}} = argmin_\theta \sum_\theta \lambda(\theta, \hat{\theta}) Pr(\theta|\mathcal{D})$$

מכאן, לפי ההגדרה הרי שהעלות היא 0 כאשר צודקים:

$$\sum_\theta \lambda(\theta, \hat{\theta}) Pr(\theta|\mathcal{D}) = \sum_{\theta=\hat{\theta}} \lambda(\theta, \hat{\theta}) Pr(\theta|\mathcal{D}) + \sum_{\theta\neq\hat{\theta}} \lambda(\theta, \hat{\theta}) Pr(\theta|\mathcal{D}) = \sum_{\theta=\hat{\theta}} 0 \cdot Pr(\theta|\mathcal{D}) + \sum_{\theta\neq\hat{\theta}} \lambda(\theta, \hat{\theta}) Pr(\theta|\mathcal{D})$$

$$= \sum_{\theta\neq\hat{\theta}} \lambda(\theta, \hat{\theta}) Pr(\theta|\mathcal{D}) = 1 - \sum_{\theta=\hat{\theta}} \lambda(\theta, \hat{\theta}) Pr(\theta|\mathcal{D})$$

לכן,

$$\hat{\theta_{0-1}} = argmin_\theta (1 - Pr(\hat{\theta}|\mathcal{D}))$$

אם כן, על מנת להביא ביטוי זה למינימום נרצה להביא את $Pr(\hat{\theta}|\mathcal{D})$ למקסימום. הערך שמביא את ה$Posterior$ למקסימום ־ הוא בדיוק MAP. מכאן ש:

$$\hat{\theta}_{0-1} = \hat{\theta}_{MAP}$$

■

3.6 סיכום

הPosterior היא פונקציה שמכילה המון מידע. כדי לקבל החלטה ־ עלינו לשלוף ממנה מס' אחד $\hat{\theta}$. איזה מס' נשלוף? זה תלוי בקנס שאנו מפחדים ממנו. ישנם 3 מקרים עיקריים:

- הפסד ריבועי: אם תטעה קצת, המשמעות שהקנס קטן. אם תטעה הרבה, הקנס עצום. לכן ־ האומד האופטימלי הוא הממוצע שכן הממוצע הוא מרכז המסה.
- הפסד מוחלט: $|\theta - \hat{\theta}|$. הטעות היא למעשה רק המרחק. האומד האופטימלי כאן הוא החציון, מדוע? זהו הערך שממזער את המרחקים המוחלטים.
- הפסד $0-1$: לא משנה כמה טעית, אם לא פגעת בול זה נחשב קנס מקסימלי. במקרה זה: האמד האופטימלי הוא השכיח ־ MAP: הכי הגיוני לבחור את הערך שיש לו את ההסתברות הגבוהה ביותר לקרות, על מנת לא לקבל קנס, זה השיא של הפונקציה ולפי הגדרה בדיוק MAP.

לכן:

כשנרצה להמנע מטעויות ענקיות ־ נשתמש בפונקציית הפסד ריבועי והערך שנשלוף יהיה הממוצע.
כשנרצה להיות באמצע האפשרויות ־ נשתמש בהפסד מוחלט והערך שנשלוף יהיה החציון.
כשנרצה לפגוע בול בסיכוי הכי גבוה ־ נשתמש ב$0-1$ והערך שנשלוף יהיה השכיח.

כפי שכבר דנו, תאוריית ההחלטה של בייס מניחה שאנחנו יודעים הכל. בפועל ־זהו לא המצב. **כאן נכנסת למידת מכונה לתמונה.** ישנן 2 אסטרטגיות עיקריות של למידת העולם:

1. אסטרטגיה גנרטיבית (Generative): לומדים בנפרד איך נראה כלב, איך נראה חתול ־ כשמגיעה תמונה חדשה: משתמשים בחוק בייס על מנת לחשב למי היא דומה יותר.

2. אסטרטגיה דיסקרימינטיבית (Discriminative): בגישה זו לא מעניין כיצד נוצרים חתולים או כלבים. מעניין רק להבדיל בניהם. לומדים ישירות פונקצייה (חוק החלטה) שמקבלת תמונה ומחזירה "כלב" או "חתול". המטרה - מינימום טעויות על הדטה.

לבסוף, נכנסת לסיפור שאלה של Generaliztion:
- Empirical risk: הטעות של המודל על הדטה שבו השתמשת לאימון.
- True risk: הטעות של המודל על כל העולם (הדטה שלא ראית).

תמיד נמדוד את עצמנו על הסיכון האמפירי - אך המטרה שלנו היא שהסיכוי האמיתי יהיה נמוך. השאלה "כמה דטה מספיק?" הוא הבסיס לתאוריה של למידת מכונה.

3.7 PAC Learning

בהרצאה הקודמת למדנו על דרכים להורדת overfitting. אחת הדרכים לעשות זאת, היא להגדיל את כמות הדטה. כעת ננסה לענות על השאלה: כמה דטה צריך מודל בשביל להיות טוב?

נניח כעת את היסודות לSupervised learning (למידה מונחית). נגדיר:
- התפלגות הנתונים $\mathcal{D} = P(x, y)$. נניח שקיים מנגנון סטטיסטי בטבע שמייצר את הדוגמאות שלנו, מנגנון זה יקרא ההתפלגות המשותפת. x הוא הקלט, וy הוא הפלט או התווית. איננו יודעים את ההתפלגות. כלומר כל דגימת אימון היא וקטור שמורכב מ2 חלקים.

y הוא תגית שמוצמדת לx. נזכר כי מתקיים: $Pr(x, y) = Pr(x) \cdot Pr(y|x)$. לכן, אפשר להסתכל על כל הנקודות בעולם כתהליך ייצור דו שלבי:

1. שלב הדגימה - המשתנה x נוצר באופן ב"ת בתגית שלו - קודם מגיע האובייקט.
2. לאחר שx כבר קיים, המנגנון מצמיד לו תווית y שנקבעת בהינתן שידוע כבר הערך של x.

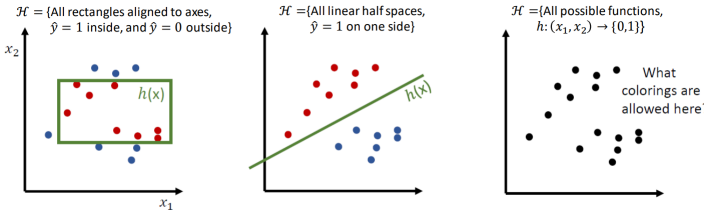
3.7.1 מטרת העל של תהליך הלמידה

אנו מעוניינים לבנות מודל. **פורמלית:** נחפש $h : X \to \hat{Y}$ היפותזה, כך שהקלט x הוא וקטור תכונות והפלט $\hat{y}$ היא התגית הנחזית. המודל צריך להסכים עם הy האמיתי. לשם כך, מוגדרת הloss function: $l(y, h(y))$ שמודדת את המחיר של הטעות.

המטרה: שh הנ"ל תמזער את $\mathbb{E}_{p(x,y)}[l(y, h(y))]$.

נתונים: מדגם $S = \{(x_i, y_i)\}_{i=1}^n$ כך שהדגימות הן $i.i.d$.

כאשר אנחנו מתכננים אלגוריתם למידה, אנחנו מגבילים אותו לבחור היפותזות ממחלקה מסויימת $\mathcal{H}$. לפנינו כמה דוגמאות:
- מרחב כל חצאי המרחבים הלינאריים (קיים קו מפריד במרחב), ומסווגים נקודות לפי המיקום שלהם (מעל או מתחת לקו)
- מרחב כל הפונקציות $h(x_1, x_2) \to \{0, 1\}$. - מה הבעיה כאן? יש לנו את כל הפונקציות האפשריות ואין לנו דרך לסווג נקודות. המודל עלול לשנן את נקודות האימון - אך יכולת ההכללה שלו לא תהיה קיימת על דוגמאות שלא ראה באימון.
- מרחב כל הלבנים שמקבילים לצירים.



כעת נגדיר פורמלית את מדידת הטעות:

הגדרה 73. (Zero-One loss) פונקציית הפסד דיסקרטית למשימת סיווג. מוגדרת כך:

$$\ell_{0-1}(h(x), y) = \left\{ \begin{array}{ll} 1 & h(x) \neq y \\ 0 & h(x) = y \end{array} \right\}$$

מכאן, השגיאה תוגדר באופן הבא:

$$Err_{\mathcal{D}}(h) = \mathbb{E}_{P(x,y)}[\ell_{0-1}(h(x), y)]$$

כלומר: שגיאת ההיפותזה h היא התוחלת מעל מרחב ההתפלגות המשותפת של פונקציית ההפסד.
מה הבעיה? איננו יודעים את $\mathcal{D}$. לכן אי אפשר לחשב את המספר הזה! לכן, נפנה למדגם S שבידינו. ונגדיר:

$$Err_S(h) = \frac{1}{n}\sum_{i=1}^{n} \ell_{0-1}(h(x_i), y)$$

כאחוז שגיאות המודל על סט האימון.

3.7.2 Realizability

נרצה לשאול שאלה לגיטימית. האם בכלל הפתרון המושלם נמצא בתוך מה שאני מחפש? כלומר, ישנה הנחה גורסת לפיה הקשר בין x לy אינו סתם רעש אקראי: אלא ממש חוקיות מוחלטת שניתן לתאר במדויק. לכן נגדיר מחלקת השערות $\mathcal{H}$ ־ קבוצת כל הפונקציות שהאלגוריתם שלנו יכול לבחור מתוכן. הנחת הRealizability־ קיימת $h^* \in \mathcal{H}$ שהיא המתייג האמיתי של הנתונים.

אם הנחת הRealizability אכן מתקיימת ־ זה גורר תוצאה דרמטית על ה$Err_{\mathcal{D}}$. קיים לפחות $h^* \in \mathcal{H}$ אחד כך ש$Err_{\mathcal{D}}(h^*) = 0$. כלומר ־ מודל זה חוזה את כל התגיות בעולם. המשמעות האמיתית: אם הרצנו אלגו' שמוצא $h \in \mathcal{H}$ שמקיים $Err_S(h) = 0$, ואנחנו תחת הנחת Realizability, אזי אנחנו יודעים שקיים פתרון שכזה והשאלה שנשאלת כעת נותרה רק: עד כמה h קרוב לh^* האמיתי.

3.7.3 Empirical Risk Minimization (ERM)

מזער סיכונים אמפירי. מדובר בכלל החלטה ־ כיוון שאיננו מכירים את העולם $\mathcal{D}$, נרצה לבחור את המודל שיודע להצליח הכי טוב עם הנתונים שיש לנו ביד. כלומר,

$$ERM(S) = argmin_{h\in\mathcal{H}} Err_S(h)$$

כלומר, נחפש את ההיפותזה h שתתן את הערך המינימלי. (נבחין ־ ERM לבסוף היא פונקציית היפותזה).

♡ **אם אנחנו מניחים ריאלזיבליות**, בהכרח קיים $h^* \in \mathcal{H}$ עבורו $Err_{\mathcal{D}}(h^*) = 0$. בפרט, הוא מושלם גם על תת קבוצה של דגימות מהעולם (המדגם S) ולכן נקבל כי בהכרח:

$$min_{h\in\mathcal{H}} Err_S(h) = 0$$

המשמעות: במצב ריאליזיבלי, אלגוריתם ERM יחזיר תמיד פונקציה עם אפס שגיאות אימון. אם לא מצאת כזו ־ או שהאלגו' לא סרק את כל $\mathcal{H}$ או שההנחה שגויה.

הערה 74. נשים לב ־ יתכנו מס' פונקציות $h_1, ..., h_k$ עבורן $\forall i \in [k]\ Err_S(h_i) = 0$. אך, יתכן שהן מאוד שונות אחת מהשנייה על נקודות שלא ראינו במדגם ־ וזהו לב ליבו של נושא ההכללה. איך נדע אילו מבין פונקציות ההיפותזה שואף לh^* האמיתי?

משפט 75. השגיאה האמפירית $Err_S(h)$ היא אמד חסר הטיה (*Unbiased*) של $Err_D(h)$.
הוכחה:

$$\mathbb{E}[Err_S(h)] = \mathbb{E}[\frac{1}{n}\sum_{i=1}^{n} \ell_{0-1}(h(x_i), y_i) = \frac{1}{n}\cdot n \cdot Err_D(h) = Err_D(h)$$

שכן, כיוון שהדגימות לקוחות מאותה התפלגות לכולם אותה התוחלת, וכן נעזרנו בליניאריות התוחלת.

■

הערה 76. המשמעות של המשפט הקודם ־ אם היינו לוקחים אנסוף מדגמים שונים בגודל n, ומחשבים לכל אחד את $Err_S(h)$, ממוצע השגיאות הללו היה השגיאה האמיתית.

3.7.4 Version Space and Bad Hypotheses

הגדרה 77. נגדיר את הVersion Space באופן הבא:

$$VS_{\mathcal{H},S} = \{h \in \mathcal{H} : Err_S(h) = 0\}$$

כלומר, כל ההיפותזות כך ששגיאת המדגם עליהם היא אפס.

במבט ריאליזיבלי, $VS_{\mathcal{H},S} \neq \emptyset$ בהכרח - אך יתכן שישנם שם "טרמפיסטים".

הגדרה 78. יהי $\varepsilon > 0$. ותהי היפותזה $h \in \mathcal{H}$. h תקרא השערה רעה אם $Err_{\mathcal{D}}(h) > \varepsilon$.

מתי היפותזה רעה שורדת בתוך הVS? אם למרות שהיא היפותזה רעה, יצא במקרה שהיא לא טעתה אף פעם על המדגם הספציפי S. מתי דבר שכזה יכול לקרות? כלומר, מהי ההסתברות שקיימת $h \in \mathcal{H}$ כך ש: $Err_{\mathcal{D}}(h) > \varepsilon$ וגם $Err_S(h) = 0$.
בעת דגימת מדגם אקראי, תמיד ישנו סיכוי δ שהמדגם הנ"ל מטעה את הERM לבחור היפותזה גרועה. כלומר,

$$Pr[\exists h \in VS : Err_{\mathcal{D}}(h) > \varepsilon] \leq \delta$$

לכן, בהסתברות של לפחות $1-\delta$ נצליח. מכאן: כל הצהרה שניתן על הדיוק של המודל תמיד תהיה מלווה בסייג מסויים (בהסתברות של לפחות $1-\delta$).

הגדרה 79. תהי $\mathcal{H}$ מחלקת היפותזות. $\mathcal{H}$ היא PAC-learnable אם קיים אלגוריתם A שמוצא השערה $h = A(S_n)$ עבור מדגם S_n כך שמתקיים:
- לכל $\varepsilon, \delta > 0$ קיים $N_{\varepsilon,\delta}$ (אשר חייב להיות פולינומי ב$\frac{1}{\delta}, \frac{1}{\varepsilon}$) כך שלכל $n > N_{\varepsilon,\delta}$ ולכל התפלגות $\mathcal{D}$ יתקיים:

$$P(Err_{\mathcal{D}}(A(S_n)) < \varepsilon) > 1 - \delta$$

כאשר ההסתברות מעל מרחבי S_n האפשריים.

אם כן, ישנם שני רכיבים שקובעים האם למידה היא מוצלחת:
רכיב ראשון: δ - קובע את הבטחון. אנחנו לא דורשים כי האלגוריתם יצליח ב100% מהמקרים. מרשים לו להכשל בהסתברות קטנה δ.
רכיב שני: ε - קובע את רמת השגיאה. לא דורשים מודל שיהיה מושלם בעולם האמיתי, ולכן מוכנים לקבל מודל כל עוד השגיאה שלו היא לכל היותר ε קבוע.
סיבוכיות: N הנ"ל חייב להיות פולינומי. מדוע? כי אם בשביל להגדיל את הדיוק פי 2 צריך פי 2^{1000} דגימות - הלמידה הופכת להיות לא פרקטית. PAC דורש שהגידול בכמות המידע יהיה סביר - פולינומי.
לכן, בשורה התחתונה: **עם מספיק נתונים** $n > N$**, אנחנו כנראה (בהסתברות** $1-\delta$ **לפחות) נוכל למצוא השערה שהיא נכונה בקירוב (שגיאה קטנה מ**ε**).**

משפט 80. חסם הדגימה למחלקת היפותזות סופית. בהינתן הנחת הריאליזיבליות, לכל $\varepsilon, \delta > 0$ אם מס' הדגימות n מקיים:

$$n > \frac{1}{\varepsilon} ln(\frac{|\mathcal{H}|}{\delta})$$

אזי לכל התפלגות $\mathcal{D}$ ולכל מדגם S_n מתקיים:

$$P(Err_{\mathcal{D}}(A(S_n)) < \varepsilon) > 1 - \delta$$

הוכחה: נסמן ב$\mathcal{H}_{bad}(D, \varepsilon) = \{h \in \mathcal{H} : Err_{\mathcal{D}}(h) > \varepsilon\}$ את מרחב ההיפותזות הרעות. נרצה לחסום את ההסתברות שהERM יבחר השערה מתוך קבוצה זו:

$$P(Err_{\mathcal{D}}(ERM(S_n)) > \varepsilon) = Pr(ERM(S_n) \in \mathcal{H}_{bad}(D, \varepsilon)$$

כיוון שאנחנו מניחים ריאליזיבליות, הERM תמיד יחזיר השערה המקיימת $Err_{S_n}(h) = 0$. לכן, כדאי שהERM יבחר השערה רעה, חייבת להיות לפחות השערה אחת ששרדה את המדגם עם אפס טעויות. נחזור על זה שוב: מה הERM עושה? הוא מחפש במרחב ההיפותזות השערה שיש לה 0 טעויות על המדגם S_n. בגלל הנחת הריזיבאליות, ישנה אחת כזו לפחות. לכן הוא תמיד יחזיר אחת כזו כך ש$Err_{S_n}(h) = 0$. מתי הוא נכשל? כאשר הוא בחר בחירה רעה גדולה מאפסילון בעולם האמיתי. נבחין כי המאורע משמאל מוכל במאורע מימין. מדוע? נניח כי $ERM(S_n) \in \mathcal{H}_{bad}$. אזי, ה$ERM$ גם בחר השערה רעה, וגם הטעות שלה היא אפס. בפרט - הטעות שלה היא אפס. לכן, אותה השערה נמצאת גם במאורע מימין. סה"כ המאורע משמאל מוכל בימין לכן נקבל אי שוויון:

$$P(ERM(S_n) \in \mathcal{H}_{bad}) \leq P(\exists h \in \mathcal{H}_{bad} : Err_{S_n}(h) = 0)$$

הקיים הופך לאיחוד כמובן. דהיינו, $P(\exists h \in \mathcal{H}_{bad} : Err_{S_n}(h) = 0) = Pr(\bigcup_{h \in \mathcal{H}_{bad}} Err_{S_n}(h) = 0)$ מכאן לפי חסם האיחוד יתקיים:

$$Pr(\bigcup_{h\in\mathcal{H}_{bad}} Err_{S_n}(h)=0) \leq \sum_{h\in\mathcal{H}_{bad}} Pr(Err_{S_n}(h)=0)$$

כעת, נרצה לחשב את ההסתברות עבור השערה רעה בודדת h. ההסתברות שהיא לא תטעה על אף דגימה, וכיוון שהדגימות הן $i.i.d$ היא:

$$Pr(Err_{S_n}(h)) = Pr(\bigcap_{i=1}^{n} h(x_i)=y_i) = \prod_{i=1}^{n} Pr(h(x_i)=y_i) = \prod_{i=1}^{n} 1 - Pr(h(x_i)\neq y_i)$$

$$\leq \prod_{i=1}^{n}(1-\varepsilon) = (1-\varepsilon)^n$$

שכן, כיוון שh רעה, ההסתברות שהיא תצדק על דגימה בודדת היא לכל היותר $1-\varepsilon$ (מדוע? אנחנו מניחים פונקציית הפסד $0-1$ בנושא זה. וכן מתקיים ש$\mathbb{E}[\ell] = P(\ell(x)=y) = 1-\varepsilon$ בפונקציה זו).
אם כן, נחזור:

$$\sum_{h\in\mathcal{H}_{bad}} Pr(Err_{S_n}(h)=0) \leq |\mathcal{H}_{bad}|\cdot(1-\varepsilon)^n$$

קיבלנו:

$$P(Err_{\mathcal{D}}(ERM(S_n))>\varepsilon) \leq |\mathcal{H}_{bad}|\cdot(1-\varepsilon)^n \leq |\mathcal{H}_{bad}|\cdot e^{-n\varepsilon}$$

באשר המעבר האחרון נכון מאי השוויון $1-x\leq e^{-x}$. על מנת שהחסם לטעות יהיה לכל היותר δ נדרוש: $|\mathcal{H}_{bad}|\cdot e^{-n\varepsilon}<\delta$. כלומר אם נפעיל ln על שני האגפים:

$$-n\varepsilon + ln(|\mathcal{H}_{bad}|) < ln(\delta) \iff n > \frac{1}{\varepsilon} ln(\frac{|\mathcal{H}_{bad}|}{\delta})$$

כנדרש.

■

מסקנה 81. כמה מסקנות שנוכל להסיק ממשפט חסם הדגימה:
א. כל מחלקה $\mathcal{H}$ סופית היא $PAC\text{-}Leranable$.
ב. אם n גדל, ε קטן.
ג. אם δ קטן יותר (רמת בטחון גבוהה יותר) אזי נאלץ להתפשר על ε גדול יותר (שגיאה גדולה יותר שאנחנו נכניס לתחום).
ד. ככל שיש יותר השערות, כלומר ככל ש$|\mathcal{H}|$ גדול יותר אזי השגיאה ε גדלה. (די אינטואיטיבי: השגיאה תגדל אם אני צריך לבחור ממקום גדול יותר כי הסיכוי של פונקציה לא טובה להבחר גדול יותר).

הנחנו ריליזיבלי - וזה נחמד, אך זה לא בהכרח המצב. לכן נגדיר:

הגדרה 82. Agnostic הוא מצב שונה מריליזיבלי. קודם לכן הנחנו שקיימת פונקציה מושלמת, כעת נניח שאין כזו. ייתכן שכל ההשערות h יטעו. אנחנו כעת לא מנסים לגרום לטעות ε להתקרב כמה שיותר לאפס אלא להתקרב כמה שיותר להשערה הכי טובה שיש לנו במרחב ההיפותזות. היעד: h^* שיש לה את השגיאה האמיתית הנמוכה ביותר. לא יודעים מיהי, אך שואפים שהERM ימצא משהו שמתקרב לביצועיה. פורמלית:

$$h^* = argmin_{h\in\mathcal{H}} Err_{\mathcal{D}}(h)$$

משפט 83. לכל $\varepsilon,\delta>0$ וכן $n>\frac{1}{\varepsilon^2}ln(\frac{2|\mathcal{H}|}{\delta})$ והתפלגות $\mathcal{D}$ מתקיים:

$$Pr[Err_{\mathcal{D}}(ERM(S_n)) - Err_D(h^*) < \varepsilon] > 1-\delta$$

כלומר, ההפרש בין השגיאה שהאלגוריתם מצא לבין השגיאה של ההשערה הטובה בתאוריה הוא קטן מאפסילון בהסתברות טובה.

3.7.5 אי שוויון הופדינג

בהינתן משתנים $Z_1, ..., Z_n$ שהם ה"הפסדים" שחווינו בכל דגימה, דגימות $i.i.d$ בלתי תלויות, וכן משתנים בטווח סגור בין 0 ל1, אזי: אם נסמן $\overline{Z} = \frac{1}{n}\sum_{i=1}^{n} Z_i$ (במונחי למידה, מדובר בErr_S). מתקיים:

$$Pr[|\overline{Z} - \mathbb{E}[Z]| > \varepsilon] \leq 2e^{-2n\varepsilon^2}$$

3.7.6 הטרייד אוף בין השונות לביאס

משפט 84. בהסתברות של לפחות $1-\delta$ מתקיים:

$$Err_{\mathcal{D}}(ERM(S_n)) < min_{h\in\mathcal{H}} Err_S(h) + \sqrt{\frac{ln(2|\mathcal{H}|/\delta)}{2n}}$$

מה המשמעות של המשפט לעיל? השגיאה האמיתית בעולם, היא לכל היותר השגיאה שמדדת במדגם שלך ועוד "קנס" על מורכבות.

- איבר ה$bias$: $min_{h\in\mathcal{H}} Err_S(h)$, זה כמה טוב המודל הצליח ללמוד את המדגם שנתנו לו. ככל ש$\mathcal{H}$ גדולה יותר, בהכרח ה$Bias$ יקטן. מדוע? כי בתוך קבוצה ענקית של פונקציות, קל למצוא אחת שמתאימה לנתונים שלך (לעומת זאת: במרחב קטן יתכן שאין אחת שתתאים).
- איבר ה$Variance$: $\sqrt{\frac{ln(2|\mathcal{H}|/\delta)}{2n}}$, מדובר במדד הפחד שלנו. כמה אנחנו חוששים שהמדגם שיקר לנו. ככל ש$\mathcal{H}$ גדולה יותר, ה$Variance$ יגדל. מדוע? ככל ששי יותר פונקציות גדל הסיכוי שאחת מהן תצא תותחית על המדגם במקרה למרות שהיא גרועה בעולם האמיתי (יש יותר פונקציות, יתכן מאוד שהרבה מהן מתאימות לנתונים אלו בלבד).

לכן אנחנו בטרייד אוף - אם $\mathcal{H}$ גדולה ההטייה תהיה אפסית אבל השונות בשמיים. לעומת זאת, אם $\mathcal{H}$ קטנה השונות תקטן אבל ההטייה תזנק כי המודל פשוט מדי.

לכן המטרה היא: למצוא את גודל $\mathcal{H}$ בו הסכום של שניהם הכי נמוך. ובמילים פשוטות: אתה רוצה מודל מספיק חכם כדי להבין את החומר, אבל לא כזה חכם שהוא יתחיל להמציא תיאוריות קונספירציה על סמך כמה דגימות בודדות.

4 הרצאה 4

4.1 מבוא

בהרצאה זו נדון במודל נפוץ בלמידת מכונה, בשם רגרסיה לינארית ועל הרחבה חשובה בשם Ridge Regression. לפני שנקפוץ למתמטיקה - נדון בשאלה חשובה: מדוע שנסתכל על מודלים לינאריים?

- לפי תאוריית התחום שלמדנו בPAC Learning, נעדיף מרחבי היפותזות פשוטים שכן נרצה לעבוד עם פונקציות שאינן מסובכות מדי. כמו כן, ככל שהמודל המתמטי פשוט יותר שגיאת ההכללה שלו "הדוקה יותר", כלומר: הפער בין השגיאה על האימון לבין השגיאה בעולם האמיתי יהיה קטן יותר מה שיוריד את הסיכון לOverfitiing (למידת יתר של הרעש בנתונים במקום התבנית).
- עבור נתונים רציפים, מרחב ההיפותזות הלינארי הוא המרחב הטבעי והפשוט ביותר עבור נתונים רציפים. אם יש רצף מספרים - קו ישר לינארי הוא הדרך הבסיסית לתאר קשר בניהם.
- רגרסיה לינארית הוא המודל היסודי ביותר בלמידה מונחית (למידה מתוך דוגמאות מתויגות של זוגות (X, y)).
- קל מאוד לחשב את המודל.

4.2 רגרסיה לינארית: הגדרת הבעיה

בסיס הנתונים - נגדיר קבוצת נתונים שנסמנה $\mathcal{D}$:

- $\{(x_i, y_i)\}_{i=1}^{n}$ כn דוגמאות אימון, כל דוגמה היא זוג של קלט ותיוג.
- הקלט $x_i \in \mathbb{R}^d$ הוא וקטור תכונות, באשר d מייצג את המימד (כמה מאפיינים יש לכל דוגמה: גובה, משקל, וכו').
- הפלט: $y_i \in \mathbb{R}$, ערך רציף שאנחנו מנסים לחזות.

המודל הלינארי: המודל מורכב משני מאפיינים עיקריים -

- וקטור משקלים $w \in \mathbb{R}^d$, לכל תכונה בx יש משקל תואם בw.
- הטיה - הBias, מדובר בסקלר $b \in \mathbb{R}$ שנקרא גם Intercept ומהווה את נק' החיתוך עם הציר.

משוואת החיזוי: עבור דוגמה x_i החיזוי של המודל יסומן ב$\hat{y}_i$ ויתקיים:

$$\hat{y}_i = f(x_i) = w^T x_i + b$$

מתמטית, נבחין כי ערך זה שקול ל:

$$\hat{y}_i = f(x_i) = \sum_{j=1}^{d} w_j x_{i,j} + b$$

לצורך דוגמה, נניח ואנחנו מעוניינים לחזות מחירו של בית בהינתן הרבה נתונים: שטח הבית, מס' החדרים, מס' חדרי השירותים, האם יש בריכה, כמה חניות יש? וכדומה, נקבל כל נתון כוקטור, נראה כי ערכים כמו "יש בריכה?" שתשובה אליהם תהיה כן/לא (בינארית) נרצה לסווג לערך נומרי - ע"י (למשל) לקבוע כי $yes = 1, no = 0$. זה די קשה להמיר זאת - **ולכן נניח בקורס כי בידינו d פיצ'רים נומריים.**

4.3 הגדרה מטריציונית

נרצה להעלים את הb, על מנת להקל עלינו במתמטיקה בהמשך. לכן: נרצה "לבלוע" את הBias לתוך וקטור המשקלים. לכן, נשנה את x_i ע"י כך שנוסיף לו קורדינטה ראשונה להיות תמיד 1, וכעת יתקיים: $x_i \in \mathbb{R}^{d+1}$. וכן, נגדיר $w_1 = b$ בוקטור המשקלים וכעת יתקיים $w \in \mathbb{R}^{d+1}$ כאשר $w = (w_1, w_2, ..., w_{d+1})^T$. כעת, נבחין כי ההגדרות אכן שקולות מתמטית ומתקיים:

$$f(x_i) = w^T x_i = \sum_{j=1}^{d+1} w_j x_{i,j} = w_1 x_{i,1} + \sum_{j=2}^{d+1} w_j x_{i,j} = b + \sum_{j=2}^{d+1} w_j x_{i,j}$$

כעת נקבל:

$$\hat{y}_i = w^T x_i$$

כמו כן, במקום לחשב חיזוי לכל דוגמה $i \in [n]$ בנפרד נרצה לחשב את כל החיזויים למדגם בבת אחת. לכן נגדיר:

- X תהיה מטריצה מגודל $n \times (d+1)$, כאשר כל שורה במטריצה היא וקטור תכונות של דוגמה אחת $x_i \in \mathbb{R}^{d+1}$.
- $\hat{Y}$ יהיה וקטור החיזויים כך ש$\hat{Y}_i = \hat{y}_i = f(x_i)$ לכל $i \in [n]$.

נקבל:

$$\hat{Y} = Xw$$

4.4 מציאת הפתרון האופטימלי

נבחין כי מטרתנו תהיה למזער את השגיאות של המודל, כלומר נרצה למזער את ערך פונקציית ההפסד, הMSE. נזכר כי:

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

בצורה מטריציונית, נכתוב זאת כך:

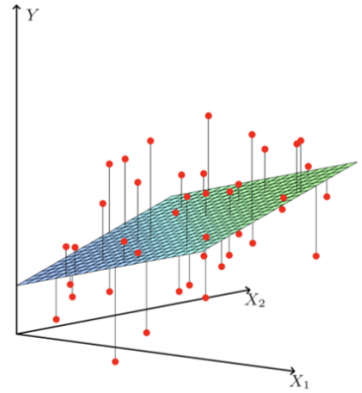
$$MSE = \frac{1}{n} (y - Xw)^T (y - Xw)$$

נחפש את וקטור המשקולות $\hat{w}$ כך שיתקיים:

$$\hat{w} = argmin_w MSE(w)$$

נבחין כי MSE היא פונקציה קמורה וריבועית של w, כלומר יש לה ערך מינימום גלובלי יחיד. לכן, נקח את הגרדיאנט ונשווה לאפס. כלומר: $\nabla_w MSE = 0$.

מה המטרה שלנו בעצם? למצוא את הקו שסכום המרחקים מהקו יהיה מינימלי, כיצד זה נראה גאומטרית מממימד גדול יותר?



המישור שבתמונה הוא המישור שבו מרחב הפתרונות שלנו נמצא, ונרצה למצוא מישור שבו סכום ההטלות של כל הדגימות מהמישור יהיה **קטן ככל הניתן**.

ובכן, נתחיל בלפשט את הביטוי:

$$MSE(w) = \frac{1}{n}(y - Xw)^T(y - Xw) = \frac{1}{n}(y^T - w^T X^T)(y - Xw) = \frac{1}{n}(y^T y - y^T Xw - w^T X^T y + w^T X^T Xw)$$

$$= \frac{1}{n}(y^T y - 2w^T X^T y + w^T X^T Xw)$$

באשר המעבר האחרון דורש הסבר. נבחין, $y^T Xw$ הוא סקלר. מדוע? $y^T \in \mathbb{R}^{1\times(d+1)}$ וכן $Xw \in \mathbb{R}^{d+1\times 1}$ ולכן $y^T Xw \in \mathbb{R}$. באופן דומה, $w^T X^T y$ הוא סקלר. כיוון שלכל סקלר מתקיים $c = c^T$ אם נפעיל על הראשון טרנספוז נקבל: $(y^T Xw)^T = w^T X^T y$, ומכאן שהמעבר האחרון די ברור.

נגזור איבר איבר:

- $\nabla(y^T y) = 0$ שכן גוזרים לפי w.
- $\nabla(-2w^T X^T y) = -2X^T y$, שכן: אם נסמן $a = X^T y$ נקבל $-2w^T a$. נזכר כי $\nabla_w(w^T a) = a$ ונקבל שהנגזרת היא $-2a = -2X^T y$.
- $\nabla(w^T X^T Xw) = 2X^T Xw$. שכן: עבור A סימטרית מתקיים $\nabla(w^T Aw) = 2Aw$, וכן $X^T X$ תמיד מטריצה סימטרית לפי לינארית 2, ומכאן שהנגזרת אכן $2X^T Xw$.

סה"כ קיבלנו:

$$\nabla_w(MSE) = \frac{2}{n}(X^T Xw - X^T y) = 0 \iff X^T Xw = X^T y$$

מדובר בפתרון הסופי. כעת, אם נניח כי $X^T X$ היא מטריצה הפיכה אכן נקבל ע"י הכפלה ב$(X^T X)^{-1}$ ונקבל:

$$\hat{w} = (X^T X)^{-1} X^T y$$

כעת, כאשר נרצה לחזות את ערך הy_{new} על דטה חדש, כלומר X_{new} כל שנעשה יהיה:

$$Y_{new} = X_{new}\hat{w} = X_{new}[(X^T X)^{-1} X^T y]$$

4.5 $\hat{y}$ הוא הטלה אורתוגונלית ובפרט ממזער את השגיאה

כעת נטען, כי החיזוי $\hat{y}$ הוא ההטלה האורתוגונלית של y על המישור של X.

הוקטור y הוא וקטור שמייצג את התשובה האמיתית שקיבלנו מהנתונים. הוא נמצא איפשהו במרחב. המטריצה X מגדירה באמצעות העמודות שלה מישור בתוך המרחב. החיזוי $\hat{y}$ הוא מוגדר לפי המודל $y = Xw$ ולכן בהכרח $\hat{y}$ הוא קומבינציה לינארית של עמודות X , כלומר $\hat{y}$ חייב להיות נקודה שיושבת במישור של X. מה הבעיה? בדרך כלל ־ הוקטור y לא נמצא בדיוק על המישור של X. המטרה שלנו ברגרסיה היא למצוא נקודה בתוך המישור $(\hat{y})$ שתהיה הכי קרובה לy שנמצא מחוץ למישור.

כיצד נמצא את הנקודה הכי קרובה לy מחוץ למישור? מורידים אנך, בהכרח זו הנקודה הכי קרובה למישור מחוץ אליו. מדובר בקו ישר שמחבר את y למישור בזווית של 90° ויפגוש את $\hat{y}$ בזווית זו. המרחק הזה (הקו שמחבר בין השניים) הוא וקטור השגיאה $e = y - \hat{y}$.
מעבר לאינטואיציה, נראה הוכחה פורמלית של הדבר:
נזכר ־ שני וקטורים מאונכים אמ"מ המכפלה הפנימית בניהם שווה לאפס. נרצה לבדוק את המכפלה בין וקטור השגיאה e לבין החיזוי $\hat{y}$. אם כן:

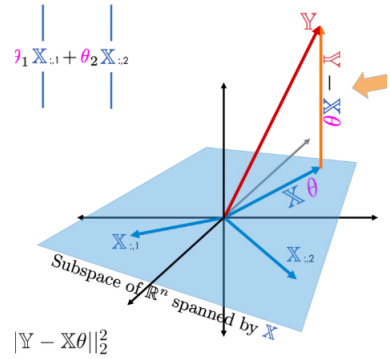
$$< y - \hat{y}, \hat{y} >= (y - \hat{y})^T \hat{y} = (y - \hat{y})^T X\hat{w} = y^T X\hat{w} - \hat{y}^T \hat{y} =$$

נבחין כי $\hat{y}^T \hat{y}$ ניתן לפירוק כך: $\hat{y}^T = (X\hat{w})^T = \hat{w}^T X^T$ ומכאן: $\hat{y}^T \hat{y} = \hat{w}^T X^T X\hat{w}$, נציב מפורשות את $\hat{w}$ משמאל: $(y^T X(X^T X)^{-1})X^T X\hat{w}$.
נראה כי קיבלנו $(X^T X)^{-1})X^T X = I$ ומכאן: $y^T X\hat{w}$

$$= y^T X\hat{w} - y^T X\hat{w} = 0$$

כלומר הוכחנו כי $< y - \hat{y}, \hat{y} >= 0$.

כלומר, הראינו כי וקטור השגיאה מאונך לוקטור החיזוי. וכיצד זה עוזר? זה גורר כי בהכרח מצאנו את הפתרון האופטימלי במסגרת הפתרון הלינארי ־ שכן המרחק הקצר ביותר בין נקודה (y) לבין מישור (מרחב ההשערות של המודל, X) הוא הקו המאונך אליו. כלומר הוכחנו פורמלית כי $\hat{y}$ הוא הנקודה במישור שמרחקה מy מינימלית.
במילים אחרות: **רגרסיה לינארית היא הטלה אורתוגונלית על המרחב** X.



4.6 רגרסיה פולינומיאלית

כיצד נשתמש במודל לינארי על מנת ללמוד על קשרים שאינם לינאריים? נשתמש ברגרסיה פולינומיאלית, ובאופן מפתיע כלל לא נצטרך לשנות את המודל אלא בסך הכל להתאים לו את הנתונים. לכן, אנחנו ניצור תכונות חדשות ע"י העלאה בחזקה של הקלט המקורי שלנו. כלומר, אם קיבלנו קלט $x \in \mathbb{R}$ נתאים לו אחים $x^1, x^2, ..., x^k$ וניצור פולינום מדרגה k:

$$\hat{y} = w_0 + w_1 x^1 + ... + w_k x^k$$

למרות שהגרף נראה כעקומה, המודל **כן לינארי** ביחס לנתונים שכן "x^2" מבחינת המודל הוא תכונה של הוקטור, ולא העלאה בריבוע של הקלט.
נבנה מטריצה חדשה:

$$X = \begin{pmatrix} 1 & x_1 & x_1^2 & \dots & x_1^k \\ 1 & x_2 & x_2^2 & & x_2^k \\ . & \dots & \dots & & \dots \\ . & \dots & \dots & & \dots \\ 1 & x_{n-1} & x_{n-1}^2 & & x_{n-1}^k \\ 1 & x_n & x_n^2 & \dots & x_n^k \end{pmatrix}$$

זו מטריצת ונדרמונדה על הערכים $(x_1, ..., x_k)$. כעת, וקטור התחזיות יקיים גם הוא $\hat{w} = (X^T X)^{-1} X^T Y$

4.7 Ridge Regression

קיים פתרון חיזוי $\hat{w}$ אם, ורק אם $X^T X$ היא מטריצה הפיכה. אם כן: מטריצה זו היא הפיכה אמ"מ עמודות המטריצה X ב"ת לינארית. כלומר ־ X צריכה להיות מדרגת עמודות מלאה. (אין אף תכונה "מיותרת" שניתן להרכיבה משילוב תכונות אחרות). ישנם שני מקרים בהם לא קיימת הופכית:

- $d > n$, במקרה זה יש לנו יותר תכונות מאשר דטה אימון. לצורך ההמחשה ־ אם ננסה להעביר קו ישר דרך נקודה אחת. יש אנסוף קווים כאלו. יש לנו עודף של חופש ־ ואין פתרון.
- $n > d$ אך אחת התכונות היא שילוב מושלם של אחרות. **זה נקרא Multicollinearity**. למשל: אם יש לנו עמודה של גובה בס"מ וגובה באינץ'. אלו נתונים כפולים ־ שיוצרים במטריצה לאחר דירוג שורת אפסים ־ והופכים אותה ללא הפיכה.

לשם כך, יש לנו פתרון: Shrinkage. במקום לחפש את השגיאה הכי נמוכה (MSE), נוסיף "קנס" על גודל המשקלים. הוספת קנס שכזה גורמת למטריצה להפוך להפיכה תמיד ־ והבעיה הזו נפתרת לנו. זה כאילו שהוספנו ערך קטן לאלכסון המטריצה, מה שמבטיח שנוכל לחשב את $\hat{w}$. כמו כן ־ הקנס הנ"ל מושך את המשקלים כלפי מטה ומונע מהם לגדול ללא שליטה. זה הופך את המודל **לרובוסטי**, הוא פחות רגיש לתנודות קטנות בנתונים. לפתרון זה קוראים Ridge Regression:

$$\hat{w}_{ridge} = argmin_w[(y - Xw)^T (y - Xw) + \lambda ||w||_2^2]$$

באשר, $\lambda > 0$ הוא היפר פרמטר ששולט בכמה שאנחנו "מכאיבים" למודל על משקולות גדולות. וכן, $||w||_2^2 = \sum_{i=1}^d w_i^2$.
על מנת למזער ביטוי זה, נגזור אותו: כפי שראינו קודם הנגזרת ללא הλ היא: $2X^T Xw - 2X^T y$ וכן

$$\nabla_w(\lambda ||w||_2^2) = 2\lambda w$$

לכן הנגזרת:

$$X^T Xw - X^T y + \lambda w = 0 \Longrightarrow (X^T X + \lambda I)w = X^T y \Longrightarrow \hat{w} = (X^T X + \lambda I)^{-1} X^T y$$

משפט 85. לכל מטריצה X ולכל $\lambda > 0$ מתקיים כי $X^T X + \lambda I$ הפיכה.
הוכחה: אנו יודעים כי הערכים העצמיים של $X^T X$ הם תמיד אי שליליים, שכן:

$$a < v, v >=< av, v >=< X^T Xa, v >=< Xv, Xv >= ||Xv||^2 \Longrightarrow a = \frac{||Xv||^2}{< v, v >}$$

אם $a = 0$, אזי כעת ע"ע תואם של $X^T X + \lambda I$ יהיה $\lambda > 0$. אחרת, נקבל כי בהכרח $< v, v >> 0$ ולכן $a > 0$. סה"כ הע"ע של $X^T X$ הם אי שליליים.
מכאן, כיוון ש$\lambda > 0$ בהכרח לכל ע"ע a של $X^T X$ מתקיים כי $a + \lambda$ הוא ע"ע של $X^T X + \lambda I$ וכן כעת $a + \lambda > 0$ ולכן הע"ע של המטריצה בהכרח אינם אפס, ולכן היא הפיכה.

■

4.8 בפרספקטיבה של SVD

כל מטריצה (מעל המרוכבים) ניתנת לפירוק לשלוש מטריצות $X = UDV^T$ כך ש:
- U היא מטריצה שמורכבת מוקטורים עצמיים או"ג של X^*X. מטריצה אורתוגונלית, כלומר $U^TU = I$.
- D היא מטריצה אלכסונית של ערכים סינגולריים σ_i שקובעים כמה כל כיוון בנתונים הוא חזק / משמעותי.
- V היא מטריצת הכיוונים של הנתונים. מטריצה אורתוגונלית, כלומר $V^TV = I$.

נתבונן בX^TX:

$$X^TX = (UDV^T)^T(UDV^T) = VD^TU^TUDV^T = VD^TDV^T = VD^2V^T$$

כאשר D^2 היא מטריצה חדשה שמוגדרת להיות $D^2 = D^TD$ שכן היא אלכסונית ריבועית, ועל אלכסונה הערכים σ_i^2. כעת, נסתכל על הפתרון לפי Ridge Regression:

$$X^TX + \lambda I = VD^2V^T + \lambda I = VD^2V^T + \lambda VIV^T = V(D^2 + \lambda I)V^T$$

כעת, נסתכל על המטריצה ההופכית למטריצה זו. לפי הכלל $(\prod_{i=1}^n A_i)^{-1} = A_n^{-1}A_{n-1}^{-1} \cdot .... \cdot A_1^{-1}$ נקבל:

$$(X^TX + \lambda I)^{-1} = [V(D^2 + \lambda I)V^T]^{-1} = (V^T)^{-1}(D^2 + \lambda I)^{-1}V^{-1}]$$

אך כיוון שV אורתוגונלית מתקיים $V^TV = I$ ולכן $V^{-1} = V^T$ וכן $(V^T)^{-1} = V$ כלומר:

$$(X^TX + \lambda I)^{-1} = V(D^2 + \lambda I)^{-1}V^T$$

כעת,

$$\hat{y} = X \cdot (X^TX + \lambda I)^{-1} \cdot X^Ty = UDV^T \cdot V(D^2 + \lambda I)^{-1}V^T \cdot VD^TU^Ty =$$

$$UD(D^2 + \lambda I)^{-1}DU^Ty$$

כעת, D ו$(D^2 + \lambda I)^{-1}$ הן מטריצות אלכסוניות. לכן מכפלתן $D(D^2 + \lambda I)^{-1}D$ לפי הגדרה ישירה ולכל איבר על האלכסון מתקיים:

$$\sigma_i \cdot \frac{1}{\sigma_i^2 + \lambda} \cdot \sigma_i = \frac{\sigma_i^2}{\sigma_i^2 + \lambda}$$

מכאן, כיוון שU היא מטריצה של וקטורים אורתוגונליים שמהווים בסיס, אשר נסמנו $\{u_1, ..., u_d\}$ נקבל:

$$\hat{y}_{ridge} = \sum_{i=1}^{d} u_i(\frac{\sigma_i^2}{\sigma_i^2 + \lambda})u_i^Ty$$

מה זה אומר בתכלס?
- הביטוי $u_i^Y y$ הוא ההטלה של y על הכיוון הi. המקדם, הוא "הוליום" בו אנחנו משאירים את המידע הנ"ל. אם σ_i גדול ביחס לλ השבר קרוב לאחד (המידע כמעט יעבור במלואו), ואם σ_i קטן מאוד (רעש) השבר שואף לאפס. (המידע נמחק).

4.9 Binary classification

קלט: נניח סט של דוגמאות מתויגות $\{(x_i, y_i)\}_{i=1}^n$. במקרה זה, נדון בבעיית סיווג. כלומר y_i איננו ערך מספרי כמו ברגרסיה אלא שיוך לקבוצות. לכל $i \in [n]$ $x_i \in \mathbb{R}^d$. וכן, $y_i \in \{+1, -1\}$ (למשל: חולה, בריא וכו').

מטרתנו **כפלט** למצוא פונקציה $f : X \to Y$ כך ש:

1. f מסכימה עם נתוני האימון, כלומר עבור x_i שראינו תוציא y_i תואם לו.
2. תבצע הכללה - בהינתן $x \notin \{x_i\}_{i=1}^n$ f תנחש נכונה אם $y = 1$ או $y = -1$ בהתאמה.

נבחין כי בסיווג, בניגוד לרגרסיה שם חיפשנו פונקציה רציפה שתעבור בין כל הנקודות, כאן נרצה למצוא מעין "קו" שחותך את המרחב ל2: כל מי שמעליו $+1$ ומתחתיו -1.

4.10 Linear-threshold neuron

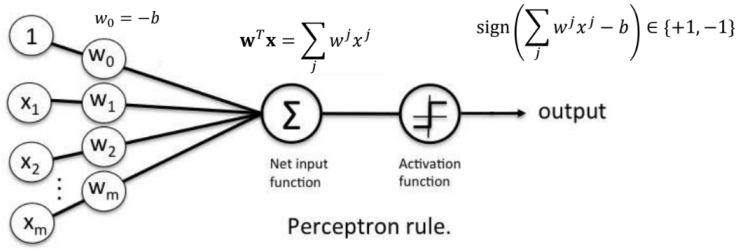
כעת, נראה כיצד מערכת מיוחדת - הניורון הבודד, הופכת קלט מספרי להחלטה של כן או לא. התהליך מתבצע בשלבים הבאים:

- קלט x_j: תכונות $(x_1, ..., x_m)$.
- משקולות: כל תכונה מקבלת משקל כמה היא משפיעה על ההחלטה.
- ה$bias$: מתקיים $w_0 = -b$, שמתאים לערך $x_0 = 1$. מדובר בסף (Threshold): כמה חזק צריך להיות הקלט כדי שהניורון ירה.
- הניורון מחשב ממוצע משוקלל של התכונה כתלות במשקל כך: $w^T x = \sum_j w^j x^j$.
- על מנת לעבור ממס' רציף לסיווג בינארי משתמשים בפונקציית $sign$. שמוגדרת כך: $sign(\sum_j w^j x^j - b)$. אם תוצאת הסכימה גדולה מהסף b נקבל 1 ואחרת אפס. מעט יותר פורמלית:

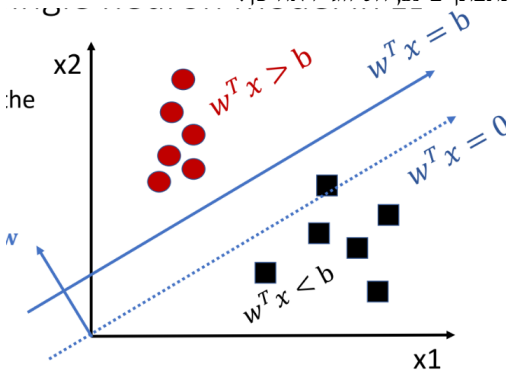
$$h_{w,b}(x) := \left\{ \begin{array}{ll} +1 & w^T x > b \\ -1 & w^T x < b \end{array} \right\} = \left\{ \begin{array}{ll} +1 & \sum_j w^j x^j - b > 0 \\ -1 & \sum_j w^j x^j - b \leq \end{array} \right\}$$

את הפונקציה $sign$ נגדיר באמצעות מרחב ההיפותזות - $h_{w,b}(x)$. כך שבהינתן w, b וx נחזיר פלט בהתאם. גאומטרית: אנו מותחים קו ישר במישור, מעליו $(w^T x - b > 0)$ נסווג 1 ומתחתיו 0.

חשוב להבחין כי w^j - הרכיב הj בוקטור w. לא חזקה.



אם נתבונן ב$\mathbb{R}^2$, הסיווג יראה כך:



נדמיין מצב של סוכריות כחולות ואדומות. אם נמצא קו ישר שנעבירו וכל מה שיהיה מתחתיו יסווג לצבע אחד ומתחתיו לצבע שני - הצלחנו. מתי מתחילה הבעיה? מה אם הסוכריות הכחולות נמצאות במרכז, והאדומות מקיפות אותן במעגל? לא משנה מה - לא נצליח להעביר קו ישר. מכאן שנקבל מסקנה: ניורון בודד יכול להגיע לדיוק מושלם אם ורק אם הנתונים פרידים לינארית. ניורון בודד לא מסוגל להתעקם, לצייר עיגול וכו'.

כעת, נבצע טריק מתמטי דומה לזה שביצענו ברגרסיה: נגדיר מעתה כי $x = (1, x_1, ..., x_d)^T$ וכן $w = (-b, w_1, ..., w_d)^T$. כעת, $w^T x$ מייצג את אותו הביטוי כמו קודם.

כשהיה לנו b נפרד, קו ההפרדה היה יכול להיות בכל מקום במרחב. כעת - במרחב החדש שיצרנו: המישור המפריד תמיד עובר דרך הראשית.

4.11 The perceptron algorithm (אלגוריתם הפרספטרון)

מדובר באחד מן האלגוריתמים הוותיקים והאלגנטיים ביותר בבינה מלאכותית. נציג את הפסודו הבא:

Init w randomly
while not all samples classified correctly:
for i=1,...,n
predict $\hat{y}_i = sign(w^T x_i)$
if $y_i \neq \hat{y}_i$:
1. if $y_i > 0 : w = w + x_i$
2. o.w: $w = w - x_i$

מדובר באלגוריתם עבור הגרסה שבה $y_i \in \{+1, -1\}$. הגרסה שבה $y_i \in \{0, 1\}$ מעט שונה.

מה עושה האלגוריתם?

האלגוריתם מאתחל וקטור משקולות אקראי w, בשלב זה כנראה שנטעה המון. האלגוריתם עובר שוב ושוב על כל הדגימות - עד שכל הדגימות מסווגות נכון. עבור כל דגימה (x_i, y_i):

א. מנבאים את מה שהמודל יגיד כרגע

ב. בודקים טעות: אם המודל צדק, ממשיכים הלאה. אם המודל טעה - מעדכנים בהתאם: אם היינו אמורים לקבל $+1$ וקיבלנו -1 נוסיף x_i למשקולות. אם היינו אמורים לקבל -1 וקיבלנו $+1$ נחסיר את x_i מהמשקולות.

נבחין כי ניתן להבין את האלגוריתם משתי דרכים:

- כשמחסירים או מוסיפים x_i לw - מסובבים את וקטור המשקולות w לכיוון הנקודה שבה טעינו. כלומר: מזיזים את קו ההפרדה כך שבפעם הבאה "יחבק" את הנקודה הזו בצד הנכון שלה.
- נסיון למזער פונקציית הפסד - כל פעם שיש טעות: מנסים לעשות צעד אחד בכיוון של הקטנת הטעות.

כעת ננסה לענות על השאלה הבאה:

מדוע הפעולה הפשוטה של חיבור או חיסור וקטורים גורמת למודל ללמוד?

נבחין שבזכות הקשר הבא ניתן להבין זאת:

$$w^T x = ||w|| \cdot ||x|| \cdot cos(\theta)$$

כאשר θ היא הזווית בין הוקטורים x, w.

- אם הזווית בין הוקטורים קטנה מ90 - $cos(\theta) > 0$ ואז $w^T x > 0$ ולכן החיזוי הוא $+1$.
- אם זווית זו קטנה גדולה מ90 (קהה): $cos(\theta) < 0$ ואז $w^T x < 0$ ולכן החיזוי הוא -1.

מה קורה כאשר בידינו טעות?

נניח כי ישנה נקודה x_i שהתווית האמיתית שלה $y_i = +1$ אך המודל חזה -1. משמעות הדבר: הזווית בין w לx גדולה מדי! לכן, אנחנו מעדכנים $w_{new} = w + x_i$. מה קורה למכפלה הסקלרית כעת?

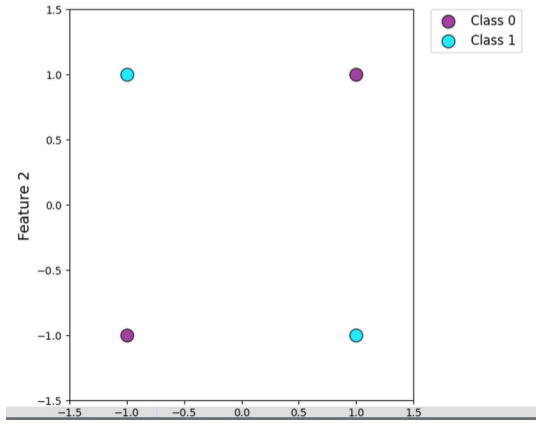
$$w_{new}^T x_i = (w + x_i)^T x_i = w^T x_i + x_i^T x_i = w^T x_i + ||x_i||^2$$

כיוון ש$||x_i||^2 > 0$ אנחנו תמיד מגדילים את ערך המכפלה הסקלרית. הגדלת המכפלה - לפי הקשר ששמנו לב אליו: הקטנת הזווית - כלומר - גורמים לחיזוי לשאוף כעת ל1! הוקטור מסתובב לכיוון הנקודה שבה טעה על מנת שבפעם הבאה יהיה קרוב יותר לצד הנכון שלה.

נכונות: לא נראה נכונות, אך ישנה הוכחה לכך שאם הנתונים אכן פרידים לינארית - האלגוריתם עוצר ומגיע ל0 טעויות.

כעת, נתבונן בדוגמה מאוד פשוטה מתי האלגוריתם לא יעבוד. מתי? כאשר הנתונים לא פרידים לינארית.

נתבונן בXor: אי אפשר להעביר שום קו בין הנקודות ולכן הפרספקטון לא יתכנס - שכן הנתונים **לא** פרידים לינארית.



4.12 Gradient Descent

בסופו של דבר, למידה היא בסך הכל בעיית אופטימיזציה של מינימום. בפתרונות האנליטיים של אלגברה לינארית ניתן לגזור, למצוא נוסחה סגורה ע"י השוואה לאפס. אבל ־ כאשר המודלים הופכים למורכבים יותר ויותר בלתי אפשרי לפתוח זאת בדף ועט. לכן: עוברים לשיטה איטרטיבית של ניסוי וטעייה חכמים.

אך, על מנת שהמחשב יוכל ללמוד עלינו לתת לו שני דברים:

- פונקציית הפסד: פונקציה $L(w)$ שמקבלת את המשקולות ומחזירה מס' בודד. המס' הזה אומר לנו כמה המודל גרוע. מטרתנו ־ למזער את הערך הזה.
- אלגוריתם חיפוש: שיטה למצוא את הw שנותן את הL המינימלי.

אם פונקציית ההפסד היא דיפרנציבאלית ־ אזי ניתן לגזור אותה. ניתן להשתמש בנגזרות על מנת לדעת להיכן ללכת. הגרדיאנט מצביע לכיוון שבו הפונקציה עולה הכי מהר. ונעזר בכך.

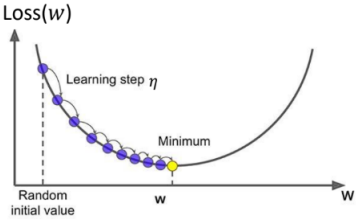
על מנת להגיע למינימום של פונקציית ההפסד, אנחנו נבצע עדכון איטרטיבי למשקולות. בכל שלב:

$$w \leftarrow w - \eta \frac{\vartheta Loss}{\vartheta w}$$

נפרק את מה שכתוב כאן:

- $\frac{\vartheta Loss}{\vartheta w}$: זהו הגרדיאנט, וקטור הנגזרות החלקיות. הוא מצביע על הכיוון בו הפונקציה עולה הכי מהר.
- הסימון $-\eta$, כיוון שאנחנו רוצים למזער ולהגיע למינימום, **נרצה ללכת בכיוון ההפוך לכיוון הגרדיאנט!** η הוא גודל הצעד ־ הוא קובע כמה אנחנו סומכים על הכיוון שמצאנו בכל פעם.

לצורך ההמחשה, ניתן לראות כמעין כדור שמתגלגל במורד פרבולה. אם η קטן מדי , נגיע למינימום ־ אך זה יקח הרבה מאוד זמן (צעדים). אם η גדול מדי ־ יתכן שנדלג מעל המינימום. בכל שלב אנחנו יורדים מכיוון הגרדיאנט, על מנת להגיע למינימום.



מתי זה בוודאות יעבוד לנו?

1. אם הפונקציה תהיה קמורה חזקה (כמו קערה), בפונקציות שכאלו אין בורות מקומיים שנתקע בהם ויש מינימום גלובלי יחיד.
2. חסם על גודל הצעד: $0 < \eta < \frac{1}{L}$ באשר L הוא קבוע ליפשיץ' של הגרדיאנט. מדובר במדד לכמה מהר השיפוע משתנה. נרצה שלא להשתגע ־ לקחת צעדים קטנים מאוד כשי לא לאבד שליטה.

4.13 GD במימד גבוה $\mathbb{R}^d$

נזכר כי כאשר וקטור המשקולות מכיל הרבה פרמטרים $\{w_i\}_{i=1}^d$ לא נוכל להסתפק בנגזרת אחת ולשם כך מוגדר הגרדיאנט:

$$\nabla l = (\frac{\vartheta l}{\vartheta w_1}, \frac{\vartheta l}{\vartheta w_2}, ..., \frac{\vartheta l}{w_d})^T$$

אם הגרדיאנט רציף, הוא מצביע לכיוון העלייה הכי חדה. לכן, באופן שקול, $-\nabla l$ הוא הכיוון היעיל ביותר להקטנת ההפסד באופן מקומי. כעת כלל העדכון יהיה:

$$w_{t+1} \leftarrow w_t - \eta \nabla l$$

באשר:
- w_{t+1} הם המשקולות החדשות.
- w_t הם המשקולות הנוכחיות.
- η קובע כמה "חזק" נדחוף את כל המשקולות יחד לכיוון החדש.

נבחין כי התהליך מתבצע באופן איטרטיבי, והעדכון קורה לכל המשקולות בו זמנית.

4.14 Stochastic Gradient Descent (SGD)

ישנה בעיה: על מנת לבצע עדכון אחד בודד של המשקולות, עלינו לחשב את הגרדיאנט על פני כל n הדגימות במערכת:
- $TotalError$: $\sum_{i=1}^{n} l(w, x_i, y_i)$ - סכום כל ההפסדים של כל הדגימות.
- עדכון המשקולות: $w^{t+1} = w^t - \eta \nabla TotalError(w^t)$ (פחות הגרדיאנט של פונקציית ההפסד על המשתנה w^t).

כאשר n הוא גדול מאוד - חישוב הגרדיאנט הופך להיות מתיש חישובית עבור עדכון בודד.

הפתרון: במקום לחשב את הגרדיאנט על כל המידע, אנחנו עושים קירוב באמצעות קבוצה קטנה **ואקראית** של דגימות שנקראת mini batch (גודל הקבוצה b יקיים $b << n$). כעת,
$BatchError$: $\sum_{i=k}^{k+b} l(w, x_i, y_i)$ - סוכמים רק את הפסד ה$batch$ הנוכחי. ומעדכנים:

$$w^{t+1} = w^t - \eta \nabla BatchError$$

כלומר, הSGD מאפשר לנו להתקרב למינימום באמצעות צעדים מהירים הרבה יותר - כי כל צעד דורש הרבה פחות חישובים וללא מעבר על כל מסד הנתונים.

4.15 הטריידאוף בין GD לSGD

נדון בשני מקרי קצה.

- $Batch = n$ (GD) כאן אנחנו משתמשים בכל הדטה בכל צעד.

יתרונות: מסלול יציב וחלק ישירות למינימום, קל לחישוב מקבילי בGPU כי זו מכפלת מטריצות ענקית.
חסרונות: איטי בטירוף. אם הדטה גדול מדי - הוא לא יכנס בזיכרון המחשב (RAM) והאלגוריתם יקרוס.

- $Batch = 1$: כאן אנחנו מעדכנים באמצעות שימוש בדגימה בודדת בכל צעד.

יתרונות: כל צעד הוא מיידי ומהיר, ה"רעש" עוזר למודל שלא להכנס לבורות אלא לקפוץ מהם החוצה.
חסרונות: המסלול שעושה האלגוריתם נראה כמו זיגזג פרוע - קשה להגיע לדיוק סופי מוחלט וזה לא מנצל היטב את הכוח של הGPU.

הפתרון: Mini Batch SGD - נקח $1 < b < n$. זו הדרך הכי טובה:
- זיכרון: נכנס בקלות לזיכרון הGPU.
- יעילות: מאפשרת לבצע כפל מטריצות מקבילי ומהיר מאוד.
- יציבות: מספקת גרדיאנט יציב מספיק כדי לא להתפרע אך משאירה מעט רעש שעוזרת ללמידה לא להתקע.

הערה 86. אז רעש זה טוב למודל? עד כה אמרנו שרעש בעייתי לנו ומהדיון כאן ניתן להבין שאנחנו צריכים להתפלל לרעש - ובכן: בתוך תהליך אופטימיזציה (חיפוש אחר מינימום) רעש הוא לא בהכרח רע לנו. ננסה לדמיין סיפור יפה להמחשה: נניח כדור שמתגלגל לתחתית של הר. אך ההר מלא בבורות קטנים לאורך הדרך. בGD רגיל - אין רעש: הכדור מתגלגל בצורה חלקה ומושלמת - אם הוא נכנס לבור קטן, שאיננו התחתית האמיתית של ההר - הוא יעצור שם. אין לו כוחות לצאת מהבור, כי השיפוע בתוך הבור אומר לו להישאר במרכז. לעומת זאת: בSGD בגלל שכל צעד מבוסס על קירוב, הכדור לא מתגלגל אלא רועד וקופץ תוך כדי תנועה: הרעידות הן הרעש. אם הכדור נתקע בבור קטן, אחת הרעידות האקראיות האלו יכולה לתת לו מספיק תנופה בכדי לצאת החוצה ולהמשיך להתגלגל מטה עד למינימום האמיתי - תחתית ההר. זה כמובן משל. מסקנה: רעש בגרדיאנט הוא טוב כי משמש בריחה ממלכודות מתמטיות בדרך לפתרון האופטימלי. אך: גם יותר מדי רעש זה לא טוב $(b = 1)$ מהסיבות שצוינו. לכן נשאל את עצמנו: איך מוצאים גודל $batch$ טוב מספיק? וזה האומנות של למידת מכונה עליה נדון בהמשך.

האם SGD באמת מתכנס לפתרון?
- GD: ההתכנסות טבעית. כיצד? ככל שאנחנו מתקרבים לתחתית הקערה - המינימום, השיפוע הופך להיות מתון יותר. לכן: הגרדיאנט שואף לאפס והצעדים הופכים להיות קטנים יותר ויותר אוטומטית. מתי עוצרים? כאשר השינוי בהפסד אפסי $(|L_t - L_{t-1}| < \varepsilon)$ או שהגרדיאנט קרוב מאוד לאפס.

- SGD: באופן תאורטי - **הוא לא באמת מתכנס!** מדוע? בגלל שאנחנו דוגמים רק חלק מהנתונים, הגרדיאנט הטוכסטי כמעט לעולם לא יהיה אפס. אם אם הגענו למינימום הגלובלי - ה$Batch$ הבא שנגריל יאמר לנו: "זוז קצת ימינה" ואחריו יגיד "זוז קצת שמאלה" וכו'. המודל ימשיך לקפץ לנצח מסביב למינימום - Variance ball.

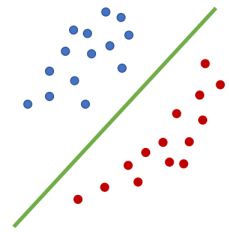
איך בכל זאת עוצרים? Learning Rate Decay

אנחנו מכריחים את הצעדים להצטמצם ע"י הקטנת הη לאורך הזמן. למשל: $\eta_t = \frac{\eta_0}{t}$: קצב הלמידה באיטרציה ה$t$ שווה לקצב ההתחלתי חלקי הזמן. ההגיון מאחורי זה הוא שבהתחלה אנחנו מרשים למודל להשתולל על מנת ללמוד את המרחב, אך ככל שהזמן עובר אנחנו מקטינים את הצעדים על מנת שייאלץ להתיישב בנקודה אחת יציבה. כלומר צריך להרגיע את המשימה לבד בכוח.

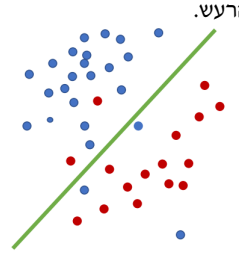
4.16 Linear separability

נתבונן במספר מקרים:

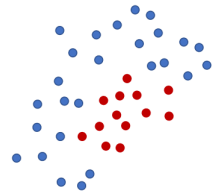
1. **מקרה ראשון:** Linearly separable data הנתונים ניתנים להפרדה לינארית. זהו המצב האידיאלי - קיים בהכרח וקטור משקולות w וסף b כך שהמודל שלנו יצדוק ב100% מהמקרים. זהו המקרה ה"ריאזיבלי". במקרה זה נוכל ונרצה להשתמש בפרספטרון.



2. **מקרה שני:** Measurement and label noise: רעש במדידה ובתיוג. כאן אנחנו למעשה מתחילים להתקרב למציאות, אך הנתונים למרות שבבסיסם נראים לינארית ישנם כמה נקודות "סוררות". כנראה - שהסיבות לכך הם טעויות אנוש בתיוג או מכשירי מדידה ישנים. הבעיה - קו ישר לא יכול להפריד בניהן בצורה מושלמת. אם ננסה בכוח להפרידם - נקבל מודל עקום מדי. במקרה זה נרצה לעבוד עם גישה הסתברותית להתמודד עם הרעש.



3. **מקרה שלישי:** נתונים לא לינאריים. זהו המקרה המורכב ביותר - הנקודות מסודרות בצורה ששום קו ישר , לא משנה היכן נשים אותו לא יצליח להפריד בניהן בצורה טובה. לכן במקרה זה המודל הלינארי לא מספיק טוב ונצטרך לעבור למודלים מורכבים יותר.



במקרה הראשון - הGS והSGD ימצאו את הקו בקלות. במקרים השני והשלישי, פונקציית ההפסד לעולם לא תגיע לאפס.

4.17 Correct Classification - הצלחה בסיווג

נגדיר כעת פורמלית מהי "הצלחה בסיווג".

מטרת הלמידה: חיפוש וקטור משקולות אופטימלי w שמתאים בצורה הטובה ביותר לנתוני האימון שלנו. האסטרטגיה היא להגדיר פונקציית הפסד שתמדוד את הטעויות שלנו, ואז למזער את הפונקציה באמצעות GD.

כיצד נראה חיזוי טוב? התנאי הוא פשוט: $y_i \cdot (w^T x_i) > 0$. נוכיח שזה עובד:

- מקרה ראשון: אכן צדקנו ו$y_i = 1$. אזי, $1 \cdot 1 > 0$. בדומה, אם צדקנו ו$y_i = -1$ נקבל $-1 \cdot -1 = 1 > 0$.
- מקרה שני: לא צדקנו. בה"כ $y_i = 1$ והחיזוי שגוי. אזי, $1 \cdot -1 < 0$, והתנאי לא מתקיים.

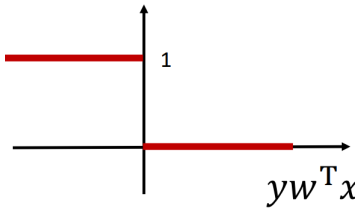
הערך $y_i(w^T x_i)$ יקרא ה$Margin$. $Margin$ חיובי מעיד על חיזוי נכון ושלילי - על חיזוי שגוי.

4.18 דוגמאות

דוגמה 87. נפתח בדוגמה ראשונה - 0-1 loss function: נגדיר את הפונקציה כך:

$$l_{0-1}(w;x,y) = \left\{ \begin{array}{cc} +1 & yw^Tx < 0 \\ 0 & yw^T > 0 \end{array} \right\}$$

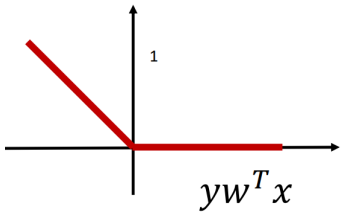
כלומר, 1 כהפסד אם טעינו בסיווג. השגיאה הכוללת הרי היא: $Err_w = \sum_{i=1}^{n} l_{0-1}(w;x_i,y_i)$
מה הבעיה? לא ניתן לחשב גרדיאנט עבור פונקציית $0-1$. מדוע? ברוב המקומות מדובר בפונקציה שהיא קו ישר אופקי: ונגזרת קו ישר אופקי הוא אפס. וכן: בעיה נוספת - בנקודה 0 יש קפיצה, לכן הפונקציה לא דפרנציאבלית. הבעיה היא שהגרדיאנט תמיד אפס: ולכן לא נבצע בכלל עדכונים. מסקנה: אי אפשר להשתמש בloss function 1-0 ישירות עם GD כי היא לא נותנת לנו כיוון להיכן להשתפר.



דוגמה 88. נגדיר את פונקציית ה$l_{hinge}(w;x,y)$:

$$l_{hinge}(w;x,y) = \left\{ \begin{array}{cc} -yw^Tx & yw^Tx < 0 \\ 0 & yw^Tx > 0 \end{array} \right\}$$

כלומר אם צדקנו ההפסד הוא אפס, ואם טעינו נשלם $-yw^Tx$. נבחין: ככל שהטעות גדולה יותר (רחוקים מהקו) הערך של ההפסד הופך לגדול יותר מתמטית. לכן ההפסד גדל. נחשב את הגרדיאנט:
כאשר צודקים - מימין: $\nabla l_{hinge} = 0$
כאשר טועים, הגרדיאנט הוא $\nabla l_{hinge} = -yx$
כעת, הפונקציה רציפה וכן כעת ניתן להשתמש בגרדיאנט למען GD.



4.19 אלגוריתם לאימון ניורון בודד באמצעות SGD וHinge loss fiction

כעת, נדון באלגוריתם המלא לאימון ניורון בודד באמצעות SGD וHinge loss function:

1. מאתחלים וקטור משקולות w רנדומלית.
2. חוזרים על הפעולות הבאות עד שהמודל מפסיק להשתפר:
3. הגרל דוגמה אחת מהדטה (x_i, y_i).
4. אם הסיווג נכון (הMargin חיובי) אזי - אל תעשה כלום.
5. אחרת, עדכן: $w = w + \eta y_i x_i$

מדוע זה עובד? נתעכב על שלב 5. אם טעינו בדוגמה חיובית: כלומר $y_i = 1$, אזי בפועל $w = w + \eta x_i$. אם טעינו בדגימה שלילית: אזי $w = w - \eta x_i$. כעת, מדוע זה עובד עם פונקציית ההפסד?

$$w_{new} = w_{old} - \eta\nabla = w_{old} - \eta(-y_i x_i) = w_{old} + \eta y_i x_i$$

שכן, $\nabla l_{hinge} = -yx$, לכן אכן האלגוריתם מתיישב עם פונקציית ההפסד.

מסקנה 89. הפרספרקטון הוא אותו אלגוריתם בדיוק כמו זה, כאשר $\eta = 1$.

4.20 רגרסיה לוגיסטית

נזכר בדוגמה השנייה שהופיעה קודם, כאשר הדטה יחסית לינארי - אך יש חריגות שמונעות מאיתנו להעביר קו ישר. עד כה עסקנו בהחלטות של סיווג לפי $\pm$. כעת נציע גישה רכה יותר שמבוססת על הסתברות. הבעיה עם פונקציית $sign$ היא שהיא בינארית לחלוטין - אם המרחק מהקו הוא 0.0000001 ו109 הוא יגיד לנו $'1'$. נרצה כי המודל יוציא מס' בין 0 ל1 שייצג את ההסתברות שהדגימה שייכת למחלקה מסויימת. למשל: $Pr(y='red'|x)$.

הגדרה 90. (פונקציית הסיגמואיד). נגדיר את הפונקצייה הבאה:

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

נעזר כעת בתכונות שלה על מנת להשיג את מטרתנו. המכפלה $w^T x$ שיכולה להיות כל מס' ב$(-\infty,\infty)$ תומר לערך ב$[0,1]$ באמצעות הפונקציה. מדוע?

$$\lim_{w^T x\to\infty}\sigma(w^T x) = \lim_{w^T x\to\infty}\frac{1}{1+e^{-w^T x}} = \lim_{w^T x\to\infty}\frac{1}{1+0} = 1$$

$$\lim_{w^T x\to-\infty}\sigma(w^T x) = \lim_{w^T x\to-\infty}\frac{1}{1+e^{-w^T x}} = \lim_{w^T x\to-\infty}\frac{1}{1+\infty} = 0$$

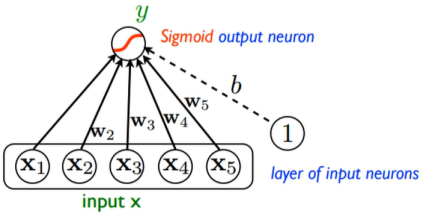
וכמו כן: אם $\sigma(w^T x=0) = \frac{1}{1+1} = \frac{1}{2}$ (חוסר ודאות מוחלט).

מדובר בפונקציה גזירה $\nabla f(x) = \frac{w\cdot e^{-w^T x}}{(1+e^{-w^T x})^2}$, ובפרט נגזרתה מקיימת: $\nabla f(x) = f(x)(1-f(x))w$.

כיצד נראה ניורון לוגיסטי? כקלט, $x\in\mathbb{R}^d$ המחוברים למשקולות $w\in\mathbb{R}^d$. המחשב כופל כל קלט במשקולתו וסוכם. $b\in\{0,1\}$ הוא האיבר החופשי. ואז - הסכום שחישבנו נכנס לפונקציית הסיגמואיד - התוצאה מהפונקציה:

$$\hat{y} = Pr[y=1|x]$$

כלומר, ההסתברות שהדגימה שייכת למחלקה 1.



מניין מקור השם? מדוע קוראים לזה 'רגרסיה' אם אנחנו מבצעים סיווג? ובכן - אנחנו מבצעים תחילה רגרסיה (חיזוי ערך רציף) על ההסתברות, ומשתמשים בתוצאה לבצע סיווג.

כעת, נשאל את עצמנו את השאלה הבאה: כיצד מודדים שגיאה כשהתוצאה היא הסתברות? כעת, נעזר בכלי שכבר השתמשנו בו קודם - Likelihood.

כיוון שהמודל מוצא הסתברות, נשאל את עצמנו: מהו הסיכוי שהמודל הנוכחי שלנו עם המשקולות w יסביר את הנתונים שראינו בפועל? נרצה למצוא את w שימקסם את ההסתברות שהתוצאות האמיתיות אכן יקרו. ובכן, ההסתברות להשתייך לדגימה חיובית $y=1$ היא פשוט $\sigma(w^T x)$ ולדגימה שלילית $y=0$ היא $1-\sigma(w^T x)$. מכאן, תחת ההנחה כי הדגימות הן iid ההסתברות לראות את כל מסד הנתונים הוא מכפלת ההסתברויות:

$$\mathcal{L}(w) = \prod_{y_i=1} Pr(y=1|x_i)\prod_{y_i=0} Pr(y=0|x_i)$$

כלומר - עוברים על כל דוגמה (מפצלים למקרים שבו $y_i=1$ או $y_i=0$ וכדומה') ומכפילים בהסתברות כי אכן המודל היה קרוב/לא קרוב לחזות. ערך זה נותן ציון מספרי אחד למודל.

כעת - יש לנו ערך שאנחנו יכולים למקסם: נרצה למקסם את $\mathcal{L}(w)$.

נמשיך לפשט את הביטוי:

$$\mathcal{L}(w) = \prod_{y_i=1} Pr(y=1|x_i) \prod_{y_i=0} Pr(y=0|x_i) = \prod_{i=1}^{n} Pr(y=1|x_i)^{y_i} \prod_{i=1}^{n} Pr(y=0|x_i)^{1-y_i} =$$

$$\prod_{i=1}^{n} \sigma(w^T x_i)^{y_i} \prod_{i=1}^{n} (1-\sigma(w^T x_i))^{1-y_i} =$$

$$\prod_{i=1}^{n} \sigma(w^T x_i)^{y_i} (1-\sigma(w^T x_i))^{1-y_i}$$

בדומה לעבר, כיוון שlog פונקצייה מונוטונית עולה, הw שממקסם את $\mathcal{L}(w)$ ימקסם את $log(\mathcal{L}(w))$. מכאן:

$$log(\mathcal{L}) = \sum_{i=1}^{n} y_i log[\sigma(w^T x_i] + (1-y_i) log[1-\sigma(w^T x_i)]$$

כעת נרצה למצוא את הנגזרת של הרגרסיה הלוגיסטית.
בתחילה, נרצה להפוך את הבעיה לבעיית אופטימיזציית מינימום שהרי לרוב אנחנו מטפלים באלגוריתמי אופטימיזציה שמוצאים מינימום - לכן נוסיף מינוס.
לפני שנתחיל בגזירה נזכר כי:

- הנגזרת של סיגמואיד:

$$\frac{d\sigma(z)}{dz} = \sigma(z)(1-\sigma(z))$$

- לפי כלל השרשרת:

$$\frac{dlow(\mathcal{L})}{dw} = \frac{dlog(\sigma(w^T x_i))}{d\sigma(w^T x_i)} \cdot \frac{d\sigma(w^T x_i)}{dw^T x_i} \cdot \frac{dw^T x_i}{dw}$$

נסמן $\sigma_i = \sigma(w^T x_i)$. מכאן עבור דגימה בודדת (x_i, y_i):

$$L_i = -[y_i ln(\sigma_i) + (1-y_i) ln(1-\sigma_i)]$$

כעת, לפי כלל השרשרת הרי שנקבל:

$$\nabla L_i = -(y_i \frac{1}{\sigma_i} \frac{d\sigma_i}{dw} + (1-y_i) \frac{1}{1-\sigma_i} \cdot \frac{d(1-\sigma_i)}{dw}]$$

$$\nabla L_i = -(y_i \frac{1}{\sigma_i} \cdot \sigma_i(1-\sigma_i) x_i + (1-y_i) \frac{1}{1-\sigma_i} \cdot (-\sigma_i(1-\sigma_i) x_i]]$$

$$\nabla L_i = -\frac{y_i}{\sigma_i} \cdot \sigma_i(1-\sigma_i) x_i - \frac{(y_i-1)(1-\sigma_i) x_i}{1-\sigma_i} = -y_i(1-\sigma_i) x_i - x_i(y_i-1)$$

$$\nabla L_i = -[y_i x_i - \sigma_i x_i] = (\sigma_i - y_i) x_i$$

לכן סה"כ קיבלנו:

$$\nabla\mathcal{L}(w) = -\sum_{i=1}^{n}[x_i(y_i - \sigma(w^T x_i)]$$

מה קיבלנו כביטוי בכל איבר בסכימה? את האיבר האמיתי y_i ובצד השני מחוסר ממנו ההערכה שלו - $\sigma(w^T x_i)$.
אם כן, כיוון שמצאנו את הגרדיאנט של הרגרסיה הלוגיסטית נוכל לעדכן את נוסחת הGD:

$$w_{t+1} = w_t - \eta\sum_{i=1}^{n} x_i[\sigma(w^T x_i) - y_i]$$

ובאופן שקול ויפה יותר:

$$w_{t+1} = w_t + \eta\sum_{i=1}^{n} x_i(y_i - \hat{y}_i)$$

לסיכום: בפרספקטרון מתקיים $\hat{y}_i = sign(w^T x_i)$ וברגרסיה לוגיסטית $\hat{y}_i = \sigma(w^T x_i)$.

4.21 מדוע רגרסיה לוגיסטית נחשבת למסווג לינארי?

הרי, אנחנו משתמשים בפונקציה מעריכית כסיגמואיד, למה רגרסיה לוגיסטית נחשבת כמסווג לינארי?
• כלל ההחלטה: אנחנו נחליט כי דגימה שייכת למחלקה 1 אם הסיכוי שהיא ב1 גדול מהסיכוי שהיא ב0:

$$Pr[y = 1|x] > Pr[y = 0|x]$$

זה קורה בדיוק כאשר הסיגמואיד גדול מחצי.
• וכן כלל ההחלטה הנ"ל, שקול ל:

$$\frac{Pr[y = 1|x]}{Pr[y = 0|x]} > 1 \iff \frac{\sigma(w^T x)}{1 - \sigma(w^T x)} = e^{w^T x} > 1$$

ערך זה $(e^{w^T x})$ נקרא ה$odds$.
• אם נפעיל ln על המשוואה הקודמת, נקבל:

$$ln(\frac{\sigma(w^T x)}{1 - \sigma(w^T x)}) = w^T x > 0$$

וכן $w^T x$ היא פונקציה לינארית לחלוטין של x.
• בשורה התחתונה - למרות שהמודל מוציא הסתברויות עגולות ורכות: קו ההפרדה שבו המודל עובר מלהמר על 0 ללהמר על 1 הוא קו ישר.
אם $w^T x > 0$ הוא מהמר על 1, אחרת הוא מהמר על אפס.
זה בדיוק כמו בפרספטרון - רק שההבדל הוא שבפרספטרון הקפיצה היא מיידית, וכאן המעבר ליד הקו הדרגתי.

5 הרצאה 5

6 הרצאה 6

7 הרצאה 7

8 הרצאה 8

9 הרצאה 9

10 הרצאה 10

11 הרצאה 11