מבוא למערכות לומדות - 67577

28 ביוני 2015

מרצה: עמית דניאלי

איני לוקחת אחריות על מה שכתוב כאן, so tread lightly
אין המרצה קשור לסיכום זה בשום דרך.
הערות יתקבלו בברכה - *noga.rotman@gmail.com*. אהבתם? יש
עוד!http://www.bit.ly/integrali

תוכן עניינים

1 על הקורס 5
1.1 הקדמה 5
1.1.1 סקירה היסטורית קצרצרה 5
1.1.2 מה עושים עם למידה חישובית? 5
1.2 מנהלות 5
1.2.1 מבנה הקורס 5

I מודל PAC ותורת ההכללה 7

2 לימוד מנתונים 8
3 מחלקת היפותזות 10
4 מודל Probably Approximately Correct (PAC) 11
5 אלגוריתם ה־ERM (Empirical Risk Minimizer) וסיבוכיות המדגם של מחלקות סופיות . . . 12
5.1 הגדרות 12
5.2 סיבוכיות המדגם של מחלקות סופיות 15
5.2.1 אפליקצה של המשפט - סיבוכיות מדגם של מחלקות בנות d־פרמטרים . . . 16
6 מימד VC והמשפט היסודי 16
6.1 מימד VC 16
6.2 הקדמה להוכחת המשפט היסודי 17
6.2.1 המשפט היסודי 18
6.3 למת המדגם הכפול 19
6.4 למת סאור־שלח 21
6.5 הוכחת המשפט היסודי כתוצאה משתי הלמות 23
7 סיכום ביניים מס' 1 - תורת ההכללה 24
7.1 אז מה בעצם למדנו עד כה? 24
7.1.1 סכמה לשימוש בלמידה 24
7.1.2 שגיאת קירוב ושגיאת הכללה - יחסי גומלין 24
7.2 מה הלאה? השלב הבא בקורס: אופטימיזציה 25
7.3 כלים טכניים ומתמטיים שראינו 25

II אלגוריתמי למידה 26

8 אופטימיזציה 26
8.1 קמירות על קצה המזלג 26
8.1.1 קבוצות קמורות 26
8.1.2 פונקציות קמורות 26
8.2 אופטימיזציה קמורה וגרדיינט דסנט (Gradient Descent) 28
8.2.1 תכונות של פונקציות קמורות 28
8.3 בעיות למידה קמורות 29
8.4 אלגוריתמים יציבים ולמידות 30
8.5 למידות של בעיות קמורות 32
8.5.1 מדוע יש לדרוש חסימות וליפשיציות? 32
8.6 למידה באמצעות Regularized Loss Minimization 32
8.6.1 הוכחת למה 8.22 34
8.7 מעבר לבעיות קמורות - ההופעה של הקושי החישובי 36
8.8 אלגוריתם ה־SVM 36
8.8.1 הצגה כ־RLM וניתוח סיבוכיות מדגם וריצה 38
8.8.2 תחליפים קמורים (Convex Surrogates) - מעבר ל־SVM 39
8.9 שיכונים במרחבים ממימד גבוה ושיטות גרעין 40
8.9.1 שיכונים במימד גבוה 40
8.9.2 שיטות גרעין 41
8.9.3 דוגמאות לגרעינים 42
8.10 ולידציה ובחירת מודל 43
9 אלגוריתמים יוריסטיים 44

9.1 אלגוריתם השכן הקרוב . . . 44
9.2 רשתות נוירונים . . . 45
9.2.1 סיבוכיות המדגם . . . 46
9.2.2 אילו פונקציות ניתן לחשב בעזרת רשת נוירונים? . . . 47
9.2.3 כיצד לאמן רשת נוירונים . . . 47
9.2.4 טרנדים עכשווים ברשתות נוירונים . . . 49
9.2.5 סיכום . . . 49
9.3 עצי החלטה . . . 49
9.3.1 סיבוכיות חישובית . . . 50
10 Boosting . . . 51
10.1 לומדים חלשים . . . 51
10.2 יערות אקראיים . . . 52
10.3 האצה אדפטיבית (AdaBoost) . . . 52
10.3.1 השגיאה האמפירית . . . 53
10.3.2 שגיאת ההכללה . . . 55
10.3.3 מלמידות חלשה לחזרה - היסטוריה והשלכות תיאורטיות של AdaBoost . . . 56
10.3.4 אפליקציה - זיהוי פנים (*Viola and Jones*) . . . 57
10.4 דלילות ובחירת פיצ'רים על קצה המזלג . . . 57
10.4.1 שיטות נוספות לבחירת פיצ'רים \ דלילות . . . 57
11 סיכום ביניים מס' 2 - תורת ההכללה ואלגוריתמי למידה . . . 57

III למידת ייצוג 61

12 הצברה - אלגוריתם ה־k-means . . . 62
12.0.2 הערות והדגמות . . . 63
12.1 אלגוריתמים נוספים ופונקציות מטרה נוספות . . . 64
12.2 הצברה מעבר ללמידת יצוג - מציאת חלוקה בעלת משמעות . . . 64
13 הורדת מימד . . . 64
13.1 ניתוח גורמים ראשיים (PCA) . . . 65
13.1.1 הערות . . . 66
13.2 הטלות מקריות . . . 67
14 הצצה ללמידת מילון . . . 68

IV נושאים נוספים 70

15 למידת אונליין . . . 70
15.1 Online classification . . . 70
15.1.1 במקרה ה־realizable . . . 71
15.1.2 במקרה הכללי (אגנוסטי) . . . 72
15.2 הוספת רנדומיות למודל . . . 73
15.2.1 Multiplicative Weights (*Hedge*, *Weighted Majority*) . . . 73
15.3 הקשר בין Online ל־Batch . . . 75

רשימת אלגוריתמים

1 Gradient Descent . . . 28
2 Hard-SVM . . . 37
3 (Soft-)SVM . . . 37
4 SVM with a mapping Ψ . . . 40
5 Model Choice . . . 44
6 Nearest Neighbor . . . 45
7 אלגוריתם בסיסי ללמידה של רשתות נוירונים . . . 48
8 אלגוריתם חמדן בסיסי ללמידת עצי החלטה . . . 50
9 אלגוריתם היערות האקראיים . . . 52
10 AdaBoost . . . 53

63 . . . . . k-means 11
64 . . . . . k-means++ initialization 12
65 . . . . . PCA 13
71 . . . . . Consistent 14
72 . . . . . Halving 15
72 . . . . . Follow the Leader 16
74 . . . . . Multiplicative Weights 17

1 על הקורס

1.1 הקדמה

הנושא של הקורס נקרא "למידה חישובית". בדקות הראשונות של השיעור ננסה להבין אינטואיטיבית ולא פורמלית מה זה, למה זה חשוב, וכו'.

מה זה למידה חישובית?

הגדרה אפשרית - התחום העוסק בפיתוח אלגוריתמים באמצעות תהליך שאנחנו קוראים לו למידה.

למידה לענייננו - התהליך בו הופכים נתונים לידע, ומתשמשים בידע הזה כדי לבצע פעולות בצורה טובה יותר.

כדי להבהיר ולהבין למה זה טוב, נביט בדוגמא (שקופית 3?). נביט בבעיית הסיווג - בהנתן תמונה צריך להחזיר תשובה האם יש בתמונה כסא או לא. אנחנו יודעים להבין בין אובייקטים שהם כסאות ולא כסאות.

גישה ראשונה - אפשר לרשום אוסף של תכונות, לחפש אותן בתמונה, ועל סמך הבדיקות האלה להכריע האם יש כסא או אין.

הבעיה: יש הרבה מאוד סוגים של כסאות. איך אפשר לסווג את כל התכונות? יהיה קשה לנו מאוד לכתוב רשימה כזו של תכונות.

בשנות השישים והשבעים זו היתה הגישה לביצוע משימות מהסוג הזה. היא נכשלה.

אולי זה לא כלכך מפתיע שהגישה הזו נכשלה; אנחנו יודעים להפריד בין כסאות ואובייקטים אחרים, אבל אף אחד לא נתן לנו רשימה של תכונות. כנראה שהראו לנו כסאות, ועם הזמן הבנו להבדיל בין כסא ללא כיסא.

הרעיון: להשתמש בפרדיגמה הזו גם על מחשבים - להסביר למחשב כיצד ללמוד.

1.1.1 סקירה היסטורית קצרצרה

תחום שקיים משנות השישים, פורח בשנות האחרונות. מדוע? אלגוריתמי למידה דורשים משאבים שהיו פחות זמינים בעבר. כמו כן, זו הגישה היחידה שעובדת.

1.1.2 מה עושים עם למידה חישובית?

- זיהוי פנים (היום כמעט אך ורק באמצעות למידה חישובית)
- זיהוי קול
- זיהוי טקסט
- דיאגנוזה של מחלות
- ביולוגיה חישובית
- ועוד..

1.2 מנהלות

תרגילים - 20% מהציון, חלקם תכנותיים, אבל לא כבדים (עשר או 15 שורות קוד).

מבחן: 80%.

מבחינת ספרות, יש הרבה מאוד ספרים על למידה חישובית. באתר יש קובץ עם רשימה. הספר שיהיה הכי קרוב הוא הספר של שי של־שוורץ ושי בן דוד.

כמו כן, המרצה יפרסם סיכומים של ההרצאות.

שתי הערות נוספות:

- יש לנו היום שלוש שעות. המרצה מציע שעה פלוס מינוס, הפסקה, ואז עוד שעה פלוס מינוס. נראה שיש הסכמה בכיתה. מסיימים קרוב לשתיים וחצי.
- לא מפורט הסילבוס במצגת, נעבור על זה בסוף ההרצאה.

1.2.1 מבנה הקורס

באופן קסום, הרי סוף ההרצאה לפני שבכלל התחלנו:

1. שבועות $1-5$: תורת ההכללה. זהו חלק שמבחינה מתמטית יהיה יחסית כבד.
2. שבועות $6-10$: אלגוריתמי למידה. בכל שבוע נעבור על מחלקת היפוטזה אחרת, ונראה אלגוריתמי למידה שמשתמשים בהן (או לפחות מנסים).

3. שבועות $11 - 12$: ייצוג נתונים.
4. שבועות $13 - 14$: נושאים נוספים, TBD.

חלק I
מודל PAC ותורת ההכללה

רוב רובה של ההרצאה היום ננסה להבין איך לנסח בעיות בלמידה חישובית. זהו נושא לא טריוויאלי בכלל.
המשימה הכי בסיסית - ללמוד מיפוי, אותו נסמן ב־$h*$. הוא ממפה לנו קלטים ממרחב X לפלטים במרחב Y, או:

$$h^* : X \to Y$$

נכתוב כמה דוגמאות מייצגות, שיהיו שונות ממה שאנחנו רגילים אליהם בדוגמאות הקלאסיות שלי מדעי המחשב.

זיהוי אובייקטים בתמונות נגיד שכל התמונות שלנו הן 100×100, התמונות הן בשחור לבן, הגוונים נעים בין 0 ל־1 (שחור ולבן)
אזי:

$$X = M_{100\times100}([0,1])$$
$$Y = \{\text{"dog"}, \text{"cat"}, \text{"plane"}, \text{"none"}\}$$

המיפוי ממפה כל תמונה ללייבל המתאים לה.

זיהוי ספאם ננסה ללמוד קובץ טקסט, וצריך להבדיל בין הודעות ספאם לבין הודעות "שהן בסדר":
צריך להחליט איך נייצג את הטקסטים שלנו. נבחר את כל המחרוזות בגודל קטן ממיליון.

$$X = \bigcup_{n=0}^{10^6} \{+=1\}^n$$
$$Y = \{\text{"spam"}, \text{"ok"}\}$$

וכמובן והמיפוי מעביר כל טקסט ללייבל המתאים לו.

התאמת מקטעי שמע לטקסט נרצה להתאים בין קבצי שמע לבין קבצי טקסט המכילים את הטקסט שנשמע בקובץ השמע.

תרגום אינטואיטיבית נראה שבמקרה זה למידה היא לא אחד מהכלים הראשונים שהיינו מפעילים - למרבה ההפתעה, היא כן.

התאמת רצפי DNA לתכונות צבע עיניים, נטייה למחלות,...

אבחון מחלות ע"ס נתונים רפואיים נתונים רפואיים - למשל סימפטומים

חיזוי מזג אוויר התאמת נתונים מטרולוגיים למזג האוויר שיהיה מחר

חיזוי ביצועי מניות ע"ס נתונים כלכליים
בכל מה שרשמנו מעלה, השיטה הטובה ביותר היא למידה חישובית, או שיטות המערבות בצורה בולטת למידה חישובית.

כמו שניתן לראות מהדוגמאות, יהיו לנו הרבה סוגים שונים של קלטים. נרצה להתייחס לכולם באיזה שהוא אופן קבוע, כדי שלא נצטרך להמציא כל הזמן את הגלגל מחדש.
משהו שאנחנו רגילים לעשות ממדעי המחשב:

המרחב X יכיל ייצוגים של הקלטים. בחירת הייצוג חשוב מאוד, ונעסוק בנושא זה בהמשך.
כעת - נדבר באופן לא פורמלי, בהמשך נגדיר את הכל בצורה מסודרת.
מרחב זה יהיה מרחב של וקטורים $X = A^n$ כאשר למשל:

$$A = \{\pm 1\}\,,\ A = \{1, ..., k\}\,,\ A = [0, 1]\,,\ A = \mathbb{R}, ...$$

כאשר באופן כללי n "יהיה גדול אבל לא ענק": $50 \leq n \leq 5 \cdot 10^6$.
נרצה שהאלגוריתם שלנו יהיה פולינימיאלי ב־n; ניקח את n בתור הפרמטר בסיבוכיות. מה זה אומר? נרצה לא רק שהתוכנה תעבוד בזמן פולינומיאלי ב־n, אלא נרצה שגם התוכנה שלומדת וגם מספר הדוגמאות יהיו תחומים ב־n.
מספר הקלטים השונים שלנו הוא לפחות אקספוננציונלי ב־n; דהיינו, אין לנו שום דרך לעבור על כל הקלטים ב־X. כלומר $|X| \geq 2^n$.

המרחב Y יכיל תיוגים אפשריים לקלטים - למשל, לשאלה "מה מופיע בתמונה", אפשר לענות "כיסא", "ברווז",.... במסגרת הקורס, נניח ש־Y הוא "די פשוט": $Y = \{1, ..., k\}$ או $Y = \mathbb{R}$.

2 לימוד מנתונים

לפעמים קשה מאוד לרשום קוד יעיל שמחשב את המיפוי; לעיתים לא קיים, אבל עם מצב זה לא נתמודד במסגרת הקורס. לפעמים הסיבה לקושי היא אחרת - זה לא שאין אלגוריתם יעיל שעושה את זה, אבל הקוד הוא מאוד מסובך, או גרוע מזה, אנחנו לא יודעים מהו. יכול להיות שהתיאור של האלגוריתם מאוד מורכב. יותר מזה, יכול להיות שאנחנו לא באמת יודעים מהו, אין לנו דרך לתאר אותו.
השיטה של למידה חישובית היא סיבה עקיפה לפתרון - על סמך מדגם המחשב יכתוב את האלגוריתם.

נקבל מדגם של דוגמאות x_i, כאשר עבור כל דוגמא, נתון לנו התיוג הנכון $h^*(x_i)$:

$$(x_1, h^*(x_1)), ..., (x_m, h^*(x_m))$$

כלומר לא באמת נקבל פונקציה, אלא יותר נכון נקבל תיאור של אלגוריתם, בתקווה יעיל, המחשב את הפונקציה הזו.

הגדרה 2.1 אלגוריתם למידה הוא אלגוריתם $\mathcal{A}$ המקבל בתור קלט מדגם $(x_1, y_1), ..., (x_m, y_m) \in X \times Y$, ומחזיר בתור פלט פונקציה $h : X \to Y$.

הערה 2.2 כאמור, מעלה מופיע שקר כלשהוא, כי זה לא באמת פונקציה (אי אפשר לייצג פונקציה ממרחב כל כך גדול בצורה "יעילה").

יש הרבה מאוד דרכים לאסוף תמונות ולתייג כל אחת מהן. אפשר לעשות חיפוש מושכל, אפשר להסתכל על סוגי התמונות, ועוד ועוד.

- איך מיוצר המדגם? (גם נשאל, איך אנחנו רוצים למדל מבחינה מתמטית את הדרך בה מיוצר המדגם?)
- איך נבחן את ביצועי $\mathcal{A}$?

במדעי המחשב בדרך כלל השאלה השניה מאוד ברורה - זמן ריצה אפשר למדוד, וכמובן נדרוש יעילות, אבל לא ברור איך אפשר להעריך את הפלט שאנחנו מקבלים. זאת, מכיוון וכמעט תמיד האלגוריתם לא יחזיר את h^* "האמיתית", ונצטרך לכמת את איכות הפונקציה שאנחנו מקבלים.
דרישה ראשונה: נרצה ש־h^* שנקבל, תיתן תשובה נכונה על הדוגמאות שנריץ עליהם. צריך לדאוג שהדרך שבה המדגם מיוצר יהיה דומה לדרך שבה נקבל את הדוגמאות בפועל.
על כן, נניח שהמדגם והאובייקטים שנראה בפועל ייוצרו באותה דרך. איך נעשה זאת? תהיה לנו התפלגות על X, נתאמן על דוגמאות שנדגמו באופן אקראי ע"י ההתפלגות הזו על X, ואת הדוגמאות ה"אמיתיות" נדגום מהמרחב באותה התפלגות.

הגדרה 2.3 (זמני) **התפלגות מייצרת נתונים** היא התפלגות $\mathcal{D}$ על X.

אם בוקטורים עסקינן, נרצה גם להגדיר מרחק:

הגדרה 2.4 פונקציית הפסד היא פונקציה $l : Y \times Y \to \mathbb{R}^+$, המקיימת:

$$l(y, y) = 0 \ \forall y \in Y$$

כעת, אנחנו מוכנים להגדיר את המדד להצלחת h:

הגדרה 2.5 ההפסד (או השגיאה) של $h : X \to Y$ הוא:

$$L_{\mathcal{D}}(h) := \mathbb{E}_{x \sim \mathcal{D}} l(h(x), h^*(x))$$

כאשר $h(x)$ היא התחזית, x נדגם לפי $\mathcal{D}$.

הערה 2.6 למה לא הגדרנו מטריקה? לפעמים זו לא מטריקה, ולא נרוויח שום דבר מהוספת מגבלות.

שתי דוגמאות לפונקציות הפסד מאוד פופולריות:

1. ההפסד $(0,1)$: זוג פלטים יקבל הפסד 0 רק אם הם זהים. זוהי פונקציית הפסד מאוד שימושית.

$$l_{0-1}(\hat{y}, y) = \begin{cases} 0 & \hat{y} = y \\ 1 & \hat{y} \neq y \end{cases}$$

2. square loss: לפעמים נרצה פונקציית הפסד יותר רציפה (ושהיא איננה מטריקה!):

$$Y = \mathbb{R}$$
$$l(\hat{y}, y) = (y - \hat{y})^2$$

שתי הדוגמאות האלו מכסות אחוז מאוד גבוה של הפונקציות שמשתמשים בהם (או וריאנט שלהם).

הערה 2.7 לפעמים, למשל בספר של שי, נפגשים עם הגדרה יותר כללית לפונקצית ההפסד:

$$l : Y^X \times X \times Y \Rightarrow l(h, x, y)$$
$$\Rightarrow L_{\mathcal{D}}(h) = \mathbb{E}_{x \sim \mathcal{D}} l(h, x, h^*(x))$$

ההגדרה שלנו היא מקרה פרטי של ההגדרה הזו.

נכליל קצת את ההגדרות שהבאנו מעלה.
עד עכשיו הנחנו שיש לנו פונקציית מטרה h^* המחזירה פלט המוגדרת היטב לכל קלט.
אולם, לא זה תמיד המקרה - למשל, הנתונים המטרולוגיים יתנו לנו אינדיקציה אך לא ודאית \ מלאה. מקרים כאלו קורים לא מעט, ונרצה למדל גם סיטואציות כאלה.
לא נניח ש־y עבור x מסויים לא נקבע באופן יחיד.
אזי, להגדרה הלא זמנית:

הגדרה 2.8 התפלגות מייצרת נתונים היא התפלגות $\mathcal{D}$ על $X \times Y$.

נרחיב את ההגדרה של ההפסד:

הגדרה 2.9 בהתייחס להגדרה מעלה:

$$L_{\mathcal{D}}(h) = \mathbb{E}_{(x,y) \sim \mathcal{D}} l(h(x), y)$$

הערה 2.10 נשים לב שבמקרה החדש יש לנו התפלגות, ולא פלט יחיד, ואין לנו יותר את h^*, אלא פונקציה הסתברותית.

המקרה שדיברנו עליו עד עכשיו מתקבל כמקרה פרטי: אם $\mathcal{D}'$ היא התפלגות על X ו־$h^* : X \to Y$, נביט בהתפלגות של $(x, h^*(x))$ כאשר $x \sim \mathcal{D}'$.
במקרה זה, נאמר ש־$\mathcal{D}$ פרידה (ניתנת למימוש, realizable) ע"י h^*.

הערה 2.11 אנחנו נניח לאורך הקורס ש־y הוא תיוג נכון (יש מקרים שזה לא נכון, אבל לא נדון בהם כאן). כמו כן, יכול להיות שיהיו להן דוגמאות בהן לאותו x נקבל פעם אחת y_1 ופעם שניה y_2, אבל בכל זאת נניח ה־y שיתקבל הוא נכון. זאת מהסיבה שדיברנו עליה קודם לכן - אין לנו דרך לקבל קלט מלא, כלומר מספיק אינפורמציה כדי לקבל באמת y יחיד, ולא מספר y־אים שונים אך נכונים בהנתן המידע שקיבלנו.

עובדה ראשונה בלמידה חישובית:

אין ארוחות חינם

"החיים לא כל כך פשוטים, ולא נוכל ללהשיג בקלות את כל מה שאנחנו רוצים".

נניח לשם פשטות כי $X = \{\pm 1\}^n$, $Y = \{\pm 1\}$, וההפסד: $l = l_{0-1}$.
היינו רוצים אידאלית אלגוריתם למידה יעיל, כלומר רץ בזמן פולינומיאלי ב־n על מספר דוגמאות פולינומיאלי ב־n, ומחזיר היפוטזה בזמן "סביר" שהטעות שלה "טובה".

הגדרה 2.12 אלגוריתם למידה $\mathcal{A}$ יקרא "מושלם" אם קיים $c > 0$ כך שלכל התפלגות $\mathcal{D}$ על x ולכל $h^* : X \to Y$, הפלט של $\mathcal{A}$ על מדגם בן n^c דוגמאות מקיים $L_{\mathcal{D}}(h) < 0.3$.

משפט 2.13 לא קיים אלגוריתם מושלם.

יש למשפט זה הרבה יותר גרסאות. למשל, ללא הנחות התוצאה תהיה גרועה מאוד.

"רעיון ההוכחה":

מאוד פשוט: אם אנחנו לא יודעים שום דבר א־פריורית איך h^* נראית או שום דבר על $\mathcal{D}$, אין סיבה להניח שנדע מה הפלט הנכון עבור דוגמאות שלא ראינו, ולכן נצדק פחות או יותר רק על מה שראינו. מכיוון ומספר הפלטים שעברנו עליהם מאוד קטן ביחס למרחב כולו, כלומר $n^c << 2^n$, נטעה על רב הדוגמאות.

המסקנה המיידית מההתחלה הזו - אם לא מניחים כלום על כלום, אין סיכוי שנצליח לבנות אלגוריתמים. לכן, כדי לקבל אלגוריתמים עלינו להניח הנחות מסויימות "על משהו". זה מביא אותנו לנושא הבא - מחלקת היפוטזות. אנחנו נלמד את מודל פק, אותו נגדיר היום.

3 מחלקת היפותזות

נפתח לנו מגרש משחקים מאוד גדול כאן, שכן הנחות לא חסר. אבל אנחנו נרצה הנחות שגם יאפשרו למידה, וגם יתקיימו ב"רב הפעמים".
ההנחה שנדבר עליה אומרת - הפונקציה שאנחנו מחפשים היא פונקציה פשוטה, ששייכת לאוסף קטן של פונקציות פשוטות. במקרה הזה באמת נוכל למצוא פונקציה טובה מספיק, כי לא נצטרך לעבור על "יותר מידי" אופציות.

הגדרה 3.1 מחלקת היפוטזות $\mathcal{H}$ היא אוסף של פונקציות מ־X ל־Y.

"הנחת PAC" (במרכאות כי לא מדוייק): קיים $h \in \mathcal{H}$ עם $L_{\mathcal{D}}(h)$ קטן.

דוגמאות:

1. פונקציונליים אפינים. זוהי מחלקה מאוד מאוד בסיסית, רלוונטית כאשר $X \subset \mathbb{R}^n$, $Y = \mathbb{R}$, ו־$\mathcal{H}$ מכילה את כל הפונקציות מהצורה:

$$x \in \mathbb{R}^n,\ h(x) = a_1 x_1 + ... + a_n x_n + b$$

הרבה פעמים תחת ההנחה שההיפותזה שלנו נמצאת במחלקה הזו "או קרובה מספיק", למידה תהיה אפשרית. נוכיח זאת בהמשך הקורס.

2. עוד מחלקה מאוד בסיסית - חצאי מרחבים. במקרה זה $X \subset \mathbb{R}^n$, $Y = \{\pm 1\}$, ו-$\mathcal{H}$ מכילה את כל הפונקציות מהצורה:

$$h(x) = sign\left(\left(\sum_i a_i x_i\right) + b\right)$$

כאשר $sign(t) = \begin{cases} 1 & t > 0 \\ -1 & t \leq 0 \end{cases}$.

3. עצי החלטה בגודל לכל היותר B - $Y = \{\pm 1\}^n$, $X = \{\pm 1\}$, B מספר טבעי. $\mathcal{H}$ מכיל את כל הפונקציות שניתן לממש ע"י עץ החלטה בגודל $\leq B$.

בתרגול ובמהלך הקורס נראה מקרים פרטיים של הדוגמאות הללו.

4 מודל $Probably\ Approximately\ Correct\ (PAC)$

הומצא לפני 40 שנה פחות או יותר. אומר: האלגוריתם יקבל מדגם, בהסתברות גבוהה (probably) נקבל שגיאה נמוכה, כלומר נכונה בקירוב (approximately correct).
ובצורה פורמלית:

הגדרה 4.1 בעיית למידה (בעיית PAC) היא רביעייה $(X, Y, l, \mathcal{H})$.

הגדרה 4.2 בהנתן התפלגות $\mathcal{D}$ על $X \times Y$, נגדיר:

$$L_{\mathcal{D}}(\mathcal{H}) := \underset{h \in \mathcal{H}}{inf} L_{\mathcal{D}}(h)$$

כלומר, השגיאה היא של ההיפותזה הכי טובה.

אנחנו נרצה למצוא היפותזה שהשגיאה שלה "מתקרבת מאוד" לשגיאה הזו.
נרצה מדד כדי לקבוע כמה דוגמאות אלגוריתם צריך כדי להחזיר היפותזה בעלת טעות שכזו. זו הגדרה עדינה בעלת לא מעט פרמטרים. נמשיך אם כך בזהירות:

הגדרה 4.3 יהא $\mathcal{A}$ אלגוריתם למידה. **סיבוכיות המדגם של** $\mathcal{A}$ עם פרמטר שגיאה $\varepsilon > 0$ ופרמטר ודאות $\delta > 0$ היא המספר המינימלי $m_{\mathcal{A}}(\varepsilon, \delta)$ עבורו לכל התפלגות $\mathcal{D}$ על $X \times Y$ אם:

$$S = \{(x_1, y_1), ..., (x_m, y_m)\} \sim \mathcal{D}^m$$

עבור $m \geq m_{\mathcal{A}}(\varepsilon, \delta)$ (כלומר הוא מדגם מספיק גדול שנדגם לפי $\mathcal{D}$), אז:

$$Pr_{S \sim \mathcal{D}^m}\left[L_{\mathcal{D}}(\mathcal{A}(S)) \geq L_{\mathcal{D}}(\mathcal{H}) + \varepsilon\right] \leq \delta$$

הסבר: מהו המספר $m_A(\varepsilon, \delta)$? הוא מספר הדוגמאות שאנחנו צריכים לראות כדי להבטיח את התנאי שאנחנו רוצים: שההסתברות שהשגיאה של ההיפותזה שהפיק האלגוריתם בהנתן המדגם תהיה גדולה מהשגיאה הכי טובה במחלקה ועוד מספר קטן כלשהוא (ε) היא קטנה (δ), כאשר S הוא מדגם אקראי, $\mathcal{A}(S)$ היא ההיפותזה המתקבלת כשהרצנו את המדגם הזה.

השימוש במחלקה מאפשר למידה (לפי המשפט שראינו). כיצד בוחרים מחלקה בצורה נכונה? יש שני אילוצים סותרים: צריך לקחת מחלקה קטנה מספיק, ומצד שני שתוכל להביע את הבעיה שלנו. יש עוד דרישה - שיהיה אפשר ללמוד את מחלקת ההיפוטזות בצורה יעילה, אבל על זה לא נדבר כרגע.
הרבה פעמים בחירת המחלקה לא תהיה טובה. למה? כל מיני סיבות: יכול להיות שבמחלקה אין היפוטזה טובה. יכול להיות שהמחלקה תהיה מאוד גדולה, ולכן למרות שיש בה היפוטזה טובה, לא נצליח למצוא היפוטזה מספיק טובה. יכולה להיות בעיה נוספת שכרגע לא נתייחס אליה - גם אם קיימת היפותזה טובה וקיים אלגוריתם שיכול להחזיר היפותזה טובה, זה פשוט יהיה קשה מידי חישובית להחזיר היפותזה כזו.

האם ההיפותזה שנחזיר היא בהכרח מהמחלקה הזו? **לא**! זאת, למרות ההנחה שאנחנו דורשים - שיש היפותזה טובה במחלקה.
אנחנו נתייחס למקרים בהם האלגוריתם דטרמיניסטי.

איך נדע איזו $\mathcal{H}$ לבחור? א־פריורית יש כאן הרבה מאוד מידע. כמו כן, אם אנחנו יודעים את זה, למה שלא נחזיר פונקציה טובה?
בוחרים את $\mathcal{H}$ על סמך תחושת בטן או נסיון, ובאמת זה לא תמיד עובד. אבל אם לא יודעים את השגיאה של המחלקה, איך בודקים את הטיב של ההגדרה שמופיעה מעלה? לא ברור, ולא תמיד אפשר. נראה שבמקרים מסויימים אפשר להבטיח שזה מתקיים.
בקורס נראה חמש־שש משפחות של מחלקות היפוטזות. בפועל כשנרצה להשתמש בלמידה, נצטרך לעבור על הרשימה הזו ועל מחלקות נוספות, ולהבין באיזו מחלקה כדאי להשתמש לבעיה שלנו.

הגדרה 4.4 נאמר שה**בעיה למידה** אם קיים אלגוריתם למידה $\mathcal{A}$ עם סיבוכיות מדגם סופית, כלומר:

$$\forall \varepsilon, \delta > 0,\ m_{\mathcal{A}}(\varepsilon, \delta) < \infty$$

נאמר שהבעיה **למידה ביעילות** אם קיים אלגוריתם למידה יעיל $\mathcal{A}$ (פולינומיאלי בקלט שלו), וכן:

1. $m_{\mathcal{A}}(\varepsilon, \delta)$ פולינומיאלית ב־$n, \frac{1}{\varepsilon}, \frac{1}{\delta}$.
2. הפלט של $\mathcal{A}$ יהיה יעיל (פולינומיאלי ב־n).

מה המשמעות של "פלט יעיל"? כאן הפלט הוא פונקציה. נרצה שהחישוב של הפלט של הפונקציה יהיה יעיל.
נשים לב כי אם הבעיה למידה ביעילות, בפרט $\mathcal{A}$ יעיל ב־$n, \frac{1}{\varepsilon}, \frac{1}{\delta}$.

5 אלגוריתם ה־ERM (Empirical Risk Minimizer) וסיבוכיות המדגם של מחלקות סופיות

5.1 הגדרות

בשעה טובה, הגענו לאלגוריתם הראשון שלנו!
הוא אלגוריתם מאוד כללי, יהיה לנו כזה לכל מחלקה, כמעט תמיד סיבוכיות המדגם שלו תהיה זו הכי טובה שנוכל לקבל. מצד שני, הוא לא יעבוד תמיד, כי בדרך כלל הוא לא יהיה יעיל.
ננסה להגיע אליו ביחד. נניח כי נתונה בעיית למידה $(X, Y, l, \mathcal{H})$, ומדגם:

$$(x_1, y_1), ..., (x_m, y_m)$$

אם היה לנו כח חישוב בלתי מוגבל באופן נאיבי, מה היינו עושים?
היינו עוברים על כל ההיפותזות, ובוחרים את הכי טובה - זו שנותנת $L_{\mathcal{D}}(h)$. נניח שיש מספר סופי של היפותזות. מה הבעיה באלגוריתם שתיארנו?
אחת, יכולות להיות כמה היפותזות שונות עם שגיאה אופטימלית. במקרה זה פשוט נבחר אחת מהן.
הבעיה האמיתית - אי אפשר לחשב את $L_{\mathcal{D}}$ לכל h: כח החישוב בלתי מוגבל, אך מספר הדוגמאות מוגבל. מה נעשה? נשערך ע"י ממוצע על המדגם שלנו.

הגדרה 5.1 השגיאה האמפירית, או **שגיאת האימון** של היפותזה $h : X \to Y$ ביחס למדגם:

$$S = \{(x_1, y_1), ..., (x_m, y_m)\}$$

היא:

$$L_S(h) := \frac{1}{m} \sum_{i=1}^{m} l(h(x_i), y_i)$$

עובדה פשוטה:

$$\mathbb{E}_S\left[L_S\left(h\right)\right] = L_{\mathcal{D}}\left(h\right)$$

ומכאן, נוכל להגדיר את האלגוריתם האמיתי:

הגדרה 5.2 אלגוריתם למידה יקרא **ERM** אם מתקיים לכל מדגם S:

$$L_S\left(\mathcal{A}\left(S\right)\right) = \underset{h\in\mathcal{H}}{inf} L_S\left(h\right)$$

ובנוסף $\mathcal{H} \ni \mathcal{A}\left(S\right)$.

כלומר, אלגוריתם ERM - empiircal risk minimizer מוגדר להיות זה הממזער את השגיאה האמפירית. how appropriate.
בשבוע הבא נתחיל מניתוח אלגוריתמי ERM.
היום נמשיך לדבר על מה שהתחלנו לדבר עליו בשיעור הקודם ־ תורת הלמידה, ובפרט ־ תורת ההכללה.
בשיעור הזה והבא נלמד חלק קטן מתורת ההכללה שיעזור לנו לחשב עבור בעיות קלסיפיקציה את ההפסד של ההיפותזות המתקבלות. זהו לא סוג הבעיות היחיד שקיים, אך נדון בהם בחלק הראשון + של הקורס.
תחילה ־ תזכורות, ואז נמשיך עם החומר החדש:
בעיית למידה:

$$(X, Y\mathcal{H}, l)$$

בעיות קלסיפיקציה:

$$Y = \{0,1\},\ l = l_{0-1},\ l_{0-1}\left(\hat{y}, y\right) = \begin{cases} 0 & \hat{y} = y \\ 1 & \hat{y} \neq y \end{cases}$$

אלגוריתם למידה הוא אלגוריתם המקבל מדגם:

$$S = \{(x_1, y_1), ..., (x_m, y_m)\}$$

ומחזיר $h : X \to Y$. השגיאה של $h : X \to Y$ ביחס להתפלגות $\mathcal{D}$ על X, Y היא $L_{\mathcal{D}}\left(h\right) = \mathbb{E}_{(x.y)\sim\mathcal{D}} l\left(h\left(x\right), y\right)$, והשגיאה של $\mathcal{H}$:

$$L_{\mathcal{D}}\left(\mathcal{H}\right) = \underset{h\in\mathcal{H}}{inf} L_{\mathcal{D}}\left(h\right)$$

נזכור כי נרצה למצוא h שתהיה לכל היותר גרועה (גדולה) באפסילון מהשגיאה של המחלקה.
מונחים נוספים:

- נאמר ש־$\mathcal{D}$ **פרידה (או ממומשת או realizable) ע"י** $h^* : X \to Y$ אם, כאשר $(x, y) \sim \mathcal{D}$, מתקיים $y = h^*\left(x\right)$ בהסתברות 1.
- נאמר ש־$\mathcal{D}$ **פרידה ע"י** $\mathcal{H}$ אם $\mathcal{D}$ פרידה ע"י איזושהיא $h^* \in \mathcal{H}$[1].

עבור התפלגות $\mathcal{D}$ על $X \times Y$, סימנו ב־$\mathcal{D}^m$ את ההתפלגות של מדגם בן m דוגמאות הנדגם לפי $\mathcal{D}$, כלומר $\mathcal{D}^m$ היא ההתפלגות של:

$$S = \{(x_1, y_1), ..., (x_m, y_m)\}$$

כאשר (x_i, y_i) ב"ת מתפלגים לפי $\mathcal{D}$.
הגדרנו גם סיבוכיות מדגם ־ עבור אלגוריתם $\mathcal{A}$, נסמן ב־$m_{\mathcal{A}}\left(\varepsilon, \delta\right)$ את המספר הקטן ביותר כך שלכל $m \geq m_{\mathcal{A}}\left(\varepsilon, \delta\right)$ ולכל התפלגות $\mathcal{D}$ על $X \times Y$:

$$\underset{S\sim\mathcal{D}^m}{Pr}\left(L_{\mathcal{D}}\left(A\left(S\right)\right) \geq L_{\mathcal{D}}\left(\mathcal{H}\right) + \varepsilon\right) \leq \delta$$

הרבה פעמים מתייחסים באופן פרטני למקרה בו ההתפלגות פרידה ע"י $\mathcal{H}$, ולכן מגדירים גם:

[1] במקרה זה נאמר שהנחת הפרידות מתקיימת, או כפי שנקרא באנגלית ־ realizability assumption. השם העברי קצת מתעתע, ומתייחס למקרי הספציפי בו מדברים על מחלקת חצאי מרחבים, ואז ההיפותזה ה"נכונה" היא מפרידה בין הדוגמאות השונות באופן מושלם, כלומר המרחב ספרבילי ־ פריד. אם לא עברתם עדיין על כל החומר ומה שכתוב כאן נשמע כמו סינית ־ אל דאגה, it will all make sense מאוחר יותר.

הגדרה 5.3 (סיבוכיות המדגם במקרה הפריד) יהא $\mathcal{A}$ אלגוריתם למידה. עבור $\varepsilon, \delta > 0$ נסמן ב־$m^r_{\mathcal{A}}(\varepsilon, \delta)$ את המספר המנימלי כך שלכל התפלגות $\mathcal{D}$ הפרידה ע"י $\mathcal{H}$ ו־$m \geq m(\varepsilon, \delta)$ מתקיים:

$$Pr_{S\sim\mathcal{D}^m}\left(L_{\mathcal{D}}\left(\mathcal{A}\left(S\right)\right) > \varepsilon\right) \leq \delta$$

אלגוריתם ה־ERM בוחר את האלגוריתם בעל השגיאה האמפירית המינימלית ־ הפעם נעשה את זה לאט יותר:

הגדרה 5.4 השגיאה האמפירית של $h : X \to Y$ ביחס למדגם S מוגדרת באופן הבא:

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

הערה 5.5 זה פשוט ממוצע.

וכעת, טענה פשוטה אך די חשובה, שתראה שעבור מדגם מספיק גדול, $L_{\mathcal{D}}(h) \approx L_S(h)$:

טענה 5.6 נסמן את השגיאה המקסימלית: $B = \sup_{\hat{y}, y \in Y} l(\hat{y}, y)$[2]. אזי לכל התפלגות $\mathcal{D}$, לכל היפותזה $h : X \to Y$ ולכל $\varepsilon > 0$:

$$\Pr_{S\sim\mathcal{D}^m}\left(\left|L_S(h) - L_{\mathcal{D}}(h)\right| \geq \varepsilon\right) < 2e^{-\frac{2\varepsilon^2 m}{B^2}}$$

כלומר "ההסתברות שהשגיאה האמפירית קטנה מהשגיאה (האמיתית) היא מאוד קטנה, ואפילו שואפת במהירות אקספוננציאלית".
לשם ההוכחה, נשתמש במשפט הופדינג:

משפט 5.7 משפט הופדינג: יהיו:

$$Z_1, ..., Z_m \in [0, B]$$

מ"מ ב"ת ש"ה עם תוחלת μ. נסמן:

$$\overline{Z} = \frac{1}{m} \cdot \sum_{i=1}^{m} Z_i$$

אזי:

$$Pr\left(\left|\overline{Z} - \mu\right| > \varepsilon\right) \leq 2\varepsilon^{-\frac{-2\varepsilon^2 m}{B^2}}$$

הוכחה: הסיבה שהטענה נובעת כמעט מידית מהמשפט היא שהאיברים $l\left(h\left(x_i\right), y_i\right)$ הם משתנים מקריים ב"ת ש"ה, ומכך נוכל להגדיר אותם כ־Z_i, וכמובן שבמקרה זה $\overline{Z} = L_S(h)$, $\mu = L_D(h)$ ־ ואז אם $S \sim \mathcal{D}^m$, מהפודינג:

$$Pr\left(\left|L_S(h) - L_{\mathcal{D}}(h)\right| > \varepsilon\right) \leq 2e^{\frac{-2\varepsilon^2 m}{B^2}}$$

■

הגדרה 5.8 אלגוריתם למידה הוא ERM אם לכל מדגם S מתקיים:

$$L_S\left(A\left(S\right)\right) = \inf_{h\in\mathcal{H}} L_S(h)$$

הערה 5.9 יכול להיות יותר מאלגוריתם ERM אחד.

הערה 5.10 הם הכי טובים, ולכן תמיד נשווה אליהם.

[2] הוא יכול להיות אינסוף, אך נניח בזמן הקרוב שהוא סופי.

5.2 סיבוכיות המדגם של מחלקות סופיות

משפט 5.11 נסמן $B = \sup_{\hat{y},y \in Y} l(\hat{y},y)$, $\mathcal{H}$ סופית. אזי לכל ERM $\mathcal{A}$ מתקיים:

$$m_{\mathcal{A}}(\varepsilon,\delta) \leq 2\left(\frac{B}{\varepsilon}\right)^2 \cdot \log\left(\frac{2|\mathcal{H}|}{\delta}\right)$$

הערה 5.12 בסיכום השיעור מופיע משפט חזק יותר ־ שמתקיים במידה ויש במחלקה היפותזה שלא טועה לעולם ־ כלומר קיום הנחת הפרידות.

הוכחה: רעיון ההוכחה: למרבה ההפתעה, נשתמש בטענה שהוכחנו קודם לכן. נשתמש בחסם האיחוד, ונראה שאם m גדול מהביטוי הזה, אז נקבל כי הטענה תהיה נכונה עבור כל ההיפותזות במחלקה. ואם זה נכון, כלומר אם לכל היפותזה השגיאה האמפירית קרובה כל כך לשגיאה האמיתית, אז כולם יהיו קרובות לשגיאה הכי טובה במחלקה.
נקבע התפלגות $\mathcal{D}$ על $X \times Y$, ויהא:

$$S = \{(x_1,y_1),...,(x_m,y_m)\} \sim \mathcal{D}^m$$

כך ש:

$$m \geq 2\left(\frac{B}{\varepsilon}\right)^2 \cdot \log\left(\frac{2|\mathcal{H}|}{\delta}\right)$$

צריך להראות שבהסתברות $\geq 1-\delta$:

$$L_{\mathcal{D}}(A(S)) \leq L_{\mathcal{D}}(\mathcal{H}) + \varepsilon$$

די להראות, שבהסתברות $\geq 1-\delta$, השגיאה האמפירית של כל ההיפותזות קרובה לשגיאה האמיתית, כלומר:

$$(1)\ \forall h \in \mathcal{H}\ |L_S(h) - L_{\mathcal{D}}(h)| < \frac{\varepsilon}{2}$$

זה יספיק, שכן במקרה הזה, מכיוון ואלגוריתם ERM מחזיר היפותזה עם שגיאה אמפירית מינימלית, השגיאה האמיתית תהיה אף היא קרובה למינימלית. קונקרטית, מתקיים:

$$\begin{aligned} L_{\mathcal{D}}(A(S)) &\leq L_S(A(S)) + \frac{\varepsilon}{2} \overset{\mathcal{A} \text{ is ERM}}{=} \inf_{h \in \mathcal{H}} L_S(h) + \frac{\varepsilon}{2} \\ &\leq \inf_{h \in \mathcal{H}} L_{\mathcal{D}}(h) + \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = L_{\mathcal{D}}(\mathcal{H}) + \varepsilon \end{aligned}$$

על כן, נותר להראות כי (1) מתקיים, וכאן נסיים. זאת נעשה ע"י חסם האיחוד. עבור היפותזה $h \in \mathcal{H}$, נסמן ב־U_h את המאורע בו:

$$|L_S(h) - L_{\mathcal{D}}(h)| \geq \frac{\varepsilon}{2}$$

ונסמן:

$$U = \bigcup_{h \in \mathcal{H}} U_h$$

צריך להראות כי ההסתברות של U קטנה מ־δ, וזאת כאמור נעשה ע"י חסם האיחוד.
נשים לב שהגדרת U כנ"ל מאפשרת לנו להשתמש בחסם הופדינג (עם $\frac{\varepsilon}{2}$ בתור הפרמטר, לא כפי שמופיע בניסוח המשפט מעלה), שכן:

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

וזהו סכום של מ"מ ב"ת ש"ה עם תוחלת $L_{\mathcal{D}}(h)$. נקבל:

$$P(U) = P\left(\bigcup_{h\in\mathcal{H}} U_h\right) \leq \sum_{h\in\mathcal{H}} P(U_h) \overbrace{\leq}^{Hoeffding} \sum_{h\in\mathcal{H}} 2e^{\frac{-2\left(\frac{\varepsilon}{2}\right)^2 m}{B^2}} = |\mathcal{H}|\cdot 2e^{\frac{-\epsilon^2 m}{2B^2}}$$
$$\overbrace{\leq}^{m\geq\ldots} |\mathcal{H}|\cdot 2e^{\frac{-\varepsilon^2\cdot 2\left(\frac{B}{\varepsilon}\right)^2\cdot\log\left(\frac{2|\mathcal{H}|}{\delta}\right)}{2B^2}} = 2\cdot|\mathcal{H}|\frac{\delta}{2|\mathcal{H}|} = \delta$$

כנדרש. ■

המשפט הזה פשוט, אבל כבר ממנו אפשר לקבל כמה מסקנות נחמדות.

5.2.1 אפליקצה של המשפט - סיבוכיות מדגם של מחלקות בנות d־פרמטרים

תהא $(X,Y,\mathcal{H},l_{0-1})$ בעיית למידה, $B=1=supl_{0-1}$. נניח שניתן לייצג כל $h\in\mathcal{H}$ ע"י d פרמטרים שכל אחד מהם מיוצג ע"י q ביטים.
במקרה הנ"ל $|\mathcal{H}|\leq 2^{d\cdot q}$, ומהמשפט, סיבוכיות המדגם של אלגוריתם ERM חסומה ע"י:

$$\frac{2\log(|\mathcal{H}|)+2\cdot\left(\frac{2}{\delta}\right)}{\varepsilon^2} \leq \frac{2dq+2\log\left(\frac{2}{\delta}\right)}{\varepsilon^2}$$

ובאמת אם נתעלם מהלוג בצג הימני (שבדר"כ יהיה הרבה יותר קטן מ־$2dq$), נקבל משהו שהוא מאוד מובן אינטואיטיבית - סיבוכיות המדגם חסומה ע"י גודל הייצוג של הבעיה.

דוגמאות כעת, לכמה דוגמאות על מנת להשתכנע שקיימות מחלקות שימושיות מהצורה הזו:

1. **מחלקת הפונקציות הניתנות למימוש ע"י קובץ** exe. **בגודל** d **ביטים**. זו מחלקה שאינטואיטיבית נרצה להשתמש בה, שכן בסופו של יום נרצה לממש את ההיפותזה $h:X\to Y$ שנלמד. למה בכל זאת לא משתמשים במחלקה הזו? למצוא ERM למחלקה הזו היא בעיה "מאוד מאוד קשה, ממש אי אפשר לעשות".
2. נעבור מחלקה שכן משתמשים בה - **מחלקת חצאי המרחבים המוגדרים ע"י Floating points**. נזכיר כי חצי מרחב היא היפותזה מהצורה:

$$x\in\mathbb{R}^d,\ h(x)=a_1x_1+\ldots+a_dx_d+b$$

אם מייצגים כל a_i,b ע"י $q=32$ ביטים (דבר שעושים בפועל), נקבל מחלקה בה כל היפותזה ניתנת לתיאור ע"י $d+1$ פרמטרים המיוצגים כל אחד ע"י $q=32$ ביטים.

6 מימד VC והמשפט היסודי

עבור הרבה מחלקות החסם שכתוב מעלה הוא חסר משמעות, למשל כי המחלקה היא אינסופית. בשיעור הזה והבא נפתח תיאוריה שתעזור לנו להגיד בצורה יותר טובה את סיבוכיות המדגם.
לכל מחלקה $\mathcal{H}$ נגדיר מספר $VC(\mathcal{H})$. המשפט היסודי יראה כי המספר הזה מאפיין בצורה מאוד טובה את סיבוכיות המדגם של המחלקה. הדבר הראשון שנעשה - נגדיר את המספר הזה, ונסביר אך הוא קשור לסיבוכיות המדגם.
נשים לב כי מעכשיו נצטמצם לקלסיפיקציה בינארית, כלומר $Y=\{0,1\}$, $l=l_{0-1}$.

6.1 מימד VC

נתחיל עם סימונים. ההגדרה קצת מבלבלת בהתחלה, נמשיך עם דוגמאות להסביר את ההגדרה הזו.

סימונים

- עבור $h:X\to\{0,1\}$ ו־$A\subset X$, נסמן $h|_A:A\to\{0,1\}$ את ה**צמצום של** h **ל־**A, היא הפונקציה שתחומה A ומקיימת:

$$\forall x\in A h|_A(x)=h(x)$$

- נסמן:

$$\mathcal{H}|_A = \{h|_A | h \in \mathcal{H}\}$$

הגדרה 6.1 תת קבוצה $A \subset X$ **תקרא מנותצת ע"י** $\mathcal{H} \subset \{0,1\}^X$ אם:

$$\mathcal{H}|_A = \{0,1\}^A$$

כלומר, אם $|\mathcal{H}|_A| = 2^{|A|}$. במקרה זה נגיד ש־$\mathcal{H}$ מנתצת את A.

ננסה להסביר למה הביטוי הזה בכלל קשור לסיבוכיות מדגם בתיאור אינטואיטיבי אותו נפרמל בהמשך:
נניח שקיימת בעיית למידה ומחלקת היפוטזות שהיא מנותצת, וניח כי A היא בת 15 איברים. אם אנחנו יודעים את ה־label (הפלטים) של חלק מהאיברים, אין לנו דרך לדעת את התשובות על כל האיברים, ולכן צריך לראות את כל A, ולכן סיבוכיות המדגם תהיה חסומה ע"י הגודל של A. ואם נרצה להביט על המקסימום ־ אם ניקח את הגודל המקסימלי של קבוצה מנותצת, נקבל חסם תחתון, ואף עליון, על הסיבוכיות.

דוגמא נתחיל עם דוגמא פשוטה כדי לנסות להבין מה הולך כאן:

$$\begin{aligned} &X = \{a,b\},\ \mathcal{H} = \{h_1, h_2\} \\ &h_1(a) = 0,\ h_1(b) = 1 \\ &h_2(a) = 0,\ h_2(b) = 0 \end{aligned}$$

ל־X יש 4 תת קבוצות במקרה הזה. האם $\{a\}$ מנותצת? לא! כי אין את הפונקציה ששולחת את האיבר ל־1: כדי לבדוק אם יחידון זה מנותץ, נסתכל על הצמצום של $\mathcal{H}|_A$:

$$\begin{aligned} &A = \{a\} \\ &\mathcal{H}|_A = \{h_1|_A,\ h_2|_A\} \\ &h_1|_A = 0,\ h_2|_A = 0 \end{aligned}$$

וזה לא מנותץ, כי למשל $h_i(a) = 1$ לא מופיעה ב־$\mathcal{H}_A$.
האם $A = \{b\}$ מנותצת? כן!
$\{a,b\}$? לא! נשים לב כי באיחוד יש 4 פונקציות, ובצמצום יש 2, אז "לא היה סיכוי בכל מקרה".
הקבוצה הריקה? תמיד!
ומכאן, להגדרה:

הגדרה 6.2 מימד VC של $\mathcal{H}$ הוא הגודל המקסימלי של קבוצה מנותצת:

$$VC(\mathcal{H}) := sup\{|A| | A \text{is shattered by } \mathcal{H}\}$$

בדוגמא שלנו, $VC(\mathcal{H}) = 1$.
באופן כללי, כדי להראות ש־$VC(\mathcal{H}) = d$, יש להראות שני דברים:

- קיימת קבוצה מנותצת בגודל d.
- אין קבוצה מנותצת בגודל $d+1$.

6.2 הקדמה להוכחת המשפט היסודי

תזכורת מהשבוע הקודם:

- $\mathcal{H} \subset \{0,1\}^X$, $l(y,\hat{y}) = \begin{cases} 1 & y \neq \hat{y} \\ 0 & y = \hat{y} \end{cases}$.

- אלגוריתם למידה מקבל מדגם $S = \{(x_1, y_1), ..., (x_m, y_m)\}$ ומחזיר:

$$h : X \to \{0, 1\}$$

- $$L_{\mathcal{D}}(h) := \mathbb{E}_{(x,y)\sim\mathcal{D}}(h(x), y) = Pr_{(x,y)\sim\mathcal{D}}(h(x) \neq y),\ L_S(h) = \frac{1}{m} \cdot \sum_{i=1}^{m} l(h(x_i), y_i) =$$

$$L_{\mathcal{D}}(\mathcal{H}) := \underset{h\in\mathcal{H}}{min} L_{\mathcal{D}}(h)$$

- $$VC(\mathcal{H}) = max\left\{|A| \,\middle|\, \mathcal{H}|_A = \{0,1\}^A\right\}$$

- ראינו משפט על סיבוכיות מדגם: אם $m \geq m_A(\varepsilon, \delta)$:

$$Pr_{S\sim\mathcal{D}^n}(L_{\mathcal{D}}(A(S)) > L_{\mathcal{D}}(\mathcal{H}) + \varepsilon) \leq \delta$$

6.2.1 המשפט היסודי

משפט 6.3 תהי $\mathcal{H}$ מחלקת היפותזות עבור קלסיפיקציה בינארית מעל X. אזי, $\mathcal{H}$ היא למידה אמ"מ $VC(\mathcal{H})$ הינו סופי.
יתרה מזו, קיימים קבועים גלובליים c_1, c_2 כך שאם $d := VC(\mathcal{H})$, אזי:

$$c_1 \cdot \frac{d + \log\left(\frac{1}{\delta}\right)}{\varepsilon^2} \leq m_{\mathcal{A}}(\varepsilon, \delta) \leq c_2 \cdot \frac{d + \log\left(\frac{1}{\delta}\right)}{\varepsilon^2}$$

ובמקרה בו הנחת הפרידות מתקיימת, אזי:

$$c_1 \cdot \frac{d + \log\left(\frac{1}{\delta}\right)}{\varepsilon} \leq m_{\mathcal{A}}(\varepsilon, \delta) \leq c_2 \cdot \frac{d \cdot \log\left(\frac{1}{\varepsilon}\right) + \log\left(\frac{1}{\delta}\right)}{\varepsilon}$$

אנחנו נוכיח רק את החסם העליון במקרה הראשון.
כמה הערות לפני ההוכחה:

- לא מדובר בהוכחה פשוטה. כנראה שמבחינה מתמטית זה השיעור הכי מסובך בקורס.
- ההוכחה מדברת על בעיות קלסיפיקציה בינארית בלבד.
- למשפט היסודי יש גם חלק שלא ניסחנו (נמצא ברשימות המרצה באתר), שנותן גם חסם תחתון, ע"י גודל דומה עם קבוע קטן יותר. לכן, מנקודת המבט הזו, ל־ERM סיבוכיות המדגם הטובה ביותר. על החסם התחתון, שהוא פשוט יותר, נדבר באחד מהתרגולים הקרובים.

מבנה ההוכחה

ההוכחה היא בעצם מסקנה משתי למות עיקרות, אותן ננסח ונוכיח.
האסטרטגיה: להראות שכאשר m "גדול", אז בהסתברות $\leq 1 - \delta$:

$$\forall h \in \mathcal{H},\ |L_{\mathcal{D}}(h) - L_S(h)| < \frac{\varepsilon}{2}$$

עבור מחלקה סופית, השתמשנו בחסם האיחוד. עכשיו לא נוכל לעשות את זה, כי יתכן ו־$\mathcal{H}$ אינסופי. הדרך שבה נתמודד עם בעיה זו היא כאמור שתי למות:

- **למת סאור־שלח:**

$$\forall A \subset X \left(|\mathcal{H}|_A|\right) \leq \left(|A|+1\right)^d,\ d = VC\left(\mathcal{H}\right)$$

איך למה זו עוזרת לנו? כשמגדילים את A, הגודל של הצמצום $\mathcal{H}$ ל־A גדל לאט. זו הדרך שלנו לעבור למחלקה "סופית", ולא להשתמש בחסם האיחוד.

- **למת המזגם הכפול** תעזור לנו לעבור מהלמה הראשונה למשפט.

6.3 למת המזגם הכפול

הגדרה 6.4 פונקציית הגידול של $\mathcal{H}$ היא הפונקציה:

$$\pi_{\mathcal{H}} : \mathbb{N} \to \mathbb{N},\ \pi_{\mathcal{H}}\left(m\right) := \underset{A \subset X\ |A| \leq m}{max} |\mathcal{H}|_A|$$

דוגמאות

- $\mathcal{H} = \{0,1\}^X$ עבור $|X| \leq \infty$. אזי:

$$\pi_{\mathcal{H}}\left(m\right) = \begin{cases} 2^m & m \leq |X| \\ 2^{|X|} & m > |X| \end{cases}$$

- $|X| = \infty$, בשביל שיהיה לנו קל לדמיין, אפשר לחשוב על $X = \{1,2,3,...\}$.

$$\mathcal{H} := \left\{h : X \to \{0,1\} \middle| \left|h^{-1}\left(1\right)\right| = 1\right\}$$
$$\Rightarrow \pi_{\mathcal{H}}\left(m\right) = m+1$$

מדוע זה נכון? עבור $A \subset X$ בגודל m:

$$\mathcal{H}|_A = \left\{h : A \to \{0,1\} \middle| \left|h^{-1}\left(1\right)\right| \leq 1\right\}$$

שכן לכל היותר אפשר למפות את המספר המקורי שלא התמפה ל־0 ל־1, אבל אם הוא בכלל לא נמצא בקבוצה הכל פשוט ימופה ל־0. ברור כי יש $m+1$ פונקציות כאלו (מיפוי של כל מספר ב־m ל־1, והפונקציה שממפה הכל לאפס). אפשר לראות שבדוגמא הזו באמת התצמצמנו במספר הפונקציות שאנחנו מסתכלים עליהם ־ ובתכונה זו נשתמש בהמשך.

למה 6.5 המזגם הכפול: לכל מחלקה מתקיים:

$$Pr_{S \sim \mathcal{D}^m}\left(\exists h \in \mathcal{H} \text{ s.t. } \left|L_S\left(h\right) - L_{\mathcal{D}}\left(h\right)\right| \geq \varepsilon\right) \leq \pi_{\mathcal{H}}\left(2m\right) \cdot 4e^{-\frac{\varepsilon^2 m}{8}}$$

שימו ♡! **הוכחת** למת המזגם הכפול לא נכללת בחומר הבחינה.
הניסוח של הלמה נכלל בחומר הבחינה.

נדבר תחילה על רעיון ההוכחה ־ ואז נפרק אותה לשתי תת־למות.
הרעיון לדעת המרצה הוא די מתוחכם ומקורי, שעושה את הדבר הבא:
יש לנו מדגם S בן m דגימות. נבצע הנחה מאוד משונה: נניח כי S נדגם בשני חלקים:
בחלק הראשון, נדגם מדגם S' שגדול מ־S המקורי:

$$S' = \{(x'_1, y'_1), ..., (x'_{2m}, y'_{2m})\}$$

בשלב השני, בוחרים חצי מן המדגם המקורי; כיצד? עוברים על כל זוג עוקב, ובוחרים אחת מהן באקראי[3].
ההנחה הזו לא מגבילה את הכלליות, שכן עדיין $S \sim \mathcal{D}^m$. איך היא עוזרת? היא מאפשרת לנו "להחליף" את ההתפלגות $\mathcal{D}$ עם המדגם הראשוני S'. חצי הלמה הראשון יראה ש"זה יחסית בסדר".

[3] הבחירה בשיטה הזו של אחד מכל זוג עוקב היא שרירותית לגמרי, אך תפשט את ההוכחה בהמשך, ועל כן אנו משתמשים ספציפית בה.

תת־למה ראשונה

למה 6.6

$$Pr\left[\overbrace{\exists h\in\mathcal{H}\text{ s.t. }\left|L_S\left(h\right)-L_{\mathcal{D}}\left(h\right)\right|\geq\varepsilon}^{A}\right]\leq Pr\left[\overbrace{\exists h\in\mathcal{H}\text{ s.t. }\left|L_S\left(h\right)-L_{S'}\left(h\right)\right|\geq\frac{\varepsilon}{4}}^{B}\right]+2\cdot\exp^{-\frac{\varepsilon^2 m}{2}}$$

הוכחה: נראה ש:

$$(*)Pr\left(B|A\right)\geq1-2\cdot\exp^{-\frac{-\varepsilon^2 m}{2}}$$

וזה מספיק, שכן אז:

$$\begin{aligned}&Pr\left(B\right)=Pr\left(B|A\right)Pr\left(A\right)+Pr\left(B|A^C\right)Pr\left(A^C\right)\\&\geq Pr\left(B|A\right)Pr\left(A\right)\geq\left(1-2\exp^{-\frac{\varepsilon^2 m}{2}}\right)Pr\left(A\right)\geq Pr\left(A\right)-2\exp^{-\frac{\varepsilon^2 m}{2}}\\&\Rightarrow Pr\left(B\right)\geq Pr\left(A\right)-2\exp^{-\frac{\varepsilon^2 m}{2}}\Rightarrow Pr\left(B\right)+2\exp^{-\frac{\varepsilon^2 m}{2}}\geq Pr\left(A\right)\end{aligned}$$

נראה אם כך את $(*)$:
נניח ש־A מתקיים, כלומר קיימת $h\in\mathcal{H}$ כך ש:

$$\left|L_D\left(h\right)-L_S\left(h\right)\right|\geq\varepsilon$$

אזי, נשים לב ש־ $S'\backslash S$ מתפלג לפי $\mathcal{D}^m$ באופן בלתי תלוי ב־S. לכן לפי הופדינג, עבור אותה h:

$$Pr\left(\left|L_{S'\backslash S}\left(h\right)-L_{\mathcal{D}}\left(h\right)\right|<\frac{\varepsilon}{2}\right)\geq1-2\cdot\exp^{-\frac{\varepsilon^2 m}{2}}$$

נטען כי אם זה קרה, אזי מתקיים מאורע B. במקרה הזה (בשורה הראשונה יש לחלק בשתיים בגלל גודל המדגם של S לעומת S'):

$$\begin{aligned}\left|L_S\left(h\right)-L_{S'}\left(h\right)\right|&=\left|L_S\left(h\right)-\frac{L_S\left(h\right)+L_{S'\backslash S}\left(h\right)}{2}\right|=\left|\frac{L_S\left(h\right)}{2}-\frac{L_{S'\backslash S}\left(h\right)}{2}\right|\\&=\frac{1}{2}\cdot\left|L_S\left(h\right)-L_{S'\backslash S}\left(h\right)\right|\geq\frac{1}{2}\left(\left|L_S\left(h\right)-L_D\left(h\right)\right|-\left|L_{\mathcal{D}}\left(h\right)-L_{S'\backslash S}\left(h\right)\right|\right)\\&\geq\frac{1}{2}\left(\varepsilon-\frac{\varepsilon}{2}\right)=\frac{\varepsilon}{4}\end{aligned}$$

■

תת־למה שניה

תת למה זו תחסום את ההסתברות של המאורע B.

למה 6.7

$$Pr\left(\exists h\in\mathcal{H}\text{ s.t. }\left|L_{S'}\left(h\right)-L_S\left(h\right)\right|\geq\varepsilon\right)\leq\pi_{\mathcal{H}}\left(2m\right)\cdot2\exp^{-2\varepsilon^2 m}$$

הוכחה: נפרק את ההוכחה לשני שלבים:
שלב א': h בודדת. מתקיים:

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

הדרך שבה בחרנו את S מתוך S' היא בחירת דוגמא אחת מתוך כל זוג דוגמאות עוקבות. ולכן, כאשר נסתכל על S' במקום S נקבל:

$$= \frac{1}{m} \cdot \sum_{i=1}^{m} \overbrace{(1-Z_i) \cdot l\left(h\left(x'_{2i-1}\right), y'_{2i-1}\right) + Z_i \cdot l\left(h\left(x'_{2i}\right), y'_{2i}\right)}^{U_i}$$

כאשר $Z_i \in \{0,1\}$ הוא מטבע הוגן (שמייצג את הבחירה האקראית שלנו מתוך הזוג). נשים לב כי $U_i \in [0,1]$ ב"ת (בהנתן S'). מתקיים ש:

$$\begin{aligned}
\mathbb{E}\left[L_S\left(h\right)\right] &= \mathbb{E}\left[\frac{1}{m} \cdot \sum_{i=1}^{m} U_i\right] = \frac{1}{m} \cdot \sum_{i=1}^{m} \mathbb{E}\left[U_i\right] \\
&= \frac{1}{m} \cdot \sum_{i=1}^{m} \frac{l\left(h\left(x'_{2i-1}\right), y'_{2i-1}\right) + l\left(h\left(x'_{2i}\right), y'_{2i}\right)}{2} \\
&= \frac{1}{2m} \cdot \sum_{i=1}^{2m} l\left(h\left(x'_i\right), y'_i\right) = L_{S'}\left(h\right)
\end{aligned}$$

ולכן נוכל להשתמש בחסם הופדינג! לפיו מתקיים:

$$Pr\left(\left|L_S\left(h\right) - L_{S'}\left(h\right)\right| \geq \varepsilon\right) \leq 2 \cdot \exp^{-2\varepsilon^2 m}$$

ועתה, נותר רק להשתמש בחסם האיחוד - אך צריך להצדיקו קודם. לשם כך, נשים לב ש:

$$\begin{aligned}
L_S\left(h\right) &= L_S\left(h|_{S'}\right) \\
L_{S'}\left(h\right) &= L_{S'}\left(h|_{S'}\right)
\end{aligned}$$

ולכן:

$$\begin{aligned}
Pr\left(\exists h \in \mathcal{H} \left|\left|L_S\left(h\right) - L_{S'}\left(h\right)\right| \geq \varepsilon\right.\right) &= Pr\left(\exists h \in \mathcal{H}|_{S'} \left|\text{""}\right.\right) \leq \left|\mathcal{H}|_{S'}\right| \cdot 2\exp^{-2\varepsilon^2 m} \\
&\leq \pi_{\mathcal{H}}\left(2m\right) \cdot 2\exp^{-\frac{\varepsilon^2 m}{2}}
\end{aligned}$$

■

כמה מילים על הלמה: נרצה להשתמש בה כדי להוכיח את המשפט היסודי. כדי להשתמש בה נצטרך חסם על פונקציית הגידול. ההוכחה של הלמה הזו תהיה קצת יותר קלה מהוכחת הלמה הראשונה.

6.4 למת סאור־שלח

למה 6.8 לכל $\mathcal{H} \subset \{0,1\}^X$, כאשר $|X| = m$ ו־$d = VC\left(\mathcal{H}\right)$, מתקיים:

$$|\mathcal{H}| \leq \left(\begin{array}{c} m \\ 0 \end{array}\right) + \left(\begin{array}{c} m \\ 1 \end{array}\right) + ... + \left(\begin{array}{c} m \\ d \end{array}\right) := N\left(d,m\right)$$

אבחנה: $N\left(d,m\right)$ הוא מספר הדרכים לבחור קבוצה בגודל $\geq d$ מתוך $\{1,...,m\}$.

הערה 6.9 החסם בלמת סאור־שלח הדוק. כלומר, קיימת דוגמא עבורה זהו בדיוק החסם. נראה זאת - המחלקה:

$$\mathcal{H} = \left\{h : [m] \to \{0,1\} \left|\left|h^{-1}\left(1\right)\right| \leq d\right.\right\}$$

מקיימת:

$$|\mathcal{H}| = N\left(d,m\right)$$

וכמו כן:

$$VC\left(\mathcal{H}\right) = d$$

כי למעשה כל תת־קבוצה בגודל d היא מנותצת. כמו כן, כל תת קבוצה בגודל $d+1$ חייבת להיות לא מנותצת. נשים לב כי גם $|\mathcal{H}| = \left(\begin{array}{c} m \\ 0 \end{array}\right) + \left(\begin{array}{c} m \\ 1 \end{array}\right) + ... + \left(\begin{array}{c} m \\ d \end{array}\right)$.

מסקנה 6.10 אם $d = VC(\mathcal{H})$, אזי:

$$\pi_{\mathcal{H}}(m) \leq N(d,m) \leq (m+1)^d$$

הוכחה: תהי $A \subset X$ בגודל $m \geq$. מההגדרה $VC(\mathcal{H}|_A) \leq d$, ולכן מלמת סאור־שלח:

$$|\mathcal{H}|_A| \leq N(d,m) \leq (m+1)^d$$

נסביר את המעבר האחרון: אם נוסיף איבר נוסף $*$, אם נרצה לבחור פחות מ־d אפשר פשוט לבחור את $(*)$. כך אפשר לתאר את הבחירה, ולכן קיבלנו את המעבר האחרון. ■

ועתה, נוכיח את הלמה, באינדוקציה: **הוכחה:** באינדוקציה על $d+m$. נתאר בצורה מסודרת איך האינדוקציה תעבוד: מה אנחנו יודעים על היחס בין m ל־d? $m \geq d \geq 0$. לכן, אם נצייר את כל הערכים האפשריים על ציר x (עבור ערכי d) ו־y (עבור ערכי m), האיברים שמעניינים אותנו אלו האיברים שמעל האלכסון. נוכיח באינדוקציה על כל האיברים על האלכסון, ועל האיברים על ציר ה־y, ואז נסיק עבור הערכים "באמצע".
נניח בה"כ כי $X = [m] := \{1, ..., m\}$

בסיס האינדוקציה: יש לנו כאמור שני בסיסים:

- עבור $d = 0$: מכיוון ו־$d = 0$, $\mathcal{H}$ מכילה לכל היותר פונקציה בודדת (אם היו יותר, אז היתה קבוצה בגודל 1 שהיתה יכולה להיות מנותצת, ואז $d \neq 0$).כאן בהכרח $|\mathcal{H}| = 1 \leq N(0,m)$.
- עבור $d = m$: במקרה זה:

$$|\mathcal{H}| \leq 2^d = N(d,d)$$

שלב האינדוקציה: נניח בה"כ $X = [m]$. המקרה $d = m$ טופל בבסיס האינדוקציה, לכן נוכל להניח ש־$m > d$.
נגדיר שתי מחלקות היפותזות:

$$\mathcal{H}_1 = \mathcal{H}|_{[m-1]}$$
$$\mathcal{H}_2 = \left\{ \mathcal{H}|_{[m-1]} : h : [m-1] \to \{0,1\} \,\middle|\, \exists h_1, h_2 \in \mathcal{H} \text{ s.t. } h_1|_{[m-1]} = h_2|_{[m-1]} = h,\ h_1 \neq h_2 \right\}$$

כלומר, $\mathcal{H}_1$ היא אוסף כל הפונקציות מ־$[m-1]$ ל־$\{0,1\}$ שניתן להרחיב לפונקציות מ־$[m]$ ל־$\{0,1\}$, השייכת ל־$\mathcal{H}$, ו־$\mathcal{H}_2$ היא אוסף כל הפונקציות $h : [m-1] \to \{0,1\}$ שניתן להרחיב **בשתי דרכים שונות** לפונקציה מ־$[m]$ ל־$\{0,1\}$ב־$\mathcal{H}$. נשים לב כי $\mathcal{H}_1 \supset \mathcal{H}_2$.
עובדה 1:

$$VC(\mathcal{H}_1) \leq d$$

הוכחה: אם $A \subset [m-1]$ מנותצת ע"י $\mathcal{H}_1$, אז היא גם מנותצת ע"י $\mathcal{H}$. מכיוון ו־$VC(\mathcal{H}) \leq d$, אזי $|A| \leq d$. ■

עובדה 2:

$$VC(\mathcal{H}_2) \leq d-1$$

הוכחה: אם $A \subset [m-1]$ מנותצת על $\mathcal{H}_2$, אז $A \cup \{m\}$ מנותצת ע"י $\mathcal{H}$, ולכן:

$$|A| = |A \cup \{m\}| - 1 \leq d-1$$

■

עובדה 3:

$$|\mathcal{H}| = |\mathcal{H}_1| + |\mathcal{H}_2|$$

הוכחה: עבור $h \in \mathcal{H}_1$ נסמן:

$$G_h = \left\{ h' \in \mathcal{H} \,\middle|\, h'|_{[m-1]} = h \right\}$$

לא קשה לראות ש־$\mathcal{H} = \dot{\bigcup}_{h\in\mathcal{H}_1} G_h$. כמו כן, לכל $h \in \mathcal{H}_1$, $|G_h| = 1$, או $|G_h| = 2$. במקרה הראשון $h \in \mathcal{H}_1$ אבל $h \notin \mathcal{H}_2$, ולכן G_h "תורמת" $|G_h| = 1$ לסכום $|\mathcal{H}_1| + |\mathcal{H}_2|$. במקרה השני, $h \in \mathcal{H}_1$ וגם $h \in \mathcal{H}_2$, ולכן G_h "תורמת" $|G_h| = 2$ לסכום הנ"ל.

לסיכום, כל $h \in \mathcal{H}$ תורם בדיוק $|G_h|$ לסכום, ומכיוון וגודל האיחוד של G_h הוא בדיוק $|\mathcal{H}|$, כך גם הסכום, כנדרש. ■

עובדה 4:

$$N(d,m) = N(d-1,m-1) + N(d,m-1)$$

הוכחה: בסיכומים יש שתי הוכחות פורמליות, כאשר אחת מהן היא סיפור קומבינטורי. המרצה העביר את הסיפור הקומבינטורי "בנפנופי ידיים"; ההוכחה די פשוטה, וראינו אותה במתמטיקה דיסקרטית. נביא כאן את ההוכחה האלגברית, המסתמכת על השיוויון הבא:

$$\begin{aligned}
&\forall j < m \quad \binom{m}{j} = \binom{m-1}{j} + \binom{m-1}{j-1} \\
\Rightarrow N(d,m-1) + N(d-1,m-1) &= \binom{m-1}{0} + ... + \binom{m-1}{d-1} + \binom{m-1}{0} + ... + \binom{m-1}{d} \\
&= \binom{m-1}{0} + \left[\binom{m-1}{0} + \binom{m-1}{1}\right] + ... + \left[\binom{m-1}{d-1} + \binom{m-1}{d}\right] \\
&= \binom{m}{0} + \binom{m}{1} + \binom{m}{d} = N(d,m)
\end{aligned}$$

■

כעת סופסוף אנחנו מוכנים לסיים: נקב זוג (m,d) עבורו נרצה להוכיח. נניח כי הלמה נכונה לכל זוג (m',d') כך ש: $m+d > m'+d'$. אזי, מההנחה ומעובדות $4-1$ נובע:

$$\begin{array}{ccc}
|\mathcal{H}| & \overset{(3)}{\underset{\downarrow}{=}} & |\mathcal{H}_1| + |\mathcal{H}_2| \\
& \overset{4}{\underset{\downarrow}{=}} & N(d,m) \\
& & \overset{induction+\ facts\ 1,2}{\underset{\downarrow}{\leq}} \\
& & N(d,m-1) + N(d-1,m-1)
\end{array}$$

וסיימנו!!! ■

6.5 הוכחת המשפט היסודי כתוצאה משתי הלמות

הוכחה: ההוכחה המפורטת "קצת מייגעת ולכן לא נעשה אותה כאן" - נמצאת בסיכומי השיעור של המרצה.
ראינו בהוכחת החסם עבור מחלקות סופיות שדי להראות שכאשר m גדול מספיק מתקיים:

$$\overbrace{Pr\left(\exists h \in \mathcal{H} \left| |L_S(h) - L_{\mathcal{D}}(h)| \geq \frac{\varepsilon}{2}\right.\right)}^{(*)} \leq \delta$$

משתי הלמות נקבל:

$$(*) \quad \overbrace{\leq}^{\text{first lemma}} \quad \pi_{\mathcal{H}}(2m) \cdot 3\exp^{-\frac{\varepsilon^2 m}{3^2}} \quad \overbrace{\leq}^{\text{second lemma}} \quad (2m+1)^d < \delta$$

כאשר ההוכחה של המעבר האחרון מופיעה בסיכומי השיעורים ו"לא מעניינת במיוחד". ■

התרגיל שפורסם היום יעזור לנו לחזור על מה שעשינו היום - זאת ע"י הוכחה מודרכת שנראית כמעט בדיוק אותו דבר, הכללה של המשפט הזה של קלסיפיקציה לא בינארית.

בשיעור הבא נסיים את התיאוריה הזו. נתחיל את השיעור בסקירה אינטואיטיבית של מה שראינו. בשלב הבא של הקורס, שיקח 6-5 שבועות, נפתח אלגוריתמי למידה.

בשלושת השיעורים הראשונים למדנו בעיקר על תורת ההכללה. תורה זו אפשרה לנו לראות מתי יש לנו מספיק דוגמאות כדי ללמוד.

חסרות לנו עדיין שתי נקודות:

- השאלה הראשונה שכמעט לא התייחסנו אליה היא, איך לבחור את $\mathcal{H}$?
- בהנתן שכבר בחרנו את $\mathcal{H}$, הסתמכנו על כך שקיימת היפותזה טובה, אבל לא דיברנו על איך למצוא אותה. כלומר, השאלה היא, איך למצוא $h \in \mathcal{H}$ עם שגיאה אמפירית נמוכה \ מינימלית.

נתחיל את השעור בסיכום של תורת ההכללה כפי שעשינו עד כה, וננסה להבין מה היא אומרת לנו על המציאות. על הדרך נדבר קצת על השאלה הראשונה. בחלק השני של השיעור ובמחשת השבועות הקרובים, נתמקד בשאלה השניה.

7 סיכום ביניים מס' 1 - תורת ההכללה

7.1 אז מה בעצם למדנו עד כה?

המטרה הבסיסית בלמידה חישובית - ללמוד $h^* : X \to Y$, וזאת בהתבסס על מדגם:

$$S = \{(x_1, y_1), ..., (x_m, y_m)\} \subset X \times Y$$

התבססנו על העובדה שקיימת התפלגות מסויימת $\mathcal{D}$ על פיה נדגם S, והגדרנו את ההפסד של היפותזה h בהתאם לה:

$$L_{\mathcal{D}}(h) = \mathbb{E}_{(x,y)\sim\mathcal{D}} l(h(x), y)$$

כאשר l מודדת את המרחק בין זוג פלטים.

האבחנה הראשונה - "אין ארוחות חינם": לא קיים אלגוריתם המסוגל ללמוד כל מיפוי כאשר כמות הדוגמאות מוגבלת. במילים אחרות, צריך איזשהוא **ידע מוקדם** על הפונקציה h^* (או ההתפלגות $\mathcal{D}$) על מנת להיות מסוגלים ללמוד.

הדרך שבה ניתן לבטא ידע מוקדם במודל PAC היא הצבעה על **מחלקת היפותזות** (כלומר אוסף $\mathcal{H} \subset Y^X$) בה אנו יודעים (או, בדר"כ, מניחים) שקיימת היפותזה טובה. תורת ההכלה שפיתחנו הראתה שכאשר $\mathcal{H}$ "קטנה" בהשוואה לכמות הדוגמאות שבידינו, השגיאה האמפירית $L_S(h)$ של כל ההיפותזות ב-$\mathcal{H}$ קרובה לשגיאה האמיתית $(L_{\mathcal{D}}(h))$. לכן, כל אלגוריתם המחזיר היפותזה ב-$\mathcal{H}$ עם שגיאה אמפירית טובה (בפרט, ERM), יחזיר בעצם היפותזה עם שגיאה אמיתית טובה. לכן, בהנתן אלגוריתם כזה, הידע המוקדם שלנו נכון ובאמת קיימת ב-$\mathcal{H}$ היפותזה טובה, נוכל ללמוד..

7.1.1 סכמה לשימוש בלמידה

1. בחר מחלקת היפותזות $\mathcal{H}$ בה מעריכים ש:
   (א) יש היפותזה טובה.
   (ב) $\mathcal{H}$ תהיה קטנה מספיק ביחס לכמות הנתונים שברשותינו.
2. (כלל ה-ERM): החזר $h \in \mathcal{H}$ עם שגיאה אמפירית $L_S(h)$ מינימלית ביחס לנתונים שאספנו.

נפרק את השגיאה של ההיפותזה שאנחנו מחזירים לשניים - השגיאה האמיתית של ההיפותזה, וההפרש בין האפרוקסימציה למינימלית. השגיאה השניה תקרא שגיאת ההכלה.

7.1.2 שגיאת קירוב ושגיאת הכללה - יחסי גומלין

נוח לפרק את $L_{\mathcal{D}}(h)$ לשני רכיבים:

$$L_{\mathcal{D}}(h) = \overbrace{L_{\mathcal{D}}(h) - L_D(\mathcal{H})}^{\text{Estimation error}} + \overbrace{L_{\mathcal{D}}(\mathcal{H})}^{\text{Approximation error}}$$

שגיאת ההכללה: ההפרש בין השגיאה (האמתית) של ההיפותזה עם השגיאה האמפירית הטובה ביותר, לבין השגיאה (האמיתית) של ההיפותזה הטובה במחלקה. הרכיב הזה יהיה קטן יותר ככל שיהיו לנו יותר דוגמאות. עבור קלסיפיקציה בינארית, תורת ההכלה שפיתחנו מאפשרת לנו לחסום את שגיאת ההכלה בעזרת $VC(\mathcal{H})$. נעיר שיש תורות דומות עבור בעיות מעבר לקלסיפיקציה בינארית (למשל בתרגיל טיפלנו בקלסיפיקציה רב מחלקתית).

שגיאת הקירוב: היא השגיאה של ההיפותזה הטובה במחלקה. יש שתי דרכים להקטין אותה:

- **ידע מוקדם:** לבחור $\mathcal{H}$ טובה, כלומר אחת שמתאימה לבעיה (שיש בה היפותזה טובה).
- **שימוש במחלקה עשירה יותר**: לבחור $\mathcal{H}$ גדולה יותר - ככל שיהיו יותר היפותזות, שגיאת הקירוב תהיה קטנה יותר.

נעיר שקיים טריידאוף בין שגיאת ההכללה לשגיאת הקירוב: ככל שנגדיל את $\mathcal{H}$, יש סיכוי טוב יותר למציאת h טובה יותר, כלומר להקטין את שגיאת הקירוב, אבל אנחנו עלולים לפגוע בשגיאת ההכללה! ככל שעובדים עם מחלקה גדולה יותר, סביר ששגיאת ההכללה תהיה גדולה יותר. על כן יש כאן משחק עדין, צריך למצוא את שיווי המשקל הטוב ביותר, ונדבר על כך בהמשך.

7.2 מה הלאה? השלב הבא בקורס: אופטימיזציה

על מנת להפעיל את הסכמה הזו, לא מספיק לבחור $\mathcal{H}$ טובה, אלא לממש בצורה טובה את שלב 2. לצורך זה, ש לפתור את הבעיה שנקרא לה $OPT(\mathcal{H})$:
בהנתן מדגם S, מצא $h \in \mathcal{H}$ עם $L_S(h)$ מינימלית.
למרבה הצער, אוסף המחלקות בהן ניתן לפתור את הבעיה הנ"ל איננו עשיר דיו. על כן, בשבועות שלאחר מכן נלמד דרכים לעבוד עם מחלקות בהן זוהי בעיה קשה חישובית.
אבל, לפני שנמשיך לשם, נזכיר את הכלים שראינו:

7.3 כלים טכניים ומתמטיים שראינו

- חסם הופדינג ושיערוך השגיאה $L_{\mathcal{D}}(h)$ ע"י השגיאה האמפירית
- חסם האיחוד
- למידה ע"י התכנסות במ"ש: לא קראנו לזה כך, אבל הראנו שעבור מדגם מספיק גדול, שגיאת המדגם דומה מאוד לשגיאה האמיתית. למחלקה קוראים "למידה באמצעות התכנסות במידה שווה"
- פונקציית הגידול ולמדת המדגם הכפול
- למת סאור שלח: פונקציה קומבינטורית עזרה לנו לחסום את פונקציית הגידול
- חישוב מימד VC: ראינו בעיקר בתרגול ובתרגילים. זה חשוב כדי להפעיל את התורה שלמדנו על מחלקות קונקרטיות אותן נפגוש בהמשך.

חלק II
אלגוריתמי למידה

8 אופטימיזציה

8.1 קמירות על קצה המזלג

כאמור, בהנתן בעיית למידה, $(X, Y, \mathcal{H}, l)$, על מנת לממש את כלל ה־ERM, אנו רוצים ללמוד את בעיית האופטימיזציה שהגדרנו מעלה $OPT(\mathcal{H})$. ככל הנראה, אלו בעיות שלא ניתן לפתור בקלות ו\או ביעילות, ולכן מסתכלים על משפחות של בעיות אופטימיזציה שאותן כן ניתן לפתור ביעילות. אחת המשפחות האלה היא **בעיות אופטימיזציה קמורות**, ולפני שניגש אליה, צריך ללמוד קצת על קמירות. בהמשך נשתמש במשפחה זו על מנת לפתור את $OPT(\mathcal{H})$.

עכשיו נלמד איך לממש ERM עבור בעיות למידה שניתן לפתור אותן ע"י בעיות אופטימיזציה קמורה. זו באמת משפחה נחמדה, לא טריוויאלית, שמשתמשים בה בפועל. עם זאת, היא לא תאפשר לנו לפתור את כל הבעיות שנרצה. לכן, נאלץ בהמשך להסתכל גם על בעיות שלא ניתן לפתור ע"י אופטימיזציה קמורה, ונלמד דרכים לטפל בהן בהמשך. אחת הדרכים: נשתמש בשיטה הזו ע"מ לפתור בעיות שאינן בעיות באמת אופטימיזציה קמורה.

8.1.1 קבוצות קמורות

הגדרה 8.1 קבוצה $W \subset \mathbb{R}^n$ תקרא **קמורה** אם לכל $x, y \in W$ ולכל $0 \leq \lambda \leq 1$, מתקיים:

$$\lambda x + (1-\lambda) y \in W$$

דוגמאות - בציור! מה זה אומר: לכל שתי נקודות בקבוצה, כל קו שעובר ביניהם מוכל בקבוצה.
נוכיח כי כדורים ב־$\mathbb{R}^n$ הם קמורים:

טענה 8.2 יהא $r > 0$. נסמן:

$$B = \{ x \in \mathbb{R}^n | \, \|x\| \leq r \}$$

אזי, B קמורה.

הוכחה: איך מוכיחים טענה מהסוג הזה? לוקחים שתי נקודות כלשהן ו־λ, ומוכיחים שהנקודה שמתקבלת היא חלק מהקבוצה.

$$\|\lambda x - (1-\lambda) y\| \overbrace{\leq}^{\triangle} \|\lambda x\| + \|(1-\lambda) y\| = |\lambda| \cdot \|x\| + |1-\lambda| \cdot \|y\|$$
$$\overbrace{\leq}^{x,y \in B} |\lambda| \cdot r + |1-\lambda| \cdot r = r$$

ולכן $\lambda x + (1-\lambda) y \in B$, כנדרש. ■

8.1.2 פונקציות קמורות

ראינו כבר בעבר כמה פונקציות קמורות[4], גם אם לא אמרנו זאת. למשל היפרבולה מחייכת - אם נחבר שתי נקודות על ההיפרבולה, הקו המחבר ביניהן הוא מעל לגרף (ועל כן מקיים את כלל הקמירות)!

הגדרה 8.3 תהא $W \subset \mathbb{R}^n$ קבוצה קמורה. פונק' $f : W \to \mathbb{R}$ תקרא **קמורה** אם לכל $x, y \in W$ ולכל $0 \leq \lambda \leq 1$:

$$f(\lambda x + (1-\lambda) y) \leq \lambda f(x) + (1-\lambda) f(y)$$

[4] בחלק מהשנים החומר הזה נכלל גם באינפי 1.

הערה 8.4 בצורה גיאומטרית, ניתן לאפיין פונקציה קמורה ע"י כך שפונקציה קמורה אמ"מ הקבוצה מעל גרף הפונקציה, הנקראת אפיגרף:

$$epi\left(f\right):=\left\{\left(x,y\right)|x\in W,\ y\geq f\left(x\right)\right\}\subset\mathbb{R}^{n+1}$$

היא קבוצה קמורה.

נשתמש בשתי למות שיעזרו לנו לזהות ולבנות פונקציות קמורות, הראשונה עבור פונקציה במשתנה אחד:

למה 8.5 תהי $f:\mathbb{R}\to\mathbb{R}$ פונקציה גזירה פעמיים. אזי f קמורה אמ"מ:

$$\forall x\in\left(a,b\right),\ f''\left(x\right)\geq0$$

ההוכחה לא קשה במיוחד, אך לא נביא אותה כאן. בעזרת למה זו אפשר להסיק, למשל ש־$x,\ \left|x\right|^{P},\ e^{x},\ -\log\left(x\right)$ קמורות, ועוד רבות אחרות.
הלמה הבאה תעזור לנו עם פונקציות במספר משתנים:

למה 8.6

1. אם $f:\mathbb{R}\to\mathbb{R}$ קמורה, ונגדיר $F:\mathbb{R}^{n}\to\mathbb{R}$ ע"י:

$$F\left(x\right)=f\left(a_{1}x_{1}+...+a_{n}x_{n}\right)$$

כאשר $a_{1},...,a_{n}\in\mathbb{R}$ קבועים. אזי F קמורה[5].
בשתי התכונות הבאות נגלה שקב' הפונקציות הקמורות סגורה תחת פעולות מסויימות:

2. (פונקציות קמורות סגורות לחיבור וכפל חיובי) אם $f_{1},...,f_{k}:W\to\mathbb{R}$ קמורות, אזי גם:

$$\sum_{i=1}^{k}\alpha_{i}f_{i}$$

עבור $\alpha_{1},...,\alpha_{k}\geq0$

3. (פונקציות קמורות סגורות תחת מקסימום) אם $f_{1},...,f_{k}:W\to\mathbb{R}$ קמורות אז כך גם:

$$f\left(x\right):=\max_{1\leq i\leq k}f_{i}\left(x\right)$$

נוכיח את 1, השאר נותר לסטודנט השקדן: **הוכחה:** יהיו $x,y\in\mathbb{R}^{n}$, $0\leq\lambda\leq1$, צ"ל:

$$F\left(\lambda x+\left(1-\lambda\right)y\right)\leq\lambda F\left(x\right)+\left(1-\lambda\right)F\left(y\right)$$

ואכן:

$$F\left(\lambda x+\left(1-\lambda\right)y\right)=f\left(\sum_{i=1}^{n}a_{i}\left[\lambda x_{i}+\left(1-\lambda\right)y_{i}\right]\right)=f\left(\lambda\overbrace{\left(\sum_{i=1}^{n}a_{i}x_{i}\right)}^{x'}+\left(1-\lambda\right)\overbrace{\left(\sum_{i=1}^{n}a_{i}y_{i}\right)}^{y'}\right)$$

$$\overbrace{\leq}^{f\text{ is convex}}\quad\lambda f\left(x'\right)+\left(1-\lambda\right)f\left(y'\right)=\lambda F\left(x\right)+\left(1-\lambda\right)F\left(y\right)$$

■

נדבר עוד על פונקציות וקבוצות קמורות בתרגיל.

[5] אכן, נראה לא מעט דוגמאות כאלה

8.2 אופטימיזציה קמורה וגרדיינט דסנט (Gradient Descent)

המוטיבציה שלנו לדבר על פונקציות קמורות היא ביצוע אופטימיזציה קמורה עליהן. לשם כך נשתמש במשפחת האלגוריתמים Gradient Descent.
נתונה[6] פונקציה:

$$f : \mathbb{R}^n \to \mathbb{R}$$

רוצים למצוא $x \in \mathbb{R}^n$ הממזערת אותה, כלומר המקיים:

$$f(x) = \inf_{x' \in \mathbb{R}^n} f(x')$$

נשים לב שאם מתחילים ב־$x' \in \mathbb{R}^n$ מסויים, אזי הכיוון בו f קטנה הכי הרבה הוא $-\nabla f(x')$. לכן, טבעי להסתכל על אלגוריתם המתקדם בכיוון זה.

אינטואיציה אפשר לחשוב על f, אם נסתכל על הגרף שלה, כמעין איזור הררי, והדבר הכי "חכם" לעשות - ללכת כל פעם בצעד קטן שיקטין את הגובה שלנו (לא נרצה צעד גדול, כי אולי הכיוון משתנה בהמשך), עד שנגיע לנקודת המינימום.
מכאן האלגוריתם:

אלגוריתם 1 Gradient Descent

מתחילים ב־$x_0 \in \mathbb{R}^n = 0$.
בכל שלב מגדירים:

$$x_{t+1} = x_t - \eta_t \cdot \nabla f(x_t)$$

כאשר η_t יכול להיות תלוי ב־t ולא להיות קבוע.

ההצעה הזו מעלה הרבה שאלות:

- האם מובטחת התכנסות לאינפימום?
- מתי האלגוריתם עוצר?
- איך לבחור את t בו עוצרים?
- איך לבחור את η_t?

השאלות האלה הן בסיס לתת־חלק די גדול לשאלות בתחום אופטימיזציה קמורה, וזה תחום שלא נכנס אליו בקורס הזה[7]. לצורך השימוש שלנו, נניח הנחות כלשהן, ונראה שעבור משפחה עשירה של פונקציות קמורות, אפשר לבחור את η_t כך שמובטח לנו שהאלגוריתם יתכנס; את העבודה עצמה נעשה בתרגול. למשל, נשתמש ב־$\eta_t = \frac{1}{\sqrt{t}}$, וערך זה יספיק לנו. עם זאת, נצביע על שתי תכונות המייחדות פונקציות קמורות, שאולי מבהירות מדוע נצפה ש־GD יתכנס עבור פונקציות קמורות.

8.2.1 תכונות של פונקציות קמורות

למה קל לעשות אופטימיזציה דיסקרטית על פונקציות קמורות? קלות זו נובעת מהתכונות של הפונקציות האלה.

כל מינימום הוא גלובלי בפונקציה כללית יתכן ויש הרבה נק' מינימום מקומי, אך הן לא יספיקו לנו. בפונקציות קמורות לעומת זאת, כל נק' מינימום היא גלובלית, ולכן הנקודה שנמצא היא באמת מה שאנחנו מחפשים.
ננסח תכונה זו בלמה שתוכח או בתרגיל או בתרגול:

למה 8.7 (לפונ' קמורות כל מינימום לוקאלי הוא גלובלי) תהא $f : W \to \mathbb{R}$ קמורה, וניח ש־$x^* \in W$ מקיימת עבור $r > 0$:

$$\forall x \ s.t. \ \|x - x^*\| < r \ f(x) \geq f(x^*)$$

אזי:

$$\forall x \in W \ f(x) \geq f(x^*)$$

[6] מה זה אומר "נתונה"? לא ניתן לייצג פונקציה כזו בצורה סופית! נחשוב על כך שניתן לחשב את הנגזרת ואת $f(x)$ בצורה יעילה.
[7] עוד בנושא זה בקורסי אלגוריתמיקה מתקדמים, כגון אופטימיזציה קמורה.

קיום סאב גרדיינטים כדי להשתמש באלגוריתם GD, צריך להיות לפונקציה גרדיינט בכל נקודה. אבל פונקציות קמורות לא תמיד גזירות, ועל כן לא בכל נקודה יהיה קיים גרדיינט. מסתבר אבל שהפונקציות הללו "כמעט גזירות": קיימת להן תכונה דומה "לעניים" שתספיק לנו, והיא קיום סאב גרדיינטים.
לדוגמא ויזואלית, נביט ב־$|x|$, שהיא פונקציה קמורה אך לא גזירה.

הגדרה 8.8 תהא $f : W \to \mathbb{R}$, $W \subset \mathbb{R}^n$ פונקציה קמורה[8]. נאמר שוקטור $v \in \mathbb{R}^n$ הוא **סאב גרדיינט** ב־$x \in W$ (לאו דווקא יחיד!), ונסמן $\nabla f(x) = v$, אם מתקיים:

$$\forall y \in W \ f(y) \geq f(x) + \langle y - x, \nabla f(x) \rangle$$

הערה 8.9 לפעמים הסאב גרדיינט הוא כמו הגרדיינט, אבל צריך פה להזהר, כי סאב גרדיינט אינו יחיד!

הערה 8.10 מסמנים את אוסף הסאב גרדיינטים בנקודה x של f ב־$\partial f(x)$.

מבחינה גיאומטרית, המישור שהוא הגרף של:

$$x \mapsto f(x) + \langle y - x, \nabla f(x) \rangle$$

נמצא מתחת ומשיק לגרף של f.

למה 8.11 לכל פונקציה קמורה קיים סאב גרדיינט בכל נקודה.

מכיוון והתכונה הזו מתקיימת, ניתן להפעיל את האלגוריתם על פונקציות קמורות בשימוש בסאב גרדיינט (במקום בגרדיינט).
תרגיל: הוכיחו את הלמה עבור פונקציות מ־$\mathbb{R}$ ל־$\mathbb{R}$[9].

הערה 8.12 למעשה, הקשר בין סאב גרדיינטים לפונקציות קמורות הוא אף חזק יותר מהלמה - קיום סאב גרדיינטים הוא איפיון נוסף לקמירות (כמו שראינו בתרגול, פונקציה f היא קמורה אמ"מ לכל נקודה $x \in W$, $\partial f(x) \neq \phi$).

8.3 בעיות למידה קמורות

נרצה להשתמש באופטימיזציה קמורה כדי לפתור בעיות למידה. לצורך עניין זה, נרצה להבין מתי בעיית ה־ERM היא בעיה קמורה, או בעיה קמורה תחת רדוקציה מסויימת.
לפני שנגדיר בצורה פורמלית מתי בעיית למידה היא בעיית למידה קמורה, נראה דוגמא קונקרטית - **בעיית רגרסיה**:

$$\begin{aligned} &X = \mathbb{R}^n, \ Y = \mathbb{R} \\ &\mathcal{H} = \{h_w(x) = \langle w, x \rangle \,|w \in \mathbb{R}^n\} \\ &l(\hat{y}, y) = (\hat{y} - y)^2 \end{aligned}$$

נביט ב־$OPT(\mathcal{H})$ - בהנתן מדגם:

$$S = \{(x_1, y_1), ..., (x_m, y_m)\} \subset \mathbb{R}^n \times \mathbb{R}$$

רוצים למצוא $h_w \in \mathcal{H}$ הממזער את:

$$L_S(h_w) = \frac{1}{m} \cdot \sum_{i=1}^{m} l(h_w(x_i), y_i)$$

ולבעיה הקונקרטית שלנו:

$$= \frac{1}{m} \cdot \sum_{i=1}^{m} (\langle w, x_i \rangle - y_i)^2$$

[8] אין באמת צורך בהגדרה בפונקציה קמורה, אבל נשתמש בה רק לפונקציות קמורות.
[9] "התרגיל הזה לא טריוואלי, אבל אפשרי".

נדגיש שכאשר אנו מנסים למזער את $L_S(h_w)$, המדגם S הינו **קבוע**.
למרבה ההפתעה, מה שקיבלנו הוא פונקציה קמורה! זוהי תכונה ראשונה שנרצה לשמר. מדוע זוהי פונקציה קמורה? נשתמש במה שראינו בשיעור הקודם: יש לנו סכום, לכן מספיק להוכיח שכל אחד מהנסכמים הוא קמור. כל אחד כזה הוא:

$$l_{(x_i,y_i)}(w) := (\langle w, x_i\rangle - y_i)^2 = f\left(\sum_{j=1}^{n} x_i^j h^j\right)$$

כאשר :

$$f(z) = (z - y_i)^2$$

נשים לב כי f היא באמת פונקציה קמורה מהגדרתה, ולכן באמת השגיאה היא פונקציה קמורה.
תכונה נוספת שתעזור לנו - הפרמטריזציה של $\mathcal{H}$ היתה באמצעות קבוצה קמורה. כלומר, היתה לנו קבוצה קמורה W (במקרה שלנו, $W = \mathbb{R}^n$), והעתקה $w \mapsto h_w$ מ־W על $\mathcal{H}$.
במובן מסויים, עשינו רדוקציה למקרה בו מחלקת ההיפותזות היא קבוצה קמורה W, ופונקצית ההפסד $L_S(w)$ הינה קמורה. נקודת המבט הזו מובלה להגדרה הבאה:
כעת, נפרמל:

הגדרה 8.13 (בעיית למידה קמורה) נאמר ש־$(X, Y, \mathcal{H}, l)$ היא **קמורה**, אם קיימת קבוצה קמורה W והעתקה $w \mapsto h_w$ מ־W <u>על</u> $\mathcal{H}$, כך שלכל זוג $(x, y) \in X \times Y$, הפונקציה:

$$l_{(x,u)}(w) = l(h_w(x), y)$$

היא קמורה.

סימנים:

$$L_S(w) = L_S(h_w),\ L_{\mathcal{D}}(w) = L_{\mathcal{D}}(h_w)$$
$$l_{(x,y)}(h_w) = l_{(x,y)}(w) := l(h_w(x), y)$$

בתרגיל או בתרגול נראה דוגמא שמכלילה את הדוגמא שראינו מעלה - רגרסיה ריבועית.
השבוע ובשבוע הבא נראה שאם הבעיה היא בעיית למידה קמורה בעלת כמה תנאים נוספים, היא ניתנת ללמידה ביעילות. האלגוריתם שילמד אותה לא יהיה ERM, אבל דומה, ויהיה ניתן לחישוב בצורה די יעילה אם ניתן לחשב סאב גרדיינטים, ו־W "לא יותר מידי מוזרה".

8.4 אלגוריתמים יציבים ולמידות

אנחנו נראה אלגוריתם שמיועד ללמוד בעיות למידה קמורות, ונרצה לנתח את סיבוכיות המדגם שלו.
צורה אחת לעשות זאת - לפתח תורה דומה לתורת ההכללה. לא נעשה זאת, כי זה מורכב וארוך, ודי דומה למה שכבר עשינו. לכן נשתמש בגישה הרבה יותר פשוטה המתבססת על יציבות: נראה שהאלגוריתם שנגדיר הוא אלגוריתם יציב (מיד נראה הגדרה), ונראה שלאלגוריתמים יציבים יש סיבוכיות מדגם קטנה, ע"י כך שנראה קיום של שלוש תכונות:

1. **האלגוריתם יציב**. כלומר, אם משנים את המדגם במעט (למשל, מחליפים דוגמא אחת), הפלט לא משתנה בהרבה.
2. **אלגוריתמים יציבים לא עושים התאמת יתר** (overfit). באופן לא מדוייק, נאמר שאלגוריתם למידה עושה אוברפיט אם הוא מחזיר היפותזה ש"מתאימה מידי" למדגם. כלומר, יש לה שגיאה אמפירית מאוד קטנה, אך שגיאה אמיתית גדולה.
3. **האלגוריתם יחזיר היפותזה עם שגיאה אמפירית קטנה** ביחס ל־$L_S(\mathcal{H})$. במילים אחרות, הוא "כמעט" ERM.

הגדרה מסודרת ל"אלגוריתם יציב" נביא מיד.
נביא אינטואיציה להתאמת יתר - בציור! אלגוריתם שהוא "קו ישר", לעומת אלגוריתם שהיא "פונקציה עולה ויורדת" (כמו ההיפותזה שאלגוריתם רגרסייה מחזיר): בעוד שהאלגוריתם השני יכול להחזיר היפוטזה עם שגיאה אמפירית אפס, ברור כי הסיכוי לקבל היפוטזה שכזו הוא קטן. אנחנו נוכיח תיאורתית כי הסתברות גבוהה זה לא קורה.

הערה 8.14 נראה שנרצה שהאלגוריתמים שלנו לא יעשה "התאמת יתר" - זו לא נראית לנו תכונה של אלגוריתם טוב. מצד שני, גם אלגוריתם שלא עושה התאמת יתר לא בטוח שהוא טוב - למשל, האלגוריתם שמחזיר היפותזה קבועה; השגיאה האמפירית של ההיפותזה לא רחוקה מהשגיאה האמיתית שלה, עם זאת, בוודאי שאלגוריתם כזה בדר"כ יחזיר היפותזה עם שגיאה גבוהה.

הערה 8.15 התאמת יתר היא תכונה של האלגוריתם, ואין לה קשר למחלקת ההיפותזות.

היום נתרכז בהוכחת התנאי השני; כלומר, נראה שאלגוריתמים יציבים לא עושים אוברפיט.
מהו אלגוריתם יציב? אינטואיטיבית, נרצה ששינוי של נקודה אחת במדגם, תגרום לשגיאה "מאוד קטנה" בנקודה שהחלפנו ואיננה כבר במדגם - כלומר, זו מעין "הוכחה" לכך שהאלגוריתם "למד באמת" את $h^*(x)$. על כן נגדיר באופן פורמלי:

הגדרה 8.16 עבור פונקציה $\varepsilon : \mathbb{N} \to \mathbb{R}^+$, נאמר שאלגוריתם $\mathcal{A}$ הוא **ε-יציב**, אם לכל:

$$S = \{(x_1, y_1), ..., (x_m, y_m)\}$$

לכל $1 \leq i \leq m$ ולכל $(x, y) \in X \times Y$, מתקיים:

$$l_{(x_i,y_i)}\left(\mathcal{A}\left(S^i\right)\right) \leq l_{(x_i,y_i)}\left(\mathcal{A}\left(S\right)\right) + \varepsilon\left(m\right)$$

כאשר S^i מתקבל מ-S ע"י החלפת (x_i, y_i) ב-(x, y).

אינטואיטיבית נצפה שאלגוריתמים יציבים לא יעשו overfit, אולי בגלל הטיעון הבא: אם ביט באלגוריתם יציב כמו קו ישר, ויש לנו דוגמא כלשהיא ושגיאה עליה, נצפה שהשגיאה על הנקודה הזו תהיה פחות או יותר השגיאה האמפירית, ואז אם נחליף את הנקודה הזו גודל השגיאה "לא ישתנה בהרבה".

למה 8.17 (אלגוריתמים יציבים לא עושים התאמת יתר) אם $\mathcal{A}$ אלגוריתם ε-יציב, אז:

$$\mathbb{E}_{S\sim\mathcal{D}^m}\left[L_{\mathcal{D}}\left(\mathcal{A}\left(S\right)\right) - L_S\left(\mathcal{A}\left(S\right)\right)\right] \leq \varepsilon\left(m\right)$$

הוכחה: תהא $(x, y) \sim \mathcal{D}$, ויהא $i \sim Uni\{1, ..., m\}$ (דוגמא אקראית שבה נחליף את אחת הדגימות, ומשתנה שתפקידו יהיה לבחור את הדגימה האקראית אותה נחליף). אזי:

$$\begin{aligned}
&\mathbb{E}_{S\sim\mathcal{D}^m}\left[L_{\mathcal{D}}\left(\mathcal{A}\left(S\right)\right) - L_S\left(\mathcal{A}\left(S\right)\right)\right] \overset{def.}{=} \mathbb{E}_{S\sim\mathcal{D}^m}\left[\mathbb{E}_{(x,y)\sim\mathcal{D}} l_{(x,y)}\left(\mathcal{A}\left(S\right)\right) - L_S\left(\mathcal{A}\left(S\right)\right)\right] \\
&= \mathbb{E}_{S\sim\mathcal{D}^m,(x,y)\sim\mathcal{D}}\left[l_{(x,y)}\left(\mathcal{A}\left(S\right)\right) - L_S\left(\mathcal{A}\left(S\right)\right)\right] = \mathbb{E}_{S,(x,y),i}\left[l_{(x,y)}\left(\mathcal{A}\left(S^i\right)\right) - L_S\left(\mathcal{A}\left(S\right)\right)\right] \\
&= \mathbb{E}_{S,(x,y),i}\left[l_{(x_i,y_i)}\left(\mathcal{A}\left(S^i\right)\right) - l_{(x_i,y_i)}\left(\mathcal{A}\left(S\right)\right)\right] \leq \varepsilon\left(m\right)
\end{aligned}$$

כאשר ה"טריק" השיוויון השלישי חוקי כי ההתפלגות של $((x, y), S)$ זהה לזו של $\left(\left(x_i, y_i, S^i\right)\right)$. ■

[10]ציינו שנדרוש מבעיית הלמידה הקמורה מספר תנאים נוספים. כעת נגדיר אותם:

הגדרה 8.18 נאמר שפונקציה $f : C \to \mathbb{R}$ היא **ρ-ליפשיצית** כאשר $C \subset \mathbb{R}^n$, ומתקיים:

$$|f(x) - f(y)| \leq \rho \|x - y\|$$

לכל $x, y \in C$.

הגדרה 8.19 נאמר שבעיית למידה קמורה היא **ρ-ליפשיצית** אם לכל $X \times Y \ni (x, y)$, $l_{(x,y)}(w)$ היא ρ-ליפשיצית..
נאמר שהבעיה **$\mathbf{R}$-חסומה** אם W מוכלת בכדור ברדיוס R; כלומר לכל $W \ni w$, $\|w\| \leq R$.

הערה 8.20 אם הבעיה היא ρ-ליפשיצית, אז מפני שממוצע של פונקציות ρ-ליפשיציות הוא ρ-ליפשיצי, מתקיים שלכל מדגם S הפונקציה $L_S(w)$ הינה ρ-ליפשיצית. לכן, עבור בעיה כנ"ל, בעיית מזעור השגיאה האמפירית הינה בעיה של מזעור פונקציה קמורה ו-ρ-ליפשיצית.

[10]מכיוון ולא נכחתי בשיעור זה, החומר נלקח ישירות מהסיכום שהעלה מרצה הקורס, עם עריכות קטנות שלי.

8.5 למידות של בעיות קמורות

נזכור כי רצינו להראות אלגוריתם שיהיה יעיל תחת דרישות לא מחמירות, ומסוגל ללמוד בעיות למידה קמורות המוגבלות באופן כלשהוא - ההגבלה תנבע מכך שבעיות הלמידה אותן האלגוריתם ילמד יהיו חסומות וליפשיציות. לפני שנציג וננתח אותו, נראה שלא ניתן ללמוד בעיות קמורות כלליות.

8.5.1 מדוע יש לדרוש חסימות וליפשיציות?

כאמור, אנו נראה שבעיות חסומות וליפשיציות הן למידות. לפני כן, נעיר שבעיות קמורות כלליות אינן למידות. נביט בבעיית הלמידה הבאה:

$$X = [0,1]\,,\ Y = \mathbb{R},\ l(\hat{y}, y) = |\hat{y} - y|\,,\ \mathcal{H} = \{h_w(x) = w \cdot x |\, w \in \mathbb{R}\}$$

אנו נראה שלכל אלגוריתם $\mathcal{A}$, מתקיים $m_{\mathcal{A}}\left(1, \frac{1}{10}\right) = \infty$. כלומר, נראה שלא קיים $m > 0$ כך שלכל התפלגות $\mathcal{D}$ על $X \times Y$ מתקיים:

$$\underset{S\sim\mathcal{D}^m}{Pr}\left(L_{\mathcal{D}}(\mathcal{A}(S)) \leq L_{\mathcal{D}}(\mathcal{H}) + 1\right) \geq \frac{9}{10}$$

לשם פשטות, נראה זאת רק עבור אלגוריתמים דטרמיניסטיים. **הוכחה:** נניח בשלילה כי קיים אלגוריתם כזה. נסמן ב-$\overline{h}$ את ההיפוטזה אותה מחזיר האלגוריתם על מדגם בן m איברים שכל הדוגמאות בו הן $X \times Y \ni (0,0)$. נניח בה"כ ש-$\overline{h}(1) \leq 0$. נבחר μ קטן מספיק כך ש: $(1-\mu)^m \geq \frac{1}{2}$. נביט כעת בהתפלגות $\mathcal{D}$ המוגדרת באופן הבא:

$$\underset{(x,y)\sim\mathcal{D}}{Pr}((x,y) = (0,0)) = 1 - \mu,\ \underset{(x,y)\sim\mathcal{D}}{Pr}\left((x,y) = \left(1, \frac{2}{\mu}\right)\right) = \mu$$

לא קשה לראות ש-$L_{\mathcal{D}}(\mathcal{H}) = 0$ (שכן להיפותזה $h(x) = \frac{2}{\mu} \cdot x$ יש שגיאה 0). כעת, כאשר $S \sim \mathcal{D}^m$, בהסתברות:

$$(1-\mu)^m \geq \frac{1}{2}$$

כל הדוגמאות במדגם הן $(0,0)$, ולכן האלגוריתם יחזיר את $\overline{h}$. במקרה הזה יתקיים:

$$\begin{aligned} L_{\mathcal{D}}(\mathcal{A}(S)) &= L_{\mathcal{D}}\left(\overline{h}\right) \\ &= (1-\mu)\left|\overline{h}(0) - 0\right| + \mu\left|\overline{h}(1) - \frac{2}{\mu}\right| \\ &\geq \mu\left|\overline{h}(1) - \frac{2}{\mu}\right| \geq 2 > 1 + L_{\mathcal{D}}(\mathcal{H}) \end{aligned}$$

כלומר, קיבלנו שבהסתברות $> \frac{1}{2}$ מתקיים:

$$L_{\mathcal{D}}(\mathcal{A}(S)) > L_{\mathcal{D}}(\mathcal{H}) + 1$$

מצד שני, הנחנו שבהסתברות $\geq \frac{9}{10}$ מתקיים:

$$L_{\mathcal{D}}(\mathcal{A}(S)) \leq L_{\mathcal{D}}(\mathcal{H}) + 1$$

כלומר, הצבענו על שני מאורעות זרים שסכום ההסתברויות שלהם גדול מ-1, בסתירה. ■

8.6 למידה באמצעות Regularized Loss Minimization

תהא $(X, Y, \mathcal{H}, l)$ בעיית למידה קמורה ביחס לפרמטריזציה $w \mapsto h_w$. נניח שמרחב הפרמטרים נתון ע"י $W \subset \mathbb{R}^n$. נניח, כמו כן, שהבעיה הינה ρ- לפישיצית ו-R-חסומה.

הגדרה 8.21 נאמר שהאלגוריתם **מממש את כלל ה-RLM עם פרמטר רגולציה** $\lambda > 0$ אם הוא ממזער את הפונקציה:

$$L_S^{\lambda}(w) := L_S(w) + \lambda\|w\|^2 = \frac{1}{m}\sum_{i=1}^{m} l_{(x_i,y_i)}(w) + \lambda\sum_{j=1}^{n} w_j^2$$

על פני כל $w \in W$.

האפקט של הוספת גורם הרגולציה כלל ה־RLM דומה מאוד לכלל ה־ERM, ההבדל היחיד הוא ההוספה של **גורם הרגולציה** $\lambda \|w\|^2$. לתוספת הזו שני אפקטים:

- כפי שנראה, לתוספת תהיה השפעה "מייצבת", וככל ש־λ יהיה גדול יותר, האלגוריתם יהיה יציב יותר. קונקרטית, נראה שכלל ה־RLM הינו $\frac{2\rho^2}{\lambda m}$־יציב.
- מצד שני, כאשר λגדול, כלל ה־RLM יתרחק מכלל ה־ERM.

הנקודה הראשונה מעודדת אותנו לקבוע λ גדול, בעוד השניה מעודדת λ קטן. אנו נראה כיצד לקבוע את λ לפי m, כך שיתקבל אלגוריתם עם סיבוכיות מדגם קרובה לאופטימלית.

יעילות כפי שנראה בתרגול, כאשר ניתן לחשב ביעילות את $l_{(x,y)}w$ ואת $\nabla l_{(x,y)}(w)$ בהנתן (x,y) ו־w, וכאשר W היא קבוצה "יפה" (למשל כאשר W הוא המרחב $\mathbb{R}^n$ כולו או כדור במרחב), קיימים אלגוריתמים יעילים המממשים את כלל ה־RLM.

ניגש כעת להוכיח שכלל ה־RLM מאפשר ללמוד בעיות קמורות, ליפשיציות וחסומות. הלמה הראשונה, והעיקרית, מראה שכלל ה־RLM יציב:

למה 8.22 כלל ה־RLM הינו $\frac{2\rho^2}{\lambda m}$ יציב.

לפני שנוכיח את הלמה, נסיק ממנה מסקנה חשובה:

מסקנה 8.23 עבור כלל ה־RLM עם $\lambda = \sqrt{\frac{2\rho^2}{R^2 m}}$ מתקיים:

$$\mathbb{E}_{S\sim\mathcal{D}^m}\left[L_{\mathcal{D}}\left(\mathcal{A}\left(S\right)\right)\right] \leq L_{\mathcal{D}}\left(\mathcal{H}\right) + \rho R\sqrt{\frac{8}{m}}$$

כעת, נוכל לעשות את מה שכיוונו לו ־ בהנתן ששלושת התכונות מתקיימות, נראה את:

הקשר לסיבוכיות המדגם

עבור $\varepsilon > 0$, אם ניקח $m \geq \frac{32\rho^2 R^2}{\varepsilon^2}$ לפי המסקנה יתקיים:

$$\mathbb{E}_{S\sim\mathcal{D}^m}\left[L_{\mathcal{D}}\left(\mathcal{A}\left(S\right)\right)\right] \leq L_{\mathcal{D}}\left(\mathcal{H}\right) + \rho R\sqrt{\frac{8}{m}} \leq L_{\mathcal{D}}\left(\mathcal{H}\right) + \frac{\varepsilon}{2}$$

מאי שיוויון מרקוב עבור המשתנה המקרי $L_{\mathcal{D}}\left(\mathcal{A}\left(S\right)\right) - L_{\mathcal{D}}\left(\mathcal{H}\right)$, נקבע שבהסתברות לפחות $\frac{1}{2}$ על פני בחירת המדגם יתקיים:

$$L_{\mathcal{D}}\left(\mathcal{A}\left(S\right)\right) \leq L_{\mathcal{D}}\left(\mathcal{H}\right) + \varepsilon$$

במילים אחרות, $m_{\mathcal{A}}\left(\varepsilon, \frac{1}{2}\right) \leq \frac{32\rho^2 R^2}{\varepsilon^2}$. נשים לב:
החסם הנ"ל הדוק עד כדי קבוע במובן הבא: קיימות בעיות למידה קמורות שהן ρ־ליפשיציות ו־R־חסומות וקיים קבוע $c > 0$ כך שלכל אלגוריתם $\mathcal{A}$ מתקיים:

$$m_{\mathcal{A}}\left(\varepsilon, \frac{1}{2}\right) > c \cdot \frac{\rho^2 R^2}{\varepsilon^2}$$

בהמשך, נראה כיצד ניתן לקבל מהחסם הנ"ל אלגוריתם (טיפה שונה) המקיים:

$$m_{\mathcal{A}}\left(\varepsilon, \delta\right) \leq C \cdot \frac{\rho^2 R^2 \cdot \log\left(\frac{1}{\delta}\right)}{\varepsilon^2}$$

עבור קבוע אוניברסיטלי $C > 0$.

נורמה מול מימד בדר"כ ρ יהיה קבוע קטן. לכן, הגורם הדומיננטי החסם על סיבוכיות המדגם שקיבלנו הינו $\left(\frac{R}{\varepsilon}\right)^2$, בשונה ממה שקיבלנו עבור בעיות קלסיפיקציה, שם הגורם הדומיננטי היה $\frac{VC(\mathcal{H})}{\varepsilon^2}$.

על המקרה $W = \mathbb{R}^2$ הרבה פעמים הבעיה תהיה ρ-ליפשיצית, אבל מרחב הפרמטרים הטבעי יהיה $\mathbb{R}^n$ שאיננו חסום. במקרה זה, טיעון דומה לטיעון שמוכיח את המסקנה יראה שהאלגוריתם המממש את כלל ה-RLM עם פרמטר $\lambda = \sqrt{\frac{2\rho^2}{R^2 m}}$ (כלומר, ממזער את $L_S^\lambda(w)$ על פני $\lambda \in \mathbb{R}^n$) יקיים:

$$\mathbb{E}_{S\sim\mathcal{D}^m}\left[L_\mathcal{D}\left(\mathcal{A}\left(S\right)\right)\right] \leq L_\mathcal{D}\left(\mathcal{H}_R\right) + \rho R\sqrt{\frac{8}{m}}$$

כאשר:

$$\mathcal{H}_R = \left\{ h_w \middle|\, \|w\| \leq R \right\}$$

ציינו קודם כי ככל שנגדיל את λ, האלגוריתם יהיה יציב יותר. עם זאת, נשים לב שככל שמגדילים את R (או באופן שקול, מקטינים את λ), $\mathcal{H}_R$ תגדל גם כן, וכך יהיה סיכוי טוב יותר שהיא מכילה היפותזה "טובה", ולכן הגורם $L_\mathcal{D}^{hinge}\left(\mathcal{H}_R\right)$ קטן. לעומת זאת, הגורם השני, $\rho R\sqrt{\frac{8}{m}}$ יגדל.
נעיר שבפועל רצים על כמה ערכי λ ובוחרים את זה שהניב את ההיפותזה עם הביצועים הכי טובים. התהליך הנ"ל נקרא model-selection, ונדבר עליו יותר בפירוט בשבוע הבא.

כעת, לבסוף, נוכיח את המסקנה: **הוכחה:** יהי $w^* \in W$ וקטור המקיים $L_\mathcal{D}(w^*) = L_\mathcal{D}(\mathcal{H})$ (נניח לשם פשוטת שקיים כזה). מהלמה ומהמשפט שהוכחנו עבור אלגוריתמים יציבים נקבל שמתקיים:

$$\begin{aligned}
\mathbb{E}_{S\sim\mathcal{D}^m}\left[L_\mathcal{D}\left(\mathcal{A}\left(S\right)\right)\right] &\leq \mathbb{E}_{S\sim\mathcal{D}^m}\left[L_S\left(\mathcal{A}\left(S\right)\right)\right] + \frac{2\varrho^2}{\lambda m} \\
&\leq \mathbb{E}_{S\sim\mathcal{D}^m}\left[L_S\left(\mathcal{A}\left(S\right)\right) + \lambda\left\|\mathcal{A}\left(S\right)\right\|^2\right] + \frac{2\rho^2}{\lambda m} \\
&\leq \mathbb{E}_{S\sim\mathcal{D}^m}\left[L_S\left(w^*\right) + \lambda\left\|w^*\right\|^2\right] + \frac{2\rho^2}{\lambda m} \\
&= L_\mathcal{D}\left(w^*\right) + \lambda\left\|w\right\|^2 + \frac{2\rho^2}{\lambda m} \\
&\leq L_\mathcal{D}\left(w^*\right) + \lambda R^2 + \frac{2\rho^2}{\lambda m} \\
&= L_D\left(\mathcal{H}\right) + \rho R\sqrt{\frac{8}{m}}
\end{aligned}$$

כאשר אי השיוויון השלישי נובע מכך ש-$\mathcal{A}$ מממש את כלל ה-RLM, ואי השיוויון הרביעי נובע מכך שהבעיה הינה R-חסומה. ■

8.6.1 הוכחת למה 8.22

הלמה נכונה באופן כללי, אך לשם פשטות, נתצמצם למקרה בו הפונקציות $w \mapsto l_{(x,y)}(w)$ גזירות פעמיים.
מה גורם לכלל ה-RLM להיות יציב? ובחן, לפונקציה מהצורה $g(w) = f(w) + \lambda\|w\|^2$ עבור $f : W \to \mathbb{R}$ קמורה, יש את התכונה הבאה: g היא 2λ-**קמורה חזק**:
אחת הדרכים לאפיין פונקציה קמורה היא שלכל נקודה $w \in W$, אם מתרחקים מ-w באיזשהוא כיוון $e \in \mathbb{R}^n$ ובודקים את ערך הפונקציה, אז השיפוע לא קטן. כלומר, הנגזרת השניה של הפונקציה $\alpha(t) = f(w+te)$ הינה אי שלילית. ל-g יש את התכונה שלא רק שהשיפוע לא קטן, למעשה הוא עולה. קונקרטית, אם $\|e\| = 1$, אז $\forall t,\ \alpha''(t) \geq 2\lambda$.
מה לקמירות וליציבות? ובכן, אפשר לקבל איזשהי אינטואיציה מהתמונה השמאלית בסיכומי הקורס (פרבולואיד $x^2 + y^2$ - רואים שהנקודה הממזערת את הפונקציה, כלומר $(0,0)$, היא יציבה במובן הבא: ככל שמתרחקים ממנה, ערך הפונקציה גדל. זה לא נכון עבור הפונקציה הימנית ($f(x,y) = x^2$ - נראית כמו מתיחה של x^2 על פני שלושה מימדים) - שם כאשר מתרחקים מהממזער (שוב - $(0,0)$) בכיוון ציר ה-y, ערך הפונקציה לא גדל בכלל!
הטענה הבאה מראה שהתכונה הזו נכונה לכל g מהצורה הנ"ל:

טענה 8.24 תהא $f : W \to \mathbb{R}$ פונקציה קמורה. נניח ש-$w^* \in W$ ממזער את הפונקציה $g(w) = f(w) + \lambda\|w\|^2$. אזי, לכל $w \in W$:

$$g(w) \geq g\left(w^*\right) + \lambda\left\|w - w^*\right\|^2$$

הוכחה: נביט בפונקציה $\alpha(t) = g(w^* + te)$, כאשר $e = \frac{w-w^*}{\|w-w^*\|}$ ו־$0 \leq t \leq \|w - w^*\|$. מכיוון ו־$w^*$ ממזער את g, $t = 0$ ממזער את α, ולכן יתקיים:

$$\alpha'(0) \geq 0 \tag{1}$$

כמו כן, אם נסמן $\beta(t) = f(w^* + te)$ נקבל שמתקיים:

$$\begin{aligned}\alpha(t) &= \beta(t) + \lambda \|w^* + te\|^2 \\ &= \beta(t) + \lambda \|w^*\|^2 + 2\lambda t \langle w^*, e\rangle + \lambda t^2 \|e\|^2 \\ &\overset{\|e\|=1}{=} \beta(t) + \lambda \|w^*\|^2 + 2\lambda t \langle w^*, e\rangle + \lambda t^2\end{aligned}$$

ומכיוון ו־β קמורה (בדקו!), $\beta''(t) \geq 0$, ולכן:

$$\alpha''(t) = \beta''(t) + 2\lambda \geq 2\lambda$$

כעת, ממשפט טיילור[11] קיים $t \in [0, \|w - w^*\|]$ עבורו מתקיים:

$$\begin{aligned}g(w) = \alpha(\|w - w^*\|) &= \alpha(0) + \alpha'(0) \cdot \|w - w^*\| + \frac{\alpha''(t)}{2} \cdot \|w - w^*\|^2 \\ &\geq g(w^*) + \lambda \|w - w^*\|^2\end{aligned}$$

■

סופסוף אנחנו מוכנים להוכיח את הלמה 8.22: **הוכחה:** נקבע:

$$S \in (X \times Y)^m,\ (x, y) \in X \times Y,\ 1 \leq i \leq m$$

יהא w_S הממזער של $L_S^\lambda(w)$, ויהא w_{S^i} הממזער של $L_{S^i}^\lambda(w)$. צריך להראות ש:

$$l_{(x_i,y_i)}(w_{S^i}) \leq l_{(x_i,y_i)}(w_S) + \frac{2\rho^2}{\lambda m}$$

מכיוון ו־$l_{(x_i,y_i)}$ היא ρ־ליפשיצית, מתקיים:

$$l_{(x_i,y_i)}(w_{S^i}) \leq l_{(x_i,y_i)}(w_S) + \rho \|w_{S^i} - w_S\| \tag{2}$$

ולכן די להראות ש־$\|w_{S^i} - w_S\| \leq \frac{2\rho}{\lambda m}$. ואכן, מההגדרה מתקיים:

$$L_{S^i}^\lambda(w) = L_S^\lambda(w) + \frac{l_{(x,y)} - l_{(x_i,y_i)}(w)}{m} \tag{3}$$

עכשיו, מהטענה הקודמת:

$$L_S^\lambda(w_{S^i}) \geq L_S^\lambda(w_S) + \lambda \|w_{S^i} - w_S\|^2 \tag{4}$$

מכיוון ו־$l_{(x,y)},\ l_{(x_i,y_i)}$ הן ρ־ליפשיציות, מתקיים:

$$\begin{aligned}\frac{l_{(x,y)}(w_{S^i}) - l_{(x_i,y_i)}(w_{S^i})}{m} &\geq \frac{l_{(x,y)}(w_S) - \rho\|w_{S^i} - w_S\| + l_{(x_i,y_i)}(w_S) - \rho\|w_{S^i} - w_S\|}{m} \\ &= \frac{l_{(x,y)}(w_S) - l_{(x_i,y_i)}(w_S)}{m} - \frac{w\rho\|w_{S^i} - w_S\|}{m}\end{aligned} \tag{5}$$

[11] נזכיר שמשפט טיילו אומר שאם $f : [0,1] \to \mathbb{R}$ גזירה פעמיים ברציפות אז יש $\xi \in (0, a)$ עבורו:

$$f(a) = f(0) + f'(0)\, a + \frac{f''(\xi)}{2} a^2$$

לסיום, מכיוון ו־w_{S^i} ממזער את $L^{\lambda}_{S^i}$, חייב להתקיים:

$$\Rightarrow 0 \geq L^{\lambda}_{S^i}(w_{S^i}) - L_{S^i}(w_S) \overbrace{=}^{(3)} L_S(w_{S^i}) - L_S(w_S) + \frac{l_{(x,y)}(w_{S^i}) - l_{(x_i,y_i)}(w_{S^i})}{m} + \frac{l_{(x,y)}(w_S) - l_{(x_i,y_i)}(w_S)}{m}$$

$$\overbrace{\geq}^{(4),(5)} \overbrace{\lambda \|w_S - w_{S^i}\|^2}^{\text{claim}} - 2\overbrace{\frac{\rho \|w_{S^i} - w_S\|}{m}}^{\text{Lifshitz}} \Rightarrow \lambda \|w_S - w_{S^i}\|^2 \leq \frac{2\rho \|w_{S^i} - w_S\|}{m}$$

$$\Rightarrow \|w_S - w_{S^i}\| \leq \frac{2\rho}{\lambda m}$$

כנדרש! ■

8.7 מעבר לבעיות קמורות - ההופעה של הקושי החישובי

הדוגמאות העיקריות לבעיות למידה "טבעיות" שהן קמורות הן בעיות רגרסיה למיניהן. כלומר, בעיות בהן $Y = \mathbb{R}$ והמרחק בין האיברים ב־Y נמדד ע"י מדד מרחב (למשל $l(\hat{y}, y) = (\hat{y} - y)^2$ או $|\hat{y} - y|$). כלל ה־ERM נותן לנו אלגוריתם יעיל ואפקטיבי לבעיות כאלו.

עם זאת, אוסף הבעיות שנרצה לפתור מכיל הרבה מאוד בעיות שאינן בעיות רגרסיה. למשל, בעיות קלסיפיקציה אינן קמורות (למשל בגלל שבבעיות קלסיפיקציה $l_{(x,y)}$ מקבל רק את הערכים 0 ו־1, ואין פונקציה קמורה עם התכונה הזו). באופן כללי יותר, בעיות בהן הפלט הוא דיסקרטי אינן קמורות.

בהמשך השיעור ובשלושת השיעורים הבאים נתרכז בשיטות לתקוף בעיות קלסיפיקציה (ונדבר קצת על בעיות נוספות). נזכיר שבעית למידה $(X, Y, \mathcal{H}, l)$ נקראת **בעית קלסיפיקציה** אם Y היא קבוצה סופית, ו:

$$l(\hat{y}, y) = l_{0-1}(\hat{y}, y) = \begin{cases} 0 & \hat{y} = y \\ 1 & \hat{y} \neq y \end{cases}$$

למרבה הצער, רב רובן של הבעיות הללו הן בעיות קשות חישובית. כלומר, ככל הנראה, לא קיים אלגוריתם יעיל המסוגל לפתור אותן.

נעיר שהקושי הוא חריף מאוד: אפילו אם מובטח לנו ש־$L_{\mathcal{D}}(\mathcal{H}) = 0$ (כלומר אנו במקרה הפריד), לא קיים אלגוריתם המסוגל להחזיר היפותזה עם שגיאה הקטנה מ 0.49999 (שגיאה של 0.5 ניתן להשיג באופן טריוויאלי, למשל ע"י הטלת מטבע)! מקרה אחד יוצא דופן הוא לימוד חצאי מרחבים, שם קיים אלגוריתם יעיל עבור המקרה הפריד (כפי שראינו בתרגול). אבל אפילו עבור חצאי מרחבים, ללא הנחת הפרידות, הבעיה הופכת למאוד קשה: ככל הנראה, לא קיים אלגוריתם יעיל המסוגל להחזיר היפותזה עם שגיאה הקטנה מ־0.49999 אפילו אם מובטח לנו ש־$L_{\mathcal{D}}(\mathcal{H}) < 0.00001$. כלומר, אפילו אם קיים חצי מרחב עם שגיאה כמעט מושלמת, עדיין לא ניתן להחזיר היפותזה עם שגיאה לא טריוויאלית.

לאור מידע זה, אנו לא נפתח אלגוריתמים הפותרים את הבעיה, או אפילו משיגים פתרון מקורב, כי פשוט אין כאלו (ככל הנראה). ניתן לחלק את האלגוריתמים שבכל זאת נראה לשני סוגים:

- **תחליפים קמורים - Convex Surrogates**: שיטות בהן אנו "מחליפים" את הבעיה בבעיה קמורה, כך שלבעיה החלופית יש את התכונות הבאות:
  - (יעילות) התחליף ניתן לפתרון ביעילות.
  - (קשר למקור) אם משיגים שגיאה טובה בתחליף, ניתן לקבל שגיאה טובה גם במקור.
- **יוריסטיקות**. המשפחה השניה מכילה אלגוריתמים הפועלים על פי כלל טבעי ואינטואיטיבי, אך אין להם, לפחות כיום, בסיס תיאורטי מוצק. למרות החוסר בבסיס תיאורטי, הרבה פעמים יוריסטיקות עובדות בצורה טובה בפועל.

היום והשבוע נתרכז בתחליפים קמורים. בשבועיים שלאחר מכן נדבר על יוריסטיקות.

8.8 אלגוריתם ה־SVM

14.05.2015

עד כה, אם נחשוב על האלגוריתמים שראינו בתרגול, לא ראינו את החשיבות של כלל ה־RLM: ראינו את גרסאות של אלגוריתם SGD גם עבור ERM וגם עבור RLM. עולה השאלה, מדוע אם כך למדנו את הכלל הזה? היום נראה מדוע.

בתרגול גם ראינו גרסא ראשונה של אלגוריתם ה־SVM, Hard SVM, המוצא על מישור המפריד את הדוגמאות החיוביות מהשליליות עם שוליים (margin) מקסימליים:

אלגוריתם 2 Hard-SVM

קלט:

$$S = \{(x_1, y_1), ..., (x_m, y_m)\} \subset \mathbb{R}^n \times \{\pm 1\}$$

פלט: $w \in \mathbb{R}^n$ עם $\|w\|^2$ מינימלי המקיים

$$\forall i \; 1 - y_i \langle w, x_i \rangle \leq 0 \tag{6}$$

זהו אלגוריתם אופטימיזציה שקל למצוא לו פתרון[12] **אם** קיים על מישור מפריד. הבעיה: הוא לא תמיד קיים! למעשה, ברב הפעמים כל וקטור $w \in \mathbb{R}^n$ יפר לפחות אחד מבין האילוצים (6).

נניח שאנחנו במצב בו אין על מישור מפריד, ועדיין אנחנו רוצים בדרך כלשהיא להפריד. אלגוריתם ה־SVM (או Soft-SVM) מהווה אנלוג של Hard-SVM הפועל גם כאשר לא קיים על מישור מפריד. מה המשמעות של המצב? כל וקטור מפר לפחות את אחד מהאילוצים, ונרצה להגיע למצב בו האילוצים מופרים "כמה שפחות". כדי לעשות כך, נרצה לכמת את הדרך בה וקטור מפר אילוץ מסויים.

הגדרה 8.25 נסמן:

$$l^{\text{hinge}}_{(x_i,y_i)}(w) = (1 - y_i \langle w, x_i \rangle)_+$$

$$a_+ = \begin{cases} 0 & a \leq 0 \\ a & a > 0 \end{cases}$$

הערה 8.26 אינטואיטיבית, מספר זה מכמת "כמה" הוקטור w מפר את האילוץ ה־i; כאשר הוא 0, האילוץ שווה ל־0 כאשר האילוץ אינו מופר, וגדל לינארית ככל שהאילוץ מופר יותר.

כמו כן, נרצה לדעת "כמה" וקטור מסויים מפר של כל האילוצים. לשם כך נגדיר:

הגדרה 8.27 נסמן:

$$L^{\text{hinge}}_S(w) := \frac{1}{m} \sum_{i=1}^{m} l^{\text{hinge}}_{(x_i,y_i)}(w)$$

כעת, האלגוריתם ינסה למזער גם את $\|w\|^2$ וגם את $L^{\text{hinge}}_S(w)$. קונקרטית, האלגוריתם יקבל בתור פרמטר ערך $\lambda > 0$ וימזער את הביטוי:

$$L^h_S(\omega) + \lambda \|w\|^2$$

על פני $w \in \mathbb{R}^n$. נסכם:

אלגוריתם 3 (Soft-)SVM

קלט: מדגם $S = \{(x_1, y_1), ..., (x_m, y_m)\} \subset \mathbb{R}^n \times \{\pm 1\}$, פרמטר רגולציה λ
פלט: $w \in \mathbb{R}^n$ הממזער את:

$$L^{\text{hinge}}_S(\omega) + \lambda \|w\|^2 = \frac{1}{m} \sum_{i=1}^{m} (1 - y_i \langle w, x_i \rangle)_+ + \lambda \|w\|^2$$

[12] לא נראה פתרונות אפשריים, אך קיימים לו בספרות.

הערה 8.28 הסיבה לנורמה: התוספת מוסיפה רגולריזציה; נראה זאת בצורה פורמלית בהמשך (זה מקטין את מחלקת ההיפותזות ולכן מקטין את שגיאת ההכללה).

הערה 8.29 הסיבה להסתכלות על רמת ההפרה - זה הופך את הבעיה לחשיבה (אחרת הבעיה איננה פתירה).

8.8.1 הצגה כ־RLM וניתוח סיבוכיות מדגם וריצה

על מנת לנתח את אלגוריתם ה־SVM ולהבין כיצד לממשו, נראה שניתן להציגו בתור אלגוריתם המממש את כלל ה־RLM ביחס לבעיה קמורה מתאימה. לאחר שנעשה זאת, נוכל להשתמש בתיאוריה שפיתחנו בשיעור הקודם. לפני שנתחיל, נכליל קצת את ההגדרה של בעיית למידה.
נציין שתמיד הסתכלנו על היפותזות מהצורה::

$$x \to sign\left(h\left(x\right)\right),\ h : X \to \mathbb{R}$$

הרבה פעמים יהיה לנו נוח (בשביל הניתוח) להסתכל ישירות על הפונקציות h ולא $x \mapsto sign\left(h\left(x\right)\right)$, ובשביל לעשות את זה במסגרת של PAC, נעשה שינוי קל בפונקציית ה־$loss$:

$$l : \mathbb{R} \times Y \to \mathbb{R}^{+}$$

כלומר, פונקציית ההפסד תחזיר ערך ממש כלשהוא במקום ערך ב־$\{\pm 1\}$. נראה לכך שתי דוגמאות:

- פונקציית ההפסד הטיבעית:

$$l_{0-1}\left(\hat{y}, y\right) = \begin{cases} 0 & \hat{y} \cdot y > 0 \\ 1 & \hat{y} \cdot y \leq 0 \end{cases}$$

- פונקציית ה־hinge loss:

$$l^{\text{hinge}}\left(\hat{y}, y\right) = \left(1 - \hat{y} \cdot y\right)_{+}$$

נסתכל על בעיית הלמידה הבאה:

$$\begin{aligned} &\left(B_\rho, \{\pm 1\}, \mathcal{H}, l^h\right) \\ &B_\rho = \{\, x \in \mathbb{R}^n \,|\, \|x\| \leq \rho\} \\ &\mathcal{H} = \{\, h_w \,|\, w \in \mathbb{R}\} \end{aligned}$$

כלומר, נניח שכל הדוגמאות נמצאות בכדור B_ρ. נשים לב כי:

- הבעיה הינה קמורה! יתר על כן, אנו נראה שהיא אף ρ־ליפשיצית, וכמו כן ניתן לחשב ביעילות את $l^{\text{hinge}}_{(x,y)}\left(w\right),\ \nabla l^{\text{hinge}}_{(x,y)}\left(w\right)$
- לכן, ניתן לממש את כלל ה־RLM ביעילות.
- יתרה מזו, כלל ה־RLM הוא בדיוק אלגוריתם ה־SVM!

מהמשפט שהוכחנו בשיעור הקודם: אם נריץ את אלגוריתם ה־SVM עם $\lambda = \sqrt{\frac{2\rho^2}{Rm}}$, נקבל ש:
נקבל:

$$\begin{aligned} &E_{S\sim\mathcal{D}^m} L^{\text{hinge}}_{\mathcal{D}}\left(A\left(S\right)\right) \leq L^{\text{hinge}}_{\mathcal{D}}\left(\mathcal{H}_R\right) + \rho R\sqrt{\frac{8}{m}} \\ &\mathcal{H}_R = \{\, h_w \,|\, \|w\| \leq R\} \end{aligned} \tag{7}$$

זה עדיין לא מספיק טוב. מדוע החסם הזה לא מעניין אותנו? את ה־hinge loss אנחנו המצאנו. היינו רוצים אינדיקציה טובה יותר - או ליתר דיוק, מעניין אותנו $L^{0-1}_D\left(\mathcal{A}\left(S\right)\right)$. בעיה זו תטופל בלמה הבאה:

למה 8.30 לכל פונקציה $h : B_\rho \to \mathbb{R}$:

$$L_{\mathcal{D}}^{0-1}(h) \leq L_{\mathcal{D}}^{\text{hinge}}(h)$$

וכך אנחנו מקבלים משהו בעל משמעות - יחד עם אי שיוויון (7) מקבלים:

משפט 8.31 אם נריץ את אלגוריתם ה־SVM עם $\lambda = \sqrt{\frac{2\rho^2}{R^2 m}}$ נקבל ש:

$$\mathbb{E}_{S\sim\mathcal{D}^m}\left[L_{\mathcal{D}}^{0-1}(\mathcal{A}(S))\right] \leq \mathbb{E}_{S\sim\mathcal{D}^m}\left[L_{\mathcal{D}}^{\text{hinge}}(\mathcal{A}(S))\right] \leq L_{\mathcal{D}}^{\text{hinge}}(\mathcal{H}_R) + \rho R\sqrt{\frac{8}{m}}$$

נשים לב ש(שוב) ככל שאנחנו מגדילים את R (או, באופן שקול, מקטינים את λ) הגורם $L_{\mathcal{D}}^{\text{hinge}}(\mathcal{H}_R)$ קטן, שכן אנו משתמשים במחלקה יותר גדולה. לעומת זאת, הגורם השני יגדל.

ניגש להוכחת הלמה: **הוכחה:** נשים לב שמתקיים:

$$l_{0-1}(\hat{y}, y) \leq l^{\text{hinge}}(\hat{y}, y) \tag{8}$$

שכן שתי הפונקציות הן מהצורה $f(\hat{y}, y)$ עבור $f : \mathbb{R} \to \mathbb{R}^+$, כאשר:

$$l^{\text{hinge}}(\hat{y}, y) = f_{\text{hinge}}(\hat{y}, y),\ f_{\text{hinge}}(x) = (1-x)^+$$
$$l^{0-1}(\hat{y}, y) = f_{0-1}(\hat{y}, y),\ f_{0-1}(x) = \begin{cases} 0 & x \geq 0 \\ 1 & x < 0 \end{cases}$$

בציור - התנהגות של שתי הפונקציות בגרף - l^{0-1} הינה פונקציית מדרגה, בעוד הפונקציה השניה לינארית עד 1 ותמיד מעליה עד נקודה זו. כמו כן, לא קשה לוודא שלכל x:

$$f_{\text{hinge}}(x) \geq f_{0-1}(x)$$

מא"ש (8) מקבלים כי:

$$L_D^{0-1}(h) = E_{(x,y)\sim\mathcal{D}}\left[l_{0-1}(h(x), y)\right] \leq \mathbb{E}\left[l^{\text{hinge}}(h(x), y)\right] = L_D^{\text{hinge}}(h)$$

■

8.8.2 תחליפים קמורים (Convex Surrogates) - מעבר ל־SVM

נאמץ את נקודה המבט הבאה בנוגע לאלגוריתם ה־SVM ולקשר שלו לבעיה של לימוד חצאי מרחבים. התחלנו מבעיה שהיינו רוצים לפתור, אך היא איננה קמורה ואף מאוד קשה חישובית ולכן אין לנו שום סיכוי[13] לעשות זאת, אפילו לא בצורה מקורבת:

$$(B_\rho, \{\pm 1\}, \mathcal{H}, l_{0-1})$$

כאשר $\mathcal{H}$ היא אוסף של הפונקציונאלים הלינאריים. על מנת לתקוף אותה בכל זאת, **החלפנו** את פונקציית ההפסד וקיבלנו:

$$\left(B_\rho, \{\pm 1\}, \mathcal{H}, l^{\text{hinge}}\right)$$

למעבר הזה היו שתי תכונות עיקריות:

- זו בעיה קמורה שניתן לפתור ביעילות.

[13] בהתאם לידע שקיים היום.

- הלמה מעלה אומרת שאם אנחנו מקבלים פתרון טוב לבעיה הזו, אזי יהיה לנו פתרון טוב לבעיה המקורית, שכן $L_{\mathcal{D}}^{0-1}(h) \leq L_{\mathcal{D}}^{\text{hinge}}(h)$.

אנו נראה (בתרגול הבא) שאפשר להשתמש ברעיון הזה כדי לפתור בעיות נוספות, מעבר לקלסיפיקציה בינארית. שיטה זו נקראת "רלקסציה קמורה".

הצעד הראשון יהיה להגדיר פורמאלית מתי פונקציית ההפסד היא תחליף קמור לפונקציית הפסד אחרת. על מנת שנוכל לטפל במגוון רחב של בעיות, נגדיר זאת עבור פונקציות $l : \mathbb{R}^k \times Y \to \mathbb{R}^+$.

הגדרה 8.32 תחליף קמור לפונ' הפסד $l : \mathbb{R}^k \times Y \to \mathbb{R}^+$ היא פונ' $l_s : \mathbb{R}^k \times Y \to \mathbb{R}^+$ המקיימת:

- לכל $y \in Y$, הפונקציה $\hat{y} \mapsto l_s(\hat{y}, y)$ הינה קמורה.
- לכל $\hat{y} \in \mathbb{R}^k$, מתקיים $l(\hat{y}, y) \leq l_s(\hat{y}, y)$.

אנו נראה בתרגול ובתרגיל שהרבה פעמים החלפה של פונ' הפסד בתחליף קמור תניב בעיית למידה אמורה אותה ניתן לפתור ביעילות.

8.9 שיכונים במרחבים ממימד גבוה ושיטות גרעין

8.9.1 שיכונים במימד גבוה

בשעה זו נדבר על הרחבה ל־SVM. למען האמת, זו דרך להשתמש ב־SVM על מחלקת היפותזות גדולה הרבה מעבר ממחלקת הפונקציונלים הלינאריים עליה דיברנו עד כה.

איך נעשה את זה? הרעיון הבסיסי יחסית פשוט: ניקח את הנתונים שלנו, ונשכן אותם במרחב גבוה מהמקורי (נחליף כל נקודה בנקודה במרחב היותר גבוה), ונפעיל את SVM על סט הנתונים החדש.

אינטואיציה מדוע לנו בכלל לעבוד במרחב גבוה מהמקורי? לעיתים, ולמעשה לרב, אנחנו פועלים במקרה הבלתי פריד ־ כלומר, כל דוגמא תפר אילוצים מסויימים, ולא קיים על מישור מפריד. הרעיון הוא שאם נעבור למרחב גבוה יותר, שם כן נוכל למצוא על מישור מפריד, ואז ההיפותזה לבעיה המקורית תשמש בהיפותזה על המרחב החדש.

קונקרטית, נגדיר את השיכון:

$$\Psi : X \to \mathbb{R}^N$$

ונבצע את התהליך הבא:

אלגוריתם 4 SVM with a mapping Ψ

קלט: מדגם $S = \{(x_1, y_1), ..., (x_m, y_m)\} \subset X \times \{\pm 1\}$, פרמטר רגולציה λ.
אלגוריתם:

1. הרץ SVM עם פרמטר λ על המדגם:

$$\Psi(S) := \{(\Psi(x_1), y_1), ..., (\Psi(x_m), y_m)\}$$

וקבל $h_w : \mathbb{R}^N \to \mathbb{R}$ עבור $w \in \mathbb{R}^N$.

2. החזר את ההיפותזה:

$$h_w^{\Psi}(x) := h_w(\Psi(x))$$

האלגוריתם הזה ממזער את ההינג'־לוס על מחלקת ההיפתזות:

$$\mathcal{H}^{\Psi} := \left\{ h_w^{\Psi} \middle| \, w \in \mathbb{R}^N \right\}$$

ובאופן קונקרטי:

משפט 8.33 נניח ש־$\Psi\left(\mathbb{R}^N\right) \subset B_\rho$. אזי, אם נריץ SVM עם המיפוי Ψ ועם $\lambda = \sqrt{\frac{2\rho^2}{R^2 m}}$ נקבל:

$$\mathbb{E}_{S\sim\mathcal{D}^m}\left[L_{\mathcal{D}}^{0-1}\left(\mathcal{A}\left(S\right)\right)\right] \leq \mathbb{E}_{S\sim\mathcal{D}^m}\left[L_{\mathcal{D}}^{\text{hinge}}\left(\mathcal{A}\left(S\right)\right)\right] \leq L_{\mathcal{D}}^{\text{hinge}}\left(\mathcal{H}_R^\Psi\right) + \rho\sqrt{\frac{8}{m}}$$

משפט זה נובע מיידית מהמשפט 8.31.

איך לבחור נכון את Ψ? המיפוי בו נשתמש צריך להיות אחד כזה המתאים ספציפית לבעיה שלפנינו, וביכולתו לעשות את הבעיה (קרוב) לפרידה (עם זאת, גם מיפויים גנריים יכולים לעזור). נראה דוגמא על מנת להמחיש:

דוגמא מדוע זה עוזר לנו? ע"י בחירת מיפוי מתאים, ניתן לקבל מחלקות עשירות. למשל, אם $X = \mathbb{R}^2$, והמיפוי הוא:

$$\Psi\left(\overset{\in X}{(x_1, x_2)}\right) = \left(1, x_1, x_2 x_1^2, x_2^2, x_1 x_2\right)$$

טענה 8.34

$$\mathcal{H}^\Psi = \left\{ p : \mathbb{R}^2 \to \mathbb{R} \middle| \, p\text{is a polynomial of deg} \leq 2 \right\}$$

הוכחה: עבור $w \in \mathbb{R}^6$ מתקיים:

$$\begin{aligned} h_w^\Psi(x) &= h_w\left(\Psi\left(x\right)\right) = \langle w, \Psi\left(x\right)\rangle \\ &= \langle (w_1, ..., w_6), (1, x_1, x_2, ..., x_1 x_2)\rangle \\ &= w_1 + w_2 x_1 + w_3 x_2 + w_4 x_1^2 + w_5 x_2^2 + w_6 x_1 x_2 \end{aligned}$$

מכאן h_w^Ψ הוא פולינום מדרגה ≥ 2 וע"י קביעה מתאימה של כל w ניתן לקבל כל פולינום. ■

לסיכום, ע"י שימוש בשיכון Ψ ומעבר ל־$\mathcal{H}^\Psi$ הגדלנו את מחלקת ההיפותזות שלנו מפונקציונלים לינאריים (פולינומים מדרגה 1) לפולינומים מדרגה 2. למשל, כעת יש לנו במחלקת ההיפותזות הממפות פנים של אליפסה ל־1 ואת החוץ שלה ל־(-1)[14], ולכן שגיאת הקירוב בהתפלגות המתוארת בתמונה למעלה הינה טובה. כמובן, ניתן להרחיב את הדוגמא הנ"ל ולקחת $X = \mathbb{R}^n$ עבור $n \geq 2$ ופולינומים מדרגה גבוהה יותר

מכשולים בדרך לפעמים השיכון הוא למימד מאוד גדול, נגיד n^d. זה מעלה לנו לנו שתי בעיות:

1. **הכללה.** לחצאי מרחבים במימד גבוה יש מימד VC גדול, לכן שימוש נאיבי בשיטה יצריך מספר דוגמאות גדול. את הפתרון לבעיה זו נקבל "בחינם" בעת הפעלת SVM ע"י בחירה של פרמטר רגולציה נכון - על ידו נוכל לשלוט בשגיאת ההכלה גם במרחבים במימד גבוה. ואכן, ראינו שבעזרת בחירה נכונה של λ ניתן לחסום את סיבוכיות המדגם ביחס לנורמה, ללא תלות במימד.
2. **חישוב.** כאשר N מאוד גדול, יהיה לנו יקר וקשה מבחינה חישובית לעבוד על $\mathbb{R}^N$ (במיוחד כש־N הוא אינסופי). נפתור זאת באמצעות שימוש בפונקציות הנקראות **גרעינים** (**kernels**).

8.9.2 שיטות גרעין

האבחנה הבסיסית מאחורי הטכניקה היא מאוד פשוטה: הרבה פעמים אנחנו לא צריכים באמת לדעת לעבור על N גדול, אלא רק לדעת לחשב את המכפלה הפנימית בין $\Psi\left(x\right)$ ל־$\Psi\left(x'\right)$, וברגע שיודעים לעשות זאת ביעילות, ניתן להשתמש ב־SVM.

לאור האבחנה הזו, נגדיר:

הגדרה 8.35 הגרעין של Ψ היא הפונקציה $k : X \times X \to \mathbb{R}$ המוגדרת ע"י:

$$k\left(x, x'\right) := \langle \Psi\left(x\right), \Psi\left(x'\right)\rangle$$

[14] שהרי אליפסה נתונה ע"י פולינום מדרגה 2.

מה צריך לדעת כדי לעבוד עם SVM בצורה יעילה?

- **(יצוג)** לייצג $w \in \mathbb{R}^N$ באופן קומפקטי.
- **(הערכה)** בהנתן (ייצוג של) $w \in \mathbb{R}^N$ ו־$x \in X$, להעריך ביעילות את $h_w^\Psi(x)$.
- **(צעד גרדיינט)** בהנתן (ייצוג של) $w \in \mathbb{R}^N$, $\eta > 0$, ודוגמא (x, y), לחשב ביעילות את היצוג של $w - \eta \nabla l^{\text{hinge}}_{(\Psi(x_i), y_i)}(w)$.

נסביר כיצד ניתן לבצע את הפעולות הללו בהנתן גרעין k כנ"ל.

יצוג מפרידים לינאריים ב־$\mathbb{R}^N$ ניצג מפרידים לינארים ב־$\mathbb{R}^N$ באמצעות צירוף לינארי של נקודות במדגם. כלומר, אנו נעבוד רק עם וקטורים מהצורה:

$$w = \alpha_1 \cdot \Psi(x_1) + ... + \alpha_m \cdot \Psi(x_m)$$

כלומר רק עם צירופים לינאריים של $\Psi(x_1), ..., \Psi(x_m)$ (כאשר $x_1, ..., x_m$ מהמדגם S). במקרה זה, מספיק לנו לשמור את $\alpha_1, ..., \alpha_m$ ואת הדוגמאות. לכל דוגמא פשוט נשמור את הסקלרים הרלוונטיים.

הערכת מפרידים לינאריים ב־$\mathbb{R}^N$ בהנתן יצוג כנ"ל של וקטור $w \in \mathbb{R}^N$, במהלך שלב האימון ובודאי גם אח"כ, נצטרך להעריך, בהנתן $x \in X$, את $h_w^\Psi(x)$. נשים לב שמתקיים:

$$h_w^\Psi(x) = \langle w, \Psi(x) \rangle = \left\langle \sum_{i=1}^m \alpha_i \Psi(x_i), \Psi(x) \right\rangle = \sum_{i=1}^m \alpha_i \langle \Psi(x_i), \Psi(x) \rangle = \sum_{i=1}^m \alpha_i k(x_i, x) \qquad (9)$$

לכן אם נוכל להעריך את הגרעין ביעילות, נוכל גם להעריך את $h_w^\Psi(x)$.

צעדי גרדיינט הפעולה האחרונה שעלינו לדעת ע"מ לממש GD או SGD היא לחשב צעדי גרדיינט. כלומר, בהנתן יצוג $(\alpha_1, ..., \alpha_m)$ של וקטור $w \in \mathbb{R}^N$ ודוגמא $(x_i, y_i) \in X \times Y$, עלינו לדעת לחשב את הייצוג של $\nabla l^h_{(\Psi(x_i), y_i)}(w)$ כצירוף לינארי של $\Psi(x_1), ..., \Psi(x_m)$. אבל, נזכור שראינו:

$$\nabla l^h_{(\Psi(x_i), y_i)}(w) = \begin{cases} -y\Psi(x_i) & y_i \langle w, \Psi(x_i) \rangle = y_i h_w^\Psi(x) < 1 \\ 0 & y_i \langle w, \Psi(x_i) \rangle = y_i h_w^\Psi(x_i) \geq 1 \end{cases}$$

לכן אנחנו רק צריכים לדעת לחשב את $\langle w, \Psi(x_i) \rangle$, ואת זה אנחנו יכולים לעשות אם יודעים לחשב את הגרעין, כפי שראינו מעלה.

הוקטורים התומכים

לאחר שלב האימון, כבר אין לנו צורך לעשות צעדי גרדיינט. נשים לב שעל מנת להעריך את הביטוי (9), די לנו לשמור בזיכרון את המקדמים $\alpha_1, ..., \alpha_m$ **שאינם 0** ואת הדוגמאות המתאימות. הרבה פעמים יש מעט מקדמים כאלה, דבר המאפשר חיסכון משמעותי בזיכרון ובזמן שלוקח להעריך את ההיפותזות. הוקטורים המתאימים למקדמים הללו (כלומר, המיפוי של הדוגמאות המתאימות במרחב $\mathbb{R}^N$) נקראים **הוקטורים התומכים** (**support vectors**).

8.9.3 דוגמאות לגרעינים

על מנת להשתמש בפרדיגמה שתיארנו מעלה, עלינו להכיר מספר גרעינים. נצביע על שניים פופולריים.

הגרעין הפולינומי נקבע n המימד המקורי של הדוגמאות, $d \geq 1$ הדרגה של הפולינומים שנרצה לקבל, ונביט במיפוי: $\Psi : \mathbb{R}^n \to \mathbb{R}^N$ המוגדר באופן הבא:

$$\Psi(x) = (x_{j_1} \cdot ... \cdot x_{j_d})_{(j_1, ..., j_d) \in \{0, ..., n\}^d}$$

ונגדיר $x_0 := 1$ ־ זה פשוט סימון. הסיבה להוספה היא טכנית ־ ע"מ לקבל מונומים שהם מדרגה קטנה או שווה ל־d. כלומר, $\Psi(x)$ הינו וקטור בן $N = (n+1)^d$ קואורדינטות. באופן דומה למה שראינו עבור $n = 2$, המיפוי הנ"ל מגדיר את $\mathcal{H}^\Psi$ להיות מרחב כל הפולינומים מדרגה $\leq d$:

$$\mathcal{H}^\Psi = \{\text{polynomials of deg} \leq \text{d}\}$$

טענה 8.36

$$k(x,x') := \langle \Psi(x), \Psi(x') \rangle = (\langle x,x' \rangle + 1)^d$$

הוכחה:

$$\begin{aligned}(\langle x,x' \rangle + 1)^d &= \left(\sum_{i=1}^{n} x_i x'_i + 1 \right)^d = (x_1 x'_1 + x_2 x'_2 + ... + x_n x'_n + 1)^d \\ &= \sum_{(j_1,...,j_d) \in \{0,..,n\}^d} \left(x_{j_1} x'_{j_1}\right) \cdot ... \cdot \left(x_{j_d} x'_{j_d}\right) \\ &= \sum_{(j_1,...,j_d) \in \{0,..,n\}^d} (x_{j_1} \cdot ... \cdot x_{j_d}) \cdot ... \cdot \left(x'_{j_1} \cdot ... \cdot x'_{j_d}\right) \\ &= \langle \Psi(x), \Psi(x') \rangle\end{aligned}$$

■

נשים לב שעבור d גדול, המרחב $\mathcal{H}^\Psi$ מאוד גדול. עם זאת, זוהי אינה סתירה ל"אין ארוחות חינם, שכן בפועל, פרמטר הרגולריזציה מוודא שאנחנו פועלים על חלק קטן מאוד מהמרחב[15].

הגרעין הגאוסי (או, גרעין RBF) המיפוי $\Psi : \mathbb{R}^n \to \mathbb{R}^\infty$ מוגדר ע"י:

$$\Psi(x) = \frac{e^{-\frac{1}{2}\|x\|^2}}{\sqrt{d!}} \left(x_{j_1} \cdot ... \cdot x_{j_d}\right)_{(j_1,...,j_d) \in \{1,...,n\}^d,\ d \in \{0,1,....\}}$$

נתעלם מהעובדה שהטווח של Ψ הוא אינסוף מימדי (נעיר שניתן לטפל בזה באופן פורמלי באמצעות התורה של מרחבי הילברט). עבור מיפוי זה, $\mathcal{H}^\Psi$ מכילה את כל הפונקציות מהצורה $e^{-\frac{1}{2}\|x\|^2} p(x)$ עבור פולינום $p : \mathbb{R}^n \to \mathbb{R}$ מדרגה כלשהיא (למעשה היא מכילה עוד פונקציות). נחשב את הגרעין עבור וקטורים $x, x' \in \mathbb{R}^n$ - מתקיים:

$$\begin{aligned}e^{-\frac{1}{2}\|x-x'\|^2} &= e^{-\frac{1}{2}\|x\|^2 - \frac{1}{2}\|x'\|^2} e^{\langle x,x' \rangle} \\ &= e^{-\frac{1}{2}\|x\| - \frac{1}{2}\|x'\|^2} \sum_{d=0}^{\infty} \frac{\langle x,x' \rangle^d}{d!} \\ &= e^{-\frac{1}{2}\|x\| - \frac{1}{2}\|x'\|^2} \sum_{d=0}^{\infty} \frac{1}{d!} \sum_{(j_1,...,j_d) \in \{1,...,n\}^d} \left(x_{j_1} x'_{j_1}\right) \cdot ... \cdot \left(x_{j_d} x'_{j_d}\right) \\ &= \sum_{d=0}^{\infty} \sum_{(j_1,...,j_d) \in \{1,...,n\}^d} \left(\frac{e^{-\frac{1}{2}\|x\|^2}}{\sqrt{d!}} x_{j_1} x'_{j_1}\right) \cdot ... \cdot \left(\frac{e^{-\frac{1}{2}\|x'\|^2}}{\sqrt{d!}} x_{j_d} x'_{j_d}\right) \\ &= \langle \Psi(x), \Psi(x') \rangle \\ \Rightarrow k(x,x') &= e^{-\frac{1}{2}\|x-x'\|^2}\end{aligned}$$

שוב נעיר כי המרחב שקיבלנו מאוד גדול, ולמעשה אוניברסלי במובן שהוא יכול לקרב כל פונקציה רציפה $f : \mathbb{R}^n \to \mathbb{R}$, ובעזרת פרמטר הרגוליזציה משתמשים בפועל בחלק קטן מאוד מהמרחב הזה.

8.10 ולידציה ובחירת מודל

כאשר אנו עובדים SVM, עלינו לבחור λ. לבחירה זו שתי השפעות, עם השלכות שונות לגבי הבחירה. כדי להבין את ההשפעה שלו, נזכר בחסם:

$$L^{0-1}_{\mathcal{D}}\left(\hat{h}\right) \leq L^h_{\mathcal{D}}(\mathcal{H}_R) + \rho R \sqrt{\frac{8}{m}}$$

כאשר היחס בין λ ל-R הפוך. אזי:

[15] על פולינומים מדרגות נמוכות, ולא "באמת" על פולינומים מדרגות גבוהות (יהיו להם מקדמים מאוד מאוד קטנים ואז הם לא ישפיעו כמעט בחישוב הסופי)

- אם λ גדול, מצטמצמים למחלקה קטנה של היפותזות, ולכן שגיאת ההכללה תהיה קטנה.
- מצד שני, אם λ קטן, אזי עובדים עם מחלקה מאוד גדולה של היפוטזות, ולכן שגיאת הקירוב תהיה קטנה. זה נכון במיוחד כאשר משתמשים בגרעינים - למשל אם משתמשים בגרעין אוניבברסלי (כמו למשל הגרעיו הגאוסי), ע"י בחירה של $\lambda > 0$ מספיק קטנה נוכל להתקרב **לכל** פונקציה.

נשאלת השאלה איך לקבוע את λ בצורה נכונה? א-פריורית, בהתחלה לא נדע מהי. לכן בדר"כ "נרוץ על כמה ערכים ונבחר את הטוב ביותר".
לפני שנראה כיצד לבחור את λ, נביט בגרף של שגיאת ההכלה מול שגיאת המדגם, כאשר ציר ה-x יהיה $\frac{1}{\lambda}$ $(\sim R)$. נשים לב שהציור רלוונטי לגרעין אוניברסלי - עבור גרעינים בעלי מימד סופי, המימד כבר בעצמו מגדיר איזשהוא פרמטר אליו השגיאה $L_{\mathcal{D}}$ תשאף (שאיננו ∞).
אחת הדרכים הפופולריות ביותר לבחירת מודל היא הדרך הבאה:

אלגוריתם 5 Model Choice

קלט: מדגם $S \in (X \times Y)^m$, $m_1 + m_2 = m$ (חלוקה של המדגם לשניים) , k (מספר הפרמטרים λאותם נרצה לבדוק).
אלגוריתם:

1. חלקו את S ל-$S_1 = S_{train}, S_2 = S_{validation}$ בגודל m_1, m_2 בהתאמה.
2. לכל $\lambda \in \left\{1, \frac{1}{2}, \frac{1}{4}, ..., 2^{-k}\right\}$, הריצו את SVM על S_1.
3. מבין k ההיפוטזות שהתקבלו, החזירו את h_λ עם $L_{S_2}(h)$ הנמוכה ביותר.

בדר"כ משתמשים ב-$k \sim 15$.
השיטה הזו מאוד כללית - כתבנו כאן ל-SVM, אבל אפשר להשתמש בה לבעיות אחרות, למשה לכל סיטואציה בה יש לנו סדרה (או רצף) של מחלקות $\mathcal{H}_1 \subset \mathcal{H}_2 \subset ...$. למשל בתרגיל נראה כיצד לעשות בחירת מודל עבור רגרסייה כאשר $\mathcal{H}^d$ הוא מרחב הפולינומים מדרגה d.
עד כאן הנושא של SVM, ואיך להשתמש בבעיות למידה קמורות כדי לפתור בעיות כמו קלסיפיקציה שאינם בהכרח קמורות. בתרגול ובתרגיל הבא נראה דוגמאות עשירות יותר.
בשיעור הבא, נדבר על יוריסטיקות.

9 אלגוריתמים יוריסטיים

הרבה מהשיטות האלה עובדות טוב בפועל, למרות שאנחנו לא תמיד יודעים לנתח אותם כמו שצריך. נדבר על כמה אלגוריתמים:

- השכן הקרוב
- עצי החלטה
- משפחה שנלמד יותר לעומק היא רשתות נוירונים, שהיא שיטה שהרבה פעמים עובדת מאוד טוב, והיא ב-cutting edge של הרבה מחקרים כיום.

9.1 אלגוריתם השכן הקרוב

זוהי הדוגמא הכי פשוטה. זהו האלגוריתם היחיד שלא יהיה בפריים של מחלקת היפותזות וכו', פשוט נסביר מה הוא עושה.
נתון X מרחב מטרי. מהי המשמעות של מרחב מטרי[16]? קיום מטריקה - פונקציה $d : X \times X \to \mathbb{R}^+$ המקיימת:

- $d(x,x) = 0$
- $d(x,y) \leq d(x,z) + d(z,y)$
- $d(x,y) = d(y,x)$

[16] מתמטיקאים, אתם אמורים כבר לדעת אותה בע"פ בשלב הזה :)

אלגוריתם 6 Nearest Neighbor

קלט: מדגם

$$S = (x_1, y_1), ..., (x_m, y_m)$$

פלט:

$$h : X \to \{\pm 1\}$$
$$h(x) = \textbf{the label of } x_i \textbf{that minimizes } d(x, x_i)$$

מה שהרבה פעמים עושים - לא מסתכלים על השכן הקרוב ביותר, אלא על קבוצת k השכנים הטובים ביותר, ומתייחסים להכרעת הרב שלהם (זאת כדי למנוע השפעה חזקה מידי של "יוצאי דופן").

9.2 רשתות נוירונים

בשבוע שעבר עזבנו את השיטה של שימוש בבעיות קמורות כדי לפתור את הבעיות שלנו, ועברנו ליוריסטיקות שלא תמיד מבינים למה הן פועלות.
מהו נוירון?

הגדרה 9.1 נוירון הוא יחידה חישובית לחישוב פונקצוית $\mathbb{R}^n \to \mathbb{R}$. יחידה זו מורכבת מ:

- n צמתי קלט.
- צומת שאחראית על הפלט.
- צמתי הקלט מחוברים לצומת של הפלט ע"י קשתות.
- פונקציית אקטיבציה: $\sigma : \mathbb{R} \to \mathbb{R}$, למשל פונקציית הסימן.

איך נוירון מחשב פונקציה? הוא צריך משקלים על הקשתות מצמתי הקלט:

$$w \in \mathbb{R}^n,\ x \mapsto \sigma(\langle w, x \rangle)$$

אנחנו לא רוצים לדבר על נוירון בודד, אלא רשת נוירונים.
רשת נוירונים מתקבלת מחיבור בין הרבה נוירונים. אנחנו נדבר על רשתות נוירונים שכבתיות, כלומר הנוירונים מחוברים בשכבות (דוגמא - שקופית 5), וכן נדבר רק על רשתות שהן feedforward, כלומר הקישורים בין הנוירונים הם רק משכבה אחת לשכבה הבאה (בלי קפיצות בין שכבות, בלי חיבור אחרונית, וכו').
השכבה הראשונה נקראת **שכבת הקלט**, האחרונה **שכבת הפלט**, והשכבות בין לבין נקראות **שכבות נסתרות**.
נגדיר באופן פורמלי:

הגדרה 9.2 רשת נוירונים היא זוג $\mathcal{N} = (G, \sigma)$, כאשר:

- $\sigma : \mathbb{R} \to \mathbb{R}$ היא פונקציית אקטיבציה.
- $G = (V, E)$ גרף מכוון בעומק d עם פלט יחיד, כלומר:
  - $V = \dot{\bigcup}_{t=0}^{d} V_t$
  - כל הצלעות הן מ-V_{t-1} ל-V_t
  - צומת פלט יחידה: $|V_d| = 1$

טרמינולוגיה

- **רוחב הגרף** הוא מספר הקודקודים בכל שכבה פנימית (כלומר לא שכבת הקלט או הפלט)
- גודל הגרף $|\mathcal{N}| = |E|$
- צמתי הקלט הם V_0, וגודל הקלט בהתאם הוא $n = |V_0|$

דוגמא לרשת נוירונים: רשת קשירה לחלוטין בעלת קלט n, רוחב l, עומק d, ופונקציית אקטיבציה σ ברשת - מסומנת בתור: $\mathcal{N}_{l,d,\sigma} = (G, \sigma)$, כאשר:

- $V = \dot{\bigcup}_{i=0}^{d} V_t$
- $\forall 1 \leq t \leq d-1 \;\; |V_t| = l$
- $|V_0| = n$
- E מורכב מכל הרשתות מ-V_{t-1} ל-V_t.

דוגמא בציור - שקופית 7.

כדי שהרשת תחשב פונקציה, כמו במקרה של הנוירון הבודד, צריך להגדיר משקלים על הקשתות. בהנתן רשת $\mathcal{N}$ עם קלט בגודל n, נקבע פונק' משקולות $w : E \to \mathbb{R}$, כאשר הרשת מחשבת את

$$h_{\mathcal{N},w} : \mathbb{R}^n \to \mathbb{R}$$

והערך $h_{\mathcal{N},w}(x)$ מתקבל ע"י העברת x ברשת. דוגמא עם $\sigma = sign$ - שקופית 8.

עכשיו יש לנו דרך לתאר את הפונקציות מ-$\mathbb{R}^n$ ל-$\mathbb{R}$, ונרצה אלגוריתם שיחזיר לנו רשת נוירונים טובה. אנחנו ניקח רשת $\mathcal{N}$ ונקבע אותה, וננסה למצוא משקולות טובים לרשת הזו - זו תהיה מחלקת ההיפותזות $h_{\mathcal{N},w}$. כדי לנתח את האלגוריתמים, נגדיר את המחלקה המתאימה:

$$\mathcal{H}_{\mathcal{N}} = \{h_{\mathcal{N},w} | w : E :\to \mathbb{R}\}$$

וכעת נרצה ללמוד את שגיאת ה-estimation, שגיאת ה-approximation, ו-optimization error, וזה מה שננסה להבין בשיעור היום.

9.2.1 סיבוכיות המדגם

נקבע לנו רשת $\mathcal{N} = (G, \sigma)$. אבחנה אחת פשוטה: על מנת להגדיר פונקציה במחלקה המוגדרת ע"י הרשת, צריך לקבל רשימה של מספרים בגודל הרשת: כלומר מספר w_i. דהיינו, מספר הפרמטרים הוא גודל הרשת - $|\mathcal{N}|$.

נזכר בתוצאה שראינו בתחילת הקורס: אם כל אחד מהמשקולות מיוצג ע"י k ביטים, נקבל את החסם: $VC(\mathcal{H}_{\mathcal{N}}) \leq k \cdot |\mathcal{N}|$.

לחסם יותר טוב:

משפט 9.3 עבור $\sigma = sign$:

$$VC(\mathcal{H}_{\mathcal{N}}) \leq C \cdot |\mathcal{N}| \cdot \log(|\mathcal{N}|)$$

לא נוכיח זאת, ההוכחה נמצאת ברשימות של שי שלו-שוורץ למתעניינים.

כמו כן:

משפט 9.4 עבור $\sigma = SIGMOID$:

$$VC(\mathcal{H}_{\mathcal{N}}) \leq C \cdot |\mathcal{N}|^2$$

מסקנה 9.5 סיבוכיות המדגם חסומה ע"י גודל הרשת.

הערה 9.6 אפשר להקטין עוד יותר את סיבוכיות המדגם ע"י רגולריזציה, כמו למשל ב-SVM, אם חוסמים את גודל הוקטור מקבלים חסם יותר טוב (בהתאם לנורמה שלו).

9.2.2 אילו פונקציות ניתן לחשב בעזרת רשת נוירונים?

בשביל פישוט, נתמקד בקלטים בוליאניים ו־$\sigma = sign$. אילו פונקציות מ־$\{\pm 1\}^n$ ל־$\{\pm 1\}$ הן ב־$\mathcal{H}_{\mathcal{N}}$?

משפט 9.7 $\mathcal{H}_{\mathcal{N}_{2^n,2,sign}}$ מכילה את כל הפונקציות $f : \{\pm 1\}^n \to \{\pm 1\}$.

ההוכחה למשפט אינה מסובכת, ונוכיח אותה בתרגיל.
כלומר, אם יש לנו רשת מספיק גדולה, אפשר לחשב כל פונקציה.

משפט 9.8 אם $\mathcal{H}_{\mathcal{N}}$ מכילה את כל הפונקציות $h : \{\pm 1\}^n \to \{\pm 1\}$, אזי $|\mathcal{N}|$ היא אקספוננציאלי ב־n.

כלומר, הרשת ענקית. נשאלת השאלה, מה ניתן לחשב ע"י רשתות קטנות?

משפט 9.9 $\mathcal{H}_{\mathcal{N}_{T,T,sign}}$ מכילה את כל הפונקציות המחושבות בזמן T!

מסקנה 9.10 כל עוד אנחנו מתעלמים מזמן אימון הרשת (מציאת המשקולות), ומתעניינים רק בפונקציות המחושבות בזמן T, אפשר להשתמש ברשת נוירונים בגודל T^3, וגם סיבוכיות המדגם היא T^3.

הערה 9.11 לפונקציה המחושבת בזמן T יש כל מיני הגדרות. ישנה הגדרה סטנדרטית - פונקציה שניתן לחשב ע"י פונקציה בוליאנית בגודל T. בהנתן ההגדרה הזו, המשפט הזה נכון. לא ניכנס להגדרה הפורמלית הזו במסגרת השיעור, אך נדון בה מעט בתרגול.

"זה אולי נראה טוב מידי".
אינטואיציה גיאומטרית לשאלה "למה רשתות נוירונים הם כלים מאוד חזקים" - שקופית 14.
ברמה 1 - אפשר לחשב את הפונקציה and (נראה זו בתרגיל - וגם פונקציות נוספות). ברמה 2 - אפשר לקבל צורה גיאומטרית במרחב: חיתוך (and) של חצאי מרחבים, יחידה. ברמה 3 - אפשר להביע איחוד של חיתוכים של חצאי מרחבים, וכך ככל שמוסיפים שכבות אפשר לקבל צורות מסובכות יותר ויותר.

9.2.3 כיצד לאמן רשת נוירונים

עד כה התעלמנו מהאספקט הזה.
באופן כללי, מה היינו רוצים לעשות? לפתור את בעיית ה־ERM: היינו רוצים למצוא היפותזה עם משקולות מתאימים. למרבה הצער:

משפט 9.12 ליישם ERM ל־$\mathcal{N}_{2,2,sign}$ היא בעיה NP קשה.

כלומר, זה גם מאוד קשה במקרים הכי פשוטים. אולי אפשר למצוא אלגוריתם אחר?

משפט 9.13 probably hard להחזיר h עם $L_D^{0-1}(h) \leq \frac{1}{2} - \frac{1}{n^{20}}$ אפילו כאשר $L_{\mathcal{D}}^{0-1}\left(\mathcal{H}_{\mathcal{N}_{\log^2(n),2,sign}}\right) = 0$.

כלומר, המצב קטסטרופלי, ואנחנו לא יכולים להחזיר משהו שהוא יותר טוב מהטלת מטבע.
מה כן עושים? רשתות נוירונים יכולות להביע מחלקות היפוטזות מעולות, אבל צריך לאמן אותן. נעשה זאת באמצעות יוריסטיקות.

יוריסטיקה בסיסית

- התחל עם משקלות רנדומליים w.
- בכל צעד, שנה את w מעט, בכיוון שמקטין את ההפסד.

אבל:

- זה לא קמור! אין שום הבטחות! יכול לקחת הרבה מאוד זמן.
- ובכל זאת, בדר"כ זה עובד... לא מבינים למה[17].

[17] ולכן מתייחסים לפעמים לרשת נוירונים כ"קסם".

יש הרבה דרכים ליישם את היוריסיטקה הבסיסית - הרי אחד:

אלגוריתם 7 אלגוריתם בסיסי ללמידה של רשתות נוירונים

- **אתחול:** כדי לבחור משקלות רנדומליים, בדר"כ בוחרים את המשקולות באופן בלתי תלוי כך ש:

$$\mathbb{E}\left[\sum_{i=1}^{k} w_{i,v}^2\right] = 1$$

למשל, דרך מקובלת היא בצורה יוניפורמית:

$$w_{i,v} \sim U\left[-\sqrt{\frac{3}{k}}, \sqrt{\frac{3}{k}}\right]$$

- **שיפור**: כמו ב־SVM, משתמשים ב־SGD, אבל כאן לא יהיה לנו הבטחה שהוא יתכנס.
  - בשלב הראשון, נקבע הפסד $l : \mathbb{R} \times Y \to \mathbb{R}^+$ (למשל, ה־hinge loss)
  - עבור $(x, y) \in X \times Y$, נסמן:

$$l_{(x,y)}(w) = l\left(h_{\mathcal{N},w}(x), y\right)$$

  - לכל סבסט של דוגמאות $S \supset S'$, נסמן:

$$L_{S'}(w) = \frac{1}{|S'|} \sum_{(x,y)\in S'} l_{(x,y)}(w)$$

  - בכל צעד, נשתמש בחוק העדכון הבא:

$$w_{t+1} = w_t - \eta_t \nabla L_{S'}(w)$$

עבור פרמטר למידה η כלשהוא, שיכול להיות קבוע או משתנה, ו־S' הוא סבסט רנדומי של דוגמאות אימון הנקרא "minibatch". אפשר להחליט כיצד לבחור אותו. למשל: $S' = S$: GD, SGD : $|S'| = 1$. ולסיום, צריך דרך, בהנתן x, y, w מסויימים, לחשב את הגרדיאנט:

$$\nabla l_{(x,y)}(w) = \left(\frac{\partial l_{(x,y)}}{\partial w_e}(w)\right)_{e\in E}$$

כיצד מחשבים את הגרדיאנט? תהי $e \in E$ הקשת שנוירון הפלט שלה הוא בשכבה ה־i. מקבעים את (x, y) ואת כל המשקולות מלבד w_e. ונסמן:

$$l_e(t) = l_{(x,y)}(w^e|t)$$

כאשר $w^e|t$ מחושבת ע"י שינוי w_e ל־t. נשים לב:

$$\frac{\partial l_{(x,y)}}{\partial w_e}(w) = l'_e(w_e),\ l_e(t) = l_y \circ h_d \circ, ..., \circ h_{i+1} \circ h_i(t)$$

כאשר

- $h_i(t)$ הוא הוקטור עם הערכים של הנוירונים בשכבה ה־i.
- עבור $j > i$, h_j מחשבת את השכבה ה־j בהנתן השכבה ה־$(j-1)$.

לכן כדי לחשב, נוכל להשתמש בכלל השרשרת. נשים לב כי הפונקציות האלו הן לא סטנדרטיות, אלא בכמה משתנים, וכן שלושה סוגי האינדקסים קצת מסבכים. אלגוריתם ה־back-propagation עושה זאת בצורה יעילה; נדבר עליו בתרגול.

9.2.4 טרנדים עכשווים ברשתות נוירונים

בניית הרשת

- עומק הרשת: בעבר לא חשבו שאפשר לאמן רשתות בעומק של יותר מ־3, אך היום מאמנים כבר רשתות בעומק 8־9 ונראה שזה עובד לא רע.
- פונקציית האקטיבציה שמתמשים בה הרבה ־ $ReLU$: $\sigma : (a) = max\{0, a\}$.
- בדר"כ משתמשים ברשתות מאוד גדולות ־ יותר פרמטרים מדוגמאות!
- זה עלול לגרום ל־overfitting.
- הרבה פעמים הבחירה של המבנה של הרשת היא בחירה מאוד חשובה. בכל מיני תחומים שונים משתמשים בפרמטרים שונים. בקורסים המדברים על בעיות ספציפיות: למשל בראיה, לומדים מבנים של רשתות שמתאימות לבעיה.

נושאים אלגוריתמיים

- dropout: חלק מהנוירונים "מכבים" באופן רנדומי במהלך האימון (מקפיאים את המשקולות ולא משנים אותם, אך מתחשבים בערך הקפוא).
- אימון ב־GPU

9.2.5 סיכום

רשתות נוירונים הם דרך להגדרי מחלקת היפותזות, והם יכולים להגדיר מחלקת היפותזות אידאלית כל עוד מתעלמים מההיבט החישובי. חישובית, לא ניתן לאמן רשת נוירונים, אבל, באופן אמפירי זה עובד!

רשתות נוירונים הן רעיון די ישן ־ מוצאו בשנות ה־40, והמוטיבציה לא היתה חישוב פונקציות, אלא מדעני מוח שניסו למצוא מודל למוח של אורגניזמים.

האלגוריתמים הבסיסיים גם הומצאו די מזמן ־ פרספטרון, בו משתמשים לאמן נוירון בודד, וה־backpropogation הומצאו בשנות השבעים

בשנות השמונים הומצאו ה־SGD.

בשנות התשעים רשתות נוירונים איבדו את הבכורה ל־SVM ובוסטינג, והחל מ־2006 חל שיפור ענק בביצועים (אולי בגלל המחשבים החזקים), והיום משתמשים בשיטה זו באסכולות רבות (למשל ב־NLP, רשתות נוירונים "מנצחות" את כל שאר האלגוריתמים שפותחו בתחום).

והנקודה המעניינת ביותר ־ **אין לנו מושג מתי רשתות נוירונים פועלות ואיך!**

9.3 עצי החלטה

מהם עצי החלטה? כמו רשת נוירונים, זהו מעין כלי חישובי שמאפשר לנו לחשב פונקציות.

הגדרה 9.14 עץ החלטה ב־n משתנים על $\{\pm 1\}^n$ מוגדר ע"י:

- עץ בינארי בעל שורש בו לכל קודקוד יש 2 או 0 ילדים.
- קודקודים פנימיים המסומנים ע"י x_i, $1 \leq i \leq n$.
- עלים מסומנים ע"י 0 או 1.
- קשתות מסומנות ע"י -1 או 1 (המגדיר לנו בעצם כיוון המשך: ימינה או שמאלה).

הגדרה 9.15 עץ החלטה T מגדיר פונקציה $h_T : \{\pm 1\}^n \to \{0, 1\}$ באופן טבעי: כדי לחשב את $h_T(x)$, התחל מהשורש וטייל בעץ לפי x עד שתגיע לעלה. $h_T(x)$ הוא התגית של העלה הזה.

אנחנו נלמד אלגוריתמים שלומדים עצי החלטות. נוכל לדבר על סיבוכיות המדגם, כושר ההבעה של עצים, ועוד. לגבי כושר ההבעה - אין הרבה מה להגיד[18].

אם מסתכלים על עצים כלליים, שיכולים להיות ענקיים, אפשר לקבל כל פונקציה. תוצאה זו תנבע מהמשפט שנוכיח מיד.

מה לגבי סיבוכיות המדגם?

משפט 9.16 תהי $\mathcal{T}_k$ מחלקת ההיפוטזות של עצי ההחלטה עם $k \leq 2^n$ עלים. אזי:

$$k \leq VC\left(\mathcal{T}_k\right) \leq 2k\left\lceil \log_2\left(4nk\right)\right\rceil$$

הערה 9.17 למה בחרנו את מספר העלים כפרמטר הסיבוכיות שלנו? יכולנו לבחור ערכים אחרים בעץ. אבל למעשה, כל הפרמטרים, בשל מבנה העץ, קשורים זה לזה, אז תחת הגדרה אחרת היינו מקבלים תוצאות מאוד דומות.

הוכחה: נוכיח המשפט ע"י כך שנוכיח שאפשר לתאר כל עץ החלטה ע"י $2k\left\lceil \log_2\left(4nk\right)\right\rceil$ ביטים.
באינדוקציה, הוכחנו במתמטיקה דיסקרטית שלעץ בעל k עלים יש $2k-1$ קודקודים. כל קודקוד ניתן לתאר ע"י $\left\lceil \log_2\left(n+1\right)+\log_2\left(2k\right)\right\rceil$ ביטים באופן הבא:[19]

- תיאור של הקודקוד (מהצורה $x_i = 1$? או עלה עם ערך $0/1$)
- זהות ההורה של הקודקוד.

החלק השני של אי השיוויון נובע מכך שלכל $A \supset \{\pm 1\}^n$ מגודל k מנותצת (נותר כתרגיל).

■

9.3.1 סיבוכיות חישובית

לגבי סיבוכיות החישוב - זוהי בעיה NPקשה[20]. האלגוריתמים שלומדים עצי החלטה משתמשים ביוריסטיקות חמדניות. יוריסטיקה בסיסית - שקופית 29.

אלגוריתם 8 אלגוריתם חמדן בסיסי ללמידת עצי החלטה

קלט: סט אימון $S \subset \{\pm 1\}^n \times \{\pm 1\}$.
פעולה:

- התחל עם עלה בודד.
- בכל צעד, החלף אחד מהעלים בעץ פשוט בעל הורה ושני עלים, בצורה שמקטינה את ההפסד מבין כל האפשרויות.
- עצור כאשר לא ניתן לשפר את התוצאה.

הערה 9.18 בדר"כ בוחרים עלה אקראי. אפשר גם את הבחירה הזו לעשות בצורה חמדנית.

האלגוריתמים שונים בהפסדים שהם מתייחסים אליהם - האלגוריתם מעלה משתמש בהפסד אמפירי. אלגוריתמים אחרים עושים אופטימיזציה באופן שונה.

נשים לב אלגוריתמים חמדנים יכולים להפיק בעצמם עצים מאוד גדולים. אפשר להתמודד עם הבעיה הזו בכמה דרכים:

- עצירה מוקדמת של האלגוריתם.
- גיזום $\left(\text{pruning}\right)$ של העץ הסופי המתקבל ע"י האלגוריתם.

[18] אז אולי בעצם לא יכולנו לדבר על זה ;)

[19] אפשר למצוא תיאורים יותר טובים, כלומר חסכניים יותר בביטים, אבל תיאור זה מספיק טוב לנו להוכחה. למשל את החלק השני המתייחס להורה אפשר לצמצם כי הרי קודקוד לא יכול להיות הורה.

[20] כמה מפתיע!

אפשר להרחיב בצורה טבעית את האלגורתמים הללו לקלטים שאינם בינאריים, ולמקרים שבהם הפלט הוא מספר מממשי או מקרים נוספים.

השבוע הבא יהיה השבוע האחרון בו נדבר על קלסיפיקציה. סיימנו פחות או יותר את כל האלגוריתמים שנדבר עליהם במסגרת הקורס. בשיעור הבא נסכם את מה שעשינו - איך להפעיל את התהליך: בחירת האלגוריתם, איך להריץ אותו וכו'. בחלק השני של השיעור נדבר על Boosting: שיטות המאפשרות לקחת אלגוריתם למידה אחד, שהוא אולי מאוד פשוט, ולהפוך אותו לאלגוריתם חזק הרבה יותר.

לאחר השיעור הזה נעזוב את הקלסיפיקציה הבינארית, ונעבור לדון בדרכים בהן אנחנו מייצגים את הקלטים שלנו, נושא ממנו התעלמנו עד כה.

10 Boosting

"לומד חלש"[21] הוא אלגוריתם למידה נאיבי הלומד היפוטזות מאוד פשוטות (מעין כללי אצבע).

ההרצאה היום תעסוק באלגוריתמים אשר לוקחים לומד חלש, ובונים ממנו "לומד חזק". כלומר, אלגוריתם למידה המסוגל ללמוד היפותזות מורכבות. אלגוריתמים (לומדים חזקים) המתאימים לתיאור הנ"ל נקראים **אלגוריתמי האצה** (**Boosting**).

ההרצאה כולה תעסוק בקלסיפיקציה בינארית. כלומר ננית ש-$Y = \{\pm 1\}$, ונרצה למזער את $L_{\mathcal{D}}^{0-1}$.

10.1 לומדים חלשים

לפני שנציג את האלגוריתמים, נביט במספר דוגמאות לאלגוריתמים המתאימים לתיאור של לומד חלש.

סיפים על הישר

כאן $X = \mathbb{R}$.

הגדרה 10.1 סף על הישר הוא היפותזה מהצורה:

$$h_\theta(x) \mapsto sign(x - \theta)$$

עבור $\theta \in \mathbb{R}$.

דוגמא ללומד חלש הוא אלגוריתם המממש את כלל ה-ERM ביחס למחלקה $\mathcal{H} = \{h_\theta | \theta \in \mathbb{R}\}$. נעיר שבתרגול 3 ראינו שניתן לעשות זאת ביעילות.

גדמי החלטה (Decision Stumps)

גדמי החלטה מכלילים סיפים על הישר. כאן $X = \mathbb{R}^n$.

הגדרה 10.2 גדם החלטה הוא היפותזה מהצורה:

$$h_{\theta,i}(x) \to sign(x_i - \theta)$$

עבור $\theta \in \mathbb{R}$ ו-$i \in [n]$.

דוגמא ללומד חלש הוא אלגוריתם המממש את כלל ה-ERM ביחס למחלקה $\mathcal{H} = \{h_{\theta,i} | \theta \in \mathbb{R},\ i \in [n]\}$. בדומה לסיפים על הישר, גם עבור גדמי החלטה ניתן לממש את כלל ה-ERM ביעילות.

גרסאות חלשות של לומדים חזקים

באופן כללי, ניתן לקחת כל אלגוריתם למידה שלמדנו, ולהפעיל אותו על תת קבוצה קטנה של הקואורדינטות, או לחלופין, לדרוש ממנו להחזיר היפותזה מאוד פשוטה (עץ החלטה עם מספר קטן של עלים, וקטור מנורמה קטנה, ועוד...).

[21] מכיוון ולא נכחתי בשיעור זה, החומר נלקח ישירות מהסיכום והמצגת שהעלה מרצה הקורס, עם עריכות קטנות שלי.

10.2 יערות אקראיים

הדרך אולי הנאיבית ביותר לבנות לומד חזק מלומד חלש היא פשוט **להריץ את הלומד החלש בכמה אופנים שונים**, לקבל היפותזות:

$$h_1, ..., h_k : X \to \{\pm 1\}$$

ולהחזיר את הכרעת הרב, כלומר את ההיפותזה:

$$h(x) = \text{Majority}(h_1(x), ..., h_k(x))$$

אלגוריתם פופולארי המממש את הרעיון הנ"ל הוא האלגוריתם הבא:

אלגוריתם 9 אלגוריתם היערות האקראיים

פרמטרים: אלגוריתם ("חלש") W הלומד עצים, מספר עצים k, מספר קואורדינטות d.
קלט: $S \in (\{\pm 1\}^n \times \{\pm 1\})^m$.

- עבור $i \in [k]$:
  - בחר באקראי תת קבוצה $C \subset [n]$ בגודל d.
  - הרץ את W ביחס לקואורדינטות ב־C, וקבל היפוטזה h_i.
- החזר את ההיפוטזה $h(x) = \text{Majority}(h_1(x), ..., h_k(x))$.

10.3 האצה אדפטיבית (AdaBoost)

אלגוריתם ה־AdaBoost משתמש בלומד החלש על תת מדגמים של המדגם הנתון:

$$S = \{(x_1, y_1), ..., (x_m, y_m)\} \subset X \times \{\pm 1\}$$

כלומר, הלומד החלש מקבל כל פעם תת־מדגם $S \supset S'$, ועליו למצוא היפותזה עם ביצועים לא טריוויאלים (שגיאה קטנה מחצי) על S'.

AdaBoost משקלל את ההיפותזות שהחזיר הלומד החלש, כך ש**במידה והלומד החלש הצליח במשימתו, להיפותזה המשוקללת תהיה שגיאה קטנה יותר**.

יהיה נוח יותר לעבוד עם **משקולים** של המדגמים במקום תתי מדגמים; AdaBoost יחזיק משקל $D_i \geq 0$ לכל דוגמא. המשקלות הללו יגדירו התפלגות:

$$\left(\overline{D}_1, ..., \overline{D}_m\right) := \frac{1}{\sum_{i=1}^n D_i}(D_1, ..., D_m)$$

על הדוגמאות.

בכל שלב, הלומד החלש ידרש למצוא היפותזה $h : X \to \{\pm 1\}$ הצודקת על רוב הדוגמאות ביחס להתפלגות הנ"ל. כלומר, היפוטזה עם שגיאת $0 - 1$ ממושקלת:

$$L_{S,D}^{0-1}(h) := \sum_{i=1}^m \overline{D}_i l_{0=1}(h(x_i), y_i)$$

לא טריוויאלית.

הערה 10.3 את רב אלגוריתמי הלמידה פשוט מאוד להרחיב לאלגוריתמים העובדים גם עם מדגמים משוקללים, שכן רב האלגוריתמים ממזערים, בין אם בצורה יוריסטית ומקומית, ובין אם בצורה ריגורוזית, את $L_S(h)$ עבור איזשהוא הפסד. לכן, ניתן להכליל אותם למדגמים משוקללים, וזאת ע"י מעבר מ־$L_S(h)$ ל־$L_{S,D}(h)$.

בכל שלב, AdaBoost יפעיל את הלומד החלש ביחס למשקלות הנוכחיים. במהלך הריצה, הוא **ירכז את המשקלות ב"דוגמאות הקשות"** - הדוגמאות שעליהן "רוב" ההיפותזות שהוחזרו ע"י הלומד החלש נכשלות.

קונקרטית, נסמן ב:

$$D^{(t)} = \left(D_1^{(t)}, ..., D_m^{(t)}\right)$$

את המשקולות בתחילת השלב ה־t. בשלב הראשון המשקולות יהיו אחידים, כלומר לכל i יתקיים $D_i^{(1)} = 1$. בשלב ה־t, האלגוריתם יריץ את הלומד החלש ביחס למשקלות $D^{(t)}$ ויקבל היפותזה $h_t : X \to \{\pm 1\}$. נסמן ב־

$$\varepsilon_t := L_{S,D^{(t)}}^{0-1}(h_t)$$

את השגיאה המשוקללת של h_t. AdaBoostיגדיר ל־h_t מקדם:

$$w_t = \frac{1}{2} \cdot \log\left(\frac{1}{\varepsilon_t} - 1\right)$$

נשים לב כי w_t נע בין 0 (עבור $\varepsilon_t = \frac{1}{2}$) ל־$\infty$, ויגדל ככל ש־$\varepsilon_t$ יקטן. המקדם הנ"ל יקבע את החשיבות לה ייחס האלגוריתם ל־h_t. החשיבות הנ"ל תבוא לידי ביטוי בדומיננטיות של h_t בהיפותזה הסופית, וגם בגודל השינוי במעבר בין $D^{(t)}$ ל־$D^{(t+1)}$. קונקרטית, בסוף השלב ה־t, AdaBoost יעדכן את המשקלות כך ש**משקל הדוגמאות עליהן** h_t **שגה יגדל פי** e^{w_t}**, בעוד המשקל על הדוגמאות עליהן** h_t **צדק יקטן פי** e^{w_t}.

בסוף הריצה, לאחר T שלבים, תוחזר ההיפותזה:

$$H_T = \sum_{t=1}^{T} w_t h_t$$

לסיכום, מה המשמעות של כל העדכונים האלו?

החשיבות של ההיפוטזה ה־h_t נקבעת ביחס הפוך לשגיאה ־ ככל שהשגיאה קטנה, החשיבות גדלה. לאחר מכן, המטרה באיטרציה הבאה היא למצוא היפותזה שתצליח במקומות בהם טעינו קודם לכן. על כן, המשקולות מעודכנים כך שאם טעינו בדוגמא מסויימת, המשקל שלה יותר גדול.

ולסיום, האלגוריתם בצורה פורמלית:

אלגוריתם 10 AdaBoost

פרמטרים: אלגוריתם ("חלש") W.

קלט: $S \in (X \times \{\pm 1\})^m$, מספר איטרציות T.

- אתחל $D^{(1)} = (1, ..., 1)$.
- עבור $t \in [T]$:
  - הרץ את W יחס למשקלות $D^{(t)}$ וקבל היפותזה $h_t : X \to \{\pm 1\}$.
  - הגדר $\varepsilon_t := L_{S,D^{(t)}}^{0-1}(h)$.
  - הגדר $w_t = \frac{1}{2} \cdot \log\left(\frac{1}{\varepsilon_t} - 1\right)$
  - עדכן $D_i^{(t+1)} = D_i^{(t)} \cdot e^{-w_t h_t(x_i) y_i}$
- החזר את ההיפוטזה $H_T = \sum_{t=1}^{T} w_t h_t$.

10.3.1 השגיאה האמפירית

נניח שהאלגוריתם החלש הצליח במשימתו, והשגיאות ε_t היו כולן קטנות מ־$\frac{1}{2}$; מה תהיה השגיאה האמפירית של ההיפותזה הסופית h? המשפט הבא חוסם את $L_S^{0-1}(h)$ ביחס לשגיאה ε_t:

משפט 10.4 נניח שלכל t, $\varepsilon_t \leq \frac{1}{2} - \gamma_t$. אזי:

$$L_S^{0-1}(h) \leq \exp\left(-2 \cdot \sum_{t=1}^{T} \gamma_t^2\right)$$

למשל, אם בכל שלב השגיאה היתה קטנה מ־0.4, נקבל:

$$L_S^{0-1}(h) \leq \exp(-0.02T)$$

כלומר, השגיאה קטנה במהירות אספוננציאלית. **הוכחה:** במסגרת ההוכחה, נראה שבכל שלב t, AdaBoost מקטין פי $e^{-2\gamma_t^2}$ את השגיאה של תחליף (קמור) מסויים ל־l_{0-1}.

הגדרה 10.5 נגדיר את **ההפסד המעריכי** (Exponential Loss)להיות הפונקציה $l_{exp} : \mathbb{R} \times \{\pm 1\} \to \mathbb{R}^+$ הבאה:

$$l_{exp}(\hat{y}, y) = exp(-\hat{y}y)$$

נשים לב שבדומה ל־l_{0-1}, l_{exp} הוא מהצורה $f(\hat{y}y)$ עבור $f : \mathbb{R} \to \mathbb{R}^+$ - l_{0-1} הוא פונקציית מדרגה שתמיד מתחת ל־l_{exp}

הוכחת הלמה הבאה פשוטה ומושארת כתרגיל:

למה 10.6 לכל $h : X \to \mathbb{R}$:

$$L_S^{0-1}(h) \leq L_S^{exp}(h)$$

לב הוכחת המשפט הוא הלמה הבאה:

למה 10.7 נסמן $H_t = \sum_{t'=1}^{t} w_{t'} h_{t'}$ (עבור $H_t := 0,\ t = 0$). אזי, לכל $t \geq 1$ מתקיים:

$$L_S^{exp}(H_t) \leq \exp\left(-\gamma_t^2\right) L_S^{exp}(H_{t-1})$$

הוכחה: ראשית, נשים לב שמתקיים (ישירות מהגדרת AdaBoost):

$$\begin{aligned}
D_i^{(t)} &:= e^{-w_{t-1}h_{t-1}(x_i)y_i} \cdot D_i^{(t-1)} \\
&= e^{-w_{t-1}h_{t-1}(x_i)y_i} \cdot e^{-w_{t-2}h_{t-2}(x_i)y_i} \cdot D_i^{(t-2)} \\
&= e^{-[w_{t-1}h_{t-1}(x_i)y_i + w_{t-2}h_{t-2}(x_i)y_i]} \cdot D_i^{(t-2)} \\
&\vdots \\
&= e^{-[w_{t-1}h_{t-1}(x_i)y_i + w_{t-2}h_{t-2}(x_i)y_i + \ldots + w_1 h_1(x_i)y_i]} \cdot \overbrace{D_i^{(1)}}^{=1} = e^{-H_{t-1}(x_i)y_i}
\end{aligned}$$

כעת, נחשב:

$$\begin{aligned}
\frac{L_S^{exp}(H_t)}{L_S^{exp}(H_{t-1})} &= \frac{\sum_{i=1}^m \exp(-H_t(x_i)\, y_i)}{\sum_{i=1}^m \exp(-H_{t-1}(x_i)\, y_i)} \\
&= \sum_{i=1}^m \frac{\exp(-H_{t-1}(x_i)\, y_i)}{\sum_{j=1}^m \exp(-H_{t-1}(x_j)\, y_j)} \cdot \exp(-w_t h_t(x_i)\, y_i) \\
&\overbrace{=}^{\text{above}} \frac{1}{\sum_{i=1}^m D_i^{(t)}} \cdot \sum_{i=1}^m D_i^{(t)} \exp(-w_t h_t(x_i)\, y_i)
\end{aligned}$$

ונשים לב שהביטוי האחרון הוא $L_{S,D^{(t)}}^{exp}(w_t h_t)$. מהי, אם כן, השגיאה של $w_t h_t$ ביחס ל־l^{exp}?
ובכן, מכיוון ו־$L_{S,D^{(t)}}^{0-1}(h_t) = \varepsilon_t$, הדוגמאות עבורן $h(x_i)\, y_i = -1$ מהוות ε_t אחוז (לפי $D^{(t)}$) מכלל הדוגמאות. הדוגמאות הללו תורמות:

$$\varepsilon_t e^{-h(x_i)y_i w_t} = \varepsilon_t e^{w_t}$$

ל־$L_{S,D^{(t)}}^{exp}(w_t, h_t)$. יתר הדוגמאות מהוות $1 - \varepsilon_t$ מכלל הדוגמאות, ועליהן $h(x_i)\, y_i = 1$, ולכן הן תורמות $(1 - \varepsilon_t)\, e^{-w_t}$לסכום. אם נחבר, נקבל:

$$\begin{aligned}
L_{S,D^{(t)}}^{exp}(w_t h_t) &= (1 - \varepsilon_t)\, e^{-w_t} + \varepsilon_t w^{w_t} \\
&= \sqrt{\frac{\varepsilon_t}{1 - \varepsilon_t}} \cdot (1 - \varepsilon_t) + \sqrt{\frac{1 - \varepsilon_t}{\varepsilon_t}} \cdot \varepsilon_t \\
&= 2\sqrt{\varepsilon_t (1 - \varepsilon_t)} = 2\sqrt{\left(\frac{1}{2} - \gamma_t\right)\left(\frac{1}{2} + \gamma_t\right)} \\
&= \sqrt{(1 - 2\gamma_t)(1 + 2\gamma_t)} = \sqrt{1 - 4\gamma_t^2} \leq \exp\left(-2\gamma_t^{2}\text{‘}\right)
\end{aligned}$$

כאשר אי השיויון האחרון נובע מכך שלכל $x \in \mathbb{R}$, $1 + x \leq e^x$. אם נציב באי השיוויון שהתקבל מעלה, נקבל את הלמה המבוקשת. ■

לסיכום, נשתמש בשתי הלמות הקודמות על מנת לסיים את הוכחת המשפט - ואכן, משתיהן נובע כי:

$$L_S^{0-1}(H_T) \overbrace{\leq}^{(1)} L_S^{exp}(H_T) \overbrace{\leq}^{(2)} \exp\left(-2\sum_{i=1}^{T}\gamma_t^w\right) \cdot L_S^{exp}(H_0) = \exp\left(\sum_{t=1}^{T}\gamma_t^2\right)$$

■

10.3.2 שגיאת ההכללה

בסעיף הקודם הראינו שכאשר מריצים את AdaBoost והלומד החלש מצליח להחזיר היפותזות לא טריוויאליות, השגיאה האמפירית היא נמוכה. כמובן והעובדה זו לא מבטיחה לנו דבר לגבי השגיאה האמיתית - יכול להיות שהאלגוריתם עשה אוברפיט!

המשפט הבא מראה שכאשר הלומד החלש מחזיר היפותזות ממחלקה ממימד d VC, אלגוריתם AdaBoost יחזיר היפותזה ממחלקה במימד VC פחות או יותר dT. לכן, אם $m \gg dT$, AdaBoost לא יעשה אוברפיט.

משפט 10.8 תהא $B \subset \{\pm 1\}^X$ מחלקת היפותזות. עבור $T \geq 1$ נגדיר:

$$\mathcal{H}(B,T) = \left\{ x \mapsto sign\left(\sum_{t=1}^{T} w_t h_t(x)\right) \middle| \, w \in \mathbb{R}^T,\ h_1,...,h_T \in B \right\}$$

נשים לב שאם הלומד החלש מחזיר היפותזות ב-B, אז AdaBoost מחזיר היפותזות ב-$\mathcal{H}(B,T)$. אזי, עבור B עם $VC(B) = d$:

$$VC(\mathcal{H}(B,T)) = O(dT\log(dT))$$

נוכיח טענה מעט יותר כללית.

בהנתן זוג מחלקות $B \subset \{\pm 1\}^X$ ו- $\mathcal{F} \subseteq \{\pm 1\}^{\{\pm 1\}^T}$, נגדיר:

$$F \circ B := \{ x \mapsto f(h_1(x),...,h_T(x)) | \, h_1,...,h_T \in B,\ f \in \mathcal{F} \}$$

נשים לב ש-$\mathcal{H}(B,T) = \mathcal{F} \circ B$, כאשר $\mathcal{F}$ היא מחלקת ההיפותזות של חצאי מרחב הומוגניים על $\mathbb{R}^T$. אם כך, המשפט הרצוי נובע מהמשפט הבא:

משפט 10.9 תחת הגדרות אלו:

$$VC(\mathcal{F} \circ B) \leq (4VC(\mathcal{F}) + 4VC(B)T)\log(2VC(\mathcal{F}) + 2VC(B)T)$$

הוכחה: נסמן:

$$d_B = VC(B),\ d_{\mathcal{F}} = VC(\mathcal{H})$$

תהא $A \subset X$ קבוצה המנותצת ע"י $\mathcal{F} \circ B$ בגודל m. עלינו להראות ש:

$$m \leq (4d_F + 4d_B T) \cdot \log(2d_F + 2d_B T)$$

נביט במחלקה $(\mathcal{F} \circ B)|_A$. על היפותזה במחלקה זו מוגדרת ע"י T פונקציות $h_1,...,h_T \in B|_A$, וע"י פונקציה $f \in \mathcal{F}|_{h(A)}$, כאשר:

$$h(A) := \{ h_1(a),...,h_T(a) | \, a \in A \}$$

ממשפט סאור־שלח יש לנו $m^{2d_B} \geq$ אפשרויות לבחור כל h_i, ולכן יש לנו $m^{2d_B T} = \left(m^{2d_B}\right)^T \geq$ אפשרויות לבחור את $h_1, ..., h_T$, ובהינתן בחירה שכזו, יש לנו $m^{2d_F} \geq |h(A)|^{2d_F}$ אפשרויות לבחור את f. מכאן:

$$|(\mathcal{F} \circ B)|_A| \leq m^{2d_F + 2d_B T}$$

מצד שני, A מנותצת, ולכן:

$$2^m \leq |(\mathcal{F} \circ B)|_A| \leq m^{2d_F + 2d_B T}$$

ולכן משני אי השיוויונות נקבל:

$$2^m \leq m^{2d_F + 2d_B T}$$

מכאן:

$$\frac{m}{\log(m)} \leq 2d_F + 2d_B \Rightarrow m \leq (4d_F + 4d_B T) \cdot \log(2d_F + 2d_B T)$$

כאשר הגרירה נובעת מכך ש:

$$\forall a, x > 0, \ \frac{x}{\log(x)} \leq a \Rightarrow x \leq 2 \cdot a \log(a)$$

טענה זו מופיעה כלמה $A.1$ בספר של שי ושי. ■

10.3.3 מלמידות חלשה לחזרה ־ היסטוריה והשלכות תיאורטיות של AdaBoost

ההיסטוריה של AdaBoost החלה בשנת 1988, אז שאלו Mike Kearns ו־Leslie Valiant את השאלה התיאורתית הבאה:

> **שאלה:** נקבע מחלקת היפותזות $\mathcal{H}$. נניח שקיים אלגוריתם יעיל שלומד את $\mathcal{H}$ במקרה הפריד, אבל **באופן חלש** ־ כלומר, עבור $\delta_0 > 0, \ \frac{1}{2} > \varepsilon_0$ **קבועים** מובטח שהאלגוריתם יחזיר, בהסתברות $\leq 1 - \delta_0$ היפותזה עם שגיאה $\geq \varepsilon_0$. האם ניתן להשתמש בו על מנת לקבל אלגוריתם למידה יעיל העובד לכל ε, δ? כלומר, האם קיימת **סכמת האצה** כללית, המאפשרת להפוך אלגוריתמים חלשים יעילים לאלגוריתמים יעילים רגילים (חזקים)?

תשובה חיובית לשאלה ניתנה בשנת 1990 ע"י Rob Schapire, אז דוקטורנט ב־MIT. מספר שנים לאחר מכן, ב־1995, הוא הציע, יחד עם יואב פורינד, את AdaBoost, שנתן אף הוא תשובה חיובית לשאלה הנ"ל. היתרון של AdaBoost על פני האלגוריתם הקודם הוא ש־AdaBoost מהיר הרבה יותר.

האלגוריתם של פורינד ושפירי זכה ועודנו זוכה להצלחה מעשית רבה בשורה של בעיות, ואף זיכה את ממציאו ב"פרס גדל" ־ אחד הפרסים הבולטים במדעי המחשב.

מלבד ההצלחה הפרקטית, לאלגוריתם היו לא מעט השלכות על התיאוריה של למידה, חלקן מפתיעות. וליאנט וקרנס שאלו את השאלה על מנת לקבל כלי המאפשר לפתח אלגוריתמים לבעיות PAC. על פניו, תשובה חיובית לשאלה שלהם תקל מאוד את המלאכה של עיצוב אלגוריתמי למידה ־ במקום לבנות אלגוריתם שצריך להחזיר שגיאה מאוד קטנה, די לפתח אלגוריתם המחזיר היפותזה עם שגיאה קטנה במעט מ־$\frac{1}{2}$.

בשנת 1988, רק ארבע שנים לאחר שהחלו לחקור למידה חישובית כחלק ממדעי המחשב, קיום של סכמת ההאצה הנ"ל היה נראה בהחלט כמו כלי שיאפשר לפתח אלגוריתמים להרבה בעיות למידה. ברבות השנים התפתחה ההבנה שלמרבה הצער, מלבד חצאי מרחבים, כמעט כל בעיות הלמידה (כפי שהן מוגדרות כיום) לא ניתנות לפתרון ביעילות. לכן, אחת המטרות המקוריות שלשמן AdaBoost פותח נכשלה.

יתר על כן, באופן אירוני משהו, אחד השימושים של AdaBoost בתיאוריה של למידה הוא להראות שבעיות למידה הן **מאוד** קשות! קיום של סכמת האצה מראה שאם קיים לבעיה אלגוריתם חלש, אז קיים לאותה בעיה גם אלגוריתם חזק. באופן שקול, אם לא קיים אלגוריתם חזק, אז לא קיים אפילו אלגוריתם חלש! לכן העובדה שלא קיים אלגוריתם יעיל גוררת שלא קיים אפילו אלגוריתם חלש לבעיה!

מכאן, כפי שאמרנו מספר פעמים במהלך הקורס, ככל הנראה רב בעיות ה־PAC הן קשות מאוד ־ במובן שאפילו במקרה פריד לא קיים אלגוריתם יעיל המסוגל להחזיר היפותזה עם שגיאה קטנה, אפילו במעט, מ־$\frac{1}{2}$.

נעיר שמלבד ההשלכה הנ"ל, ל־AdaBoost קשרים נוספים להרבה מושגים בסיסיים בלמידה, ביניהם רגולציה, שוליים רחבים, ועוד.

10.3.4 אפליקציה - זיהוי פנים $(Viola\ and\ Jones)$

(באופן ויזואלי - בשקופיות להרצאה 8 באתר הקורס). המשימה - תיוג ריבועים בתמונה כפרצוף או לא פרצוף. כללי אצבע:

- "אזור העיניים בדר"כ כהה יותר מהלחיים"
- "גשר האף בהיר יותר מהעיניים"

מטרה - שימוש ב-AdaBoost לשילוב מספר כללי אצבע לקבלת מזהה פנים.
הלומד החלש הוא ERM על כל ההיפותזות מהצורה:

$$h(x) = sign(g_{R,t}(x) - \theta)$$

כאשר R הוא ריבוע המיושר לצירים, $\theta \in \mathbb{R}$, ו-$t \in \{A, B, C, D\}$ הוא סוג המתאים למאסק מסויים (כאשר כל אחד מייצג כלל האצבע כלשהוא), המגדיר בתמונה שני איזורים - איזור אדום ואיזור כחול (כאשר לא כל התמונה חייבת להופיע בחלוקה זו). $g_{R,t}(x)$ הוא סכום כל ערכי הפיקסל באיזור האדום, פחות כל ערכי הפיקסל המופיעים באיזור הכחול.
ואכן, הקלספיירים הנבחרים ע"י AdaBoost לפי היישום של Viola and Jones, הם המאסקים המתאימים לשני כללי האצבע שרשמנו.

10.4 דלילות ובחירת פיצ'רים על קצה המזלג

נניח והמחלקה של הלומד החלש הינה:

$$\mathcal{H} = \{h_1, ..., h_N\}$$

תחילה ממפים את x ל-$\mathbb{R}^N$ בעזרת:

$$\Psi(x) = (h_1(x), ..., h_N(x))$$

לאחר מכן, AdaBoost לומד קלסיפייר:

$$h_w : \mathbb{R}^N \to \mathbb{R},\ w \in \mathbb{R}^N$$

נשים לב כי w דליל! רק T פיצ'רים שאינם 0 נבחרים. מקבלים שיפור של סיבוכיות המדגם (אנלוגיה לרגולציה ב-SVM), וכן זמן אימון ובדיקה מהירים (אנלוגיה לקרנלים ב-SVM). כמו כן, ניתן להשתמש בפיצ'רים שנבחרים כפיצ'רים טובים\אינפורמטיביים עבבור בעיות למידה נוספות.

10.4.1 שיטות נוספות לבחירת פיצ'רים \ דלילות

- **פילטרים**: באופן בלתי תלוי להעריך כל פיצ'ר (למשל, למדוד את ההתאמה שלו עם ההיפותזה המתקבלת בסוף התהליך), ואז לבחור את T הטוב ביותר
- **חמדנות "קדימה"**: להתחיל עם פיצ'ר יחיד, ובאופן חמדן להוסיף עוד
- **חמדנות "אחורה"**: להתחיל עם כל הפיצ'רים, ואז באופן חמדן להוריד
- **רגולציית** l^1: SVM עם פקטור הרגולציה $\lambda \sum_{i=1}^{N} |w_i|$
- שימוש בשכבה האחרונה של רשת נוירונים
- עוד בפרק 25 בספר של שי ושי

11 סיכום ביניים מס' 2 - תורת ההכללה ואלגוריתמי למידה

[22]ניזכר כי המטרה הבסיסית בלמידה חישובית היא ללמוד מיפוי:

$$h^* : X \to Y$$

על סמך מדגם:

$$S = \{(x_1, y_1), ..., (x_m, y_m)\} \subset X \times Y$$

[22]מכיוון ולא נכחתי בשיעור זה, החומר נלקח ישירות מהסיכום והמצגת שהעלה מרצה הקורס, עם עריכות קטנות שלי.

חלק א' - מודל PAC ותורת ההכללה

בשלוש ההרצאות הראשונות הגדרנו את מודל PAC ללמידה, ופיתחנו את תורת ההכללה, המאפשרת לנו להבין כמה דוגמאות עלינו לראות על מנת ללמוד.

במודל התיאורטי שהצגנו, הנחנו שקיימת התפלגות $\mathcal{D}$ ממנה נדגמות הדוגמאות. בהתאם לכך, הגדרנו את ההפסד של היפותזה $h : X \to Y$ ע"י:

$$L_{\mathcal{D}}(h) = \mathbb{E}_{(x,y)\sim\mathcal{D}} l(h(x), y)$$

האבחנה הראשונה שעשינו היא ש"אין ארוחות חינם". כלומר, לא קיים אלגוריתם המסוגל ללמוד כל מיפוי כאשר כמות הדוגמאות מוגבלת. במילים אחרות, אנחנו צריכים איזשהו **ידע מוקדם** על הפונקציה h^* (או ההתפלגות $\mathcal{D}$) על מנת להיות מסוגלים ללמוד. הדרך בה ניתן לבטא ידע מוקדם במודל PAC הוא הצבעה על **מחלקת היפותזות** (כלומר, אוסף $\mathcal{H} \subset Y^X$) בה אנו משערים שקיימת היפותזה טובה.

תורת ההכללה שפיתחנו הראתה שכאשר $\mathcal{H}$ "קטנה" בהשוואה לכמות הדוגמאות שבידנו, השגיאה האמפירית $(L_S(h))$ של **כל ההיפותזות ב-$\mathcal{H}$** קרובה לשגיאה האמיתית $(L_{\mathcal{D}}(h))$. לכן, כל אלגוריתם המחזיר היפותזה ב-$\mathcal{H}$ עם שגיאה אמפירית טובה, יחזיר בעצם היפותזה עם שגיאה אמיתית טובה. לכן, בהנתן אלגוריתם שכזה, אם הידע המוקדם שלנו נכון ובאמת קיימת ב-$\mathcal{H}$ היפותזה טובה, נוכל ללמוד.

חלק ב' - אלגוריתמי למידה

בחמשת השבועות שלאחר מכן למדנו שורה של אלגוריתמי למידה:

- אלגוריתמי RLM לבעיות קמורות, בפרט רגרסיה לינארית (כלומר, רגרסייה בה ההיפותזות הן פונקציונאלים לינאריים)
- אלגוריתם ה-SVM לקלסיפיקציה בעזרת מפרידים לינארים
- שיטות גרעין, המאפשרות לנו ללמוד במימד גבוה, וכפועל יוצא להרחיב את המחלקת ההיפותזות שלנו
- רשתות נוירונים
- עצי החלטה
- השכן הקרוב
- יערות אקראיים
- AdaBoost

משני החלקים עולה השיטה הבאה ללמוד:

1. בחרו אלגוריתם למידה.
2. קבעו את הפרמטרים של האלגוריתם (הגרעין ופרטמר הרגולריזציה ב-SVM, הגודל והמבנה של הרשת עבור אלגוריתמים הלומדים רשתות נוירונים, הלומד החלש ומספר האיטרציות ב-AdaBoost וכו'), כך שיעשה אופטימיזציה על מחלקת היפותזות $\mathcal{H}$ בה אתם מעריכים ש-
   (א) יש היפותזה טובה.
   (ב) היא מספיק קטנה ביחס לכמות הנתונים שברשותכם.
   (ג) אתם מעריכים שהאלגוריתם יצליח למצוא בה היפותזה טובה.
3. הריצו את האלגוריתם ביחס לנתונים שאספתם.

שגיאת קירוב, שגיאת הכללה ושגיאת האופטימיזציה - יחסי גומלין

בשימוש בלמידה חישובית, לפי השיטה שתוארה, יש לעשות הרבה מאוד בחירות:

- ראשית, יש לבחור את אלגוריתם הלמידה.
- לאחר מכן, צריך לבחור את הפרמטרים של האלגוריתם: הבחירות הללו ישפיעו על מחלקת ההיפותזות בה האלגוריתם יחפש היפותזה טובה.

- לבסוף, צריך לבחור מימוש קונקרטי של האלגוריתם: GD, SGD או שיטה אחרת עבור SVM ורשתות נוירונים, השיטה המדוייקת בה עושים את הבחירה החמדנית בלמידת עצים, וכו'. עבור אלגוריתמים יוריסטיים (במיוחד עצים ורשתות נוירונים), הבחירה של המימוש הקונקרטי עשויה להשפיע על איכות ההיפותזה אותה ימצא האלגוריתם.

על מנת להעריך את ההשפעה של הבחירות השונות, ואת השיקולים שיש לקחת בחשבון, נוח לפרק את $L_{\mathcal{D}}(h)$ לשלושה רכיבים:

$$L_{\mathcal{D}}(h) = \overbrace{L_{\mathcal{D}}(h) - L_S(h)}^{\text{Estimation error}} + \overbrace{L_S(h) - L_S(\mathcal{H})}^{\text{Optimization error}} + \overbrace{L_S(\mathcal{H})}^{\text{Approximation error}}$$

שגיאת ההכללה

שגיאת ההכללה היא ההפרש בין השגיאה האמיתית של ההיפותזה לבין השגיאה האמפירית שלה. הרכיב הזה יהיה קטן יותר ככל ש־$\mathcal{H}$ תהיה קטנה יותר, וככל שיהיו לנו יותר דוגמאות. עבור קלסיפיקציה בינארית, תורת ההכללה שפיתחנו מאפשרת לנו לחסום את שגיאת ההכללה באמצעות $\mathrm{VC}(\mathcal{H})$.

שגיאת הקירוב

שגיאת הקירוב היא השגיאה של ההיפותזה הטובה במחלקה. על מנת להקטין אותה, ניתן לנקוט בשתי דרכים:

- **ידע מוקדם**. אם אנחנו, או מומחה בבעיה הספציפית שעל הפרק, מצליח להצביע על מחלקה בה באמת יש היפותזה טובה, שגיאת הקירוב תהיה קטנה. לכן, לפני שמשתמשים באלגוריתמי למידה, כדאי להבין טוב את הבעיה שעומדת לפנינו, ולחשוב באיזו משפחה של פונקציות נוכל למצוא היפותזה הקרובה להיפותזה אותה אנו רוצים ללמוד.
- **שימוש במחלקה עשירה יותר**. ככל שהמחלקה אותה נבחר תכיל יותר היפותזות, שגיאת הקירוב תהיה קטנה יותר.

שגיאת האופטימיזציה

שגיאת האופטימיזציה היא ההפרש בין השגיאה האמפירית לבין השגיאה האמפירית של ההיפותזה הטובה במחלקה. אם נשתמש באלגוריתם ERM, שגיאה זו תהיה 0. למרבה הצער, ככל הנראה, עבור רב המחלקות לא ניתן לממש את אלגוריתם ה־ERM ביעילות. לכן, אנו נאלצים להשתמש ביוריסטיקות. כפועל יוצא, שגיאת האופטימיזציה עשויה להיות חיובית.

עבור אלגוריתמים המבוססים על תחליף קמור, ניתן לחסום את הסכום של שגיאת האופטימיזציה ושגיאת האפרוקסימציה ע"י שגיאת האפרוקסימציה על התחליף הקמור. לכן, במקרה של כשלון, אנו לכל הפחות נדע שהתחליף הקמור לא מספיק עשיר. למרבה הצער, עבור יתר אלגוריתמי הקלסיפיקציה, לא קיימות כיום שיטות המאפשרות לנתח את שגיאת האופטימיזציה, והיא עשויה להשתנות מאוד כאשר עוברים בין מימושים שונים של אלגוריתמים שונים. התורה הקיימת כיום לא מאפשרת לנו לעשות ניתוח מושכל של שגיאת האופטימיזציה, ובפועל מסתמכים הרבה על נסיון העבר. למשל:

- ברשתות נוירונים, ככל שהרשת עמוקה יותר, שגיאת האימון תגדל. נעיר שכיום מצליחים להגיע לביצועים טובים עם רשתות בעומק עד 20 פחות או יותר.
- הנסיון מהשנים האחרונות מראה שהרבה פעמים SGD עובד טוב יותר מ־GD כאשר משתמשים ברשתות נוירונים.
- כמו כן, פרמוטציות אקראיות של המשקלות בזמן האימון לעיתים משתפרות את שגיאת האימון.

הערות נוספות

נעיר מספר הערות נוספות שכדאי לקחת בחשבון כאשר משתמשים בלמידה:

- **ניסוי וטעיה**. אידאלית, היינו רוצים שיהיה לנו אלגוריתם למידה אחר, שנדע שהוא "הטוב ביותר" או לכל הפחות לא רחוק מהטוב ביותר. כאמור, זה רחוק מלהיות המצב, ולאלגוריתמים\פרמטרים\מימושים שונים יש יתרונים שונים. עבור בעיות למידה פשוטות, שימוש סטנדרטי באלגוריתם פשוט (נאמר, SVM ללא גרעין) עשוי להניב תוצאות טובות. עבור בעיות קשות יותר, מבין אוסף הבחירות שניתן לעשות כאשר משתמשים בלמידה, ככל הנראה הבחירה הראשונה שנעשה לא תהיה הטובה ביותר. לכן, השימוש בלמידה, בשונה משימוש באלגוריתמים, מצריך לא מעט ניסוי וטעיה.

- **ולידציה**. חלק מהפרמטרים של האלגוריתמים הם מורכבים למדי (למשל, המבנה של רשת הנוירונים). לעומת זאת, חלק מהפרמטרים הם פשוטים ו"חד מימדיים" (למשל, פרמטר הרגולריזציה, או מספר האיטרציה ב־AdaBoost). במקרים כאלו ניתן להשתמש בולידציה על מנת לבחור את הערך המתאים לפרמטר.

- **התאמת יתר** $(Overfit)$ **ובדיקת ההיפותזה הסופית**. לאחר תהליך הלמידה, כדאי לנו לבדוק את ההיפותזה שיצרנו. לעיתים משתמשים באלגוריתמים ללא חסם הכללה, או עם חסם הכללה גרוע, ובמקרה כזה יש סכנה לביצוע התאמת יתר. גם אם משתמשים באלגוריתמים עבורים מובטח שלא נעשה התאמת יתר, לעיתים בכל זאת עושים התאמר יתר הנובעת, למשל, מהעובדה שאנו משתמשים שוב ושוב באלגוריתמים שונים על אותו מדגם אימון. מחסם הופדינג, על מנת לשערך את $L_{\mathcal{D}}(h)$ עד כדי ε, אנו זקוקים ל־$\left(\frac{1}{\varepsilon^2}\right)$ דוגמאות; נעיר שהדוגמאות הללו צריכות להיות **דוגמאות חדשות, ובלתי תלויות** בדברים שעשינו קודם - בפרט, החסם אינו תקף כאשר משתמשים בדוגמאות שאימנו עליהן את האלגוריתם, או אפילו השתמשנו בהן על מנת לבדוק\לאמן אלגוריתמים אחרים בתהליך של ניסוי וטעיה.
נעיר שבפועל לא משתמשים כל פעם בדוגמאות חדשות, לכן, על מנת להשאיר את תהליך הבדיקה אמין, בדר"כ משתמשים במדגם בדיקה גדול יחסית (כ־50־10 אחוז מכלל הדוגמאות).

- **דרישות נוספות מההיפותזה הנלמדת**. לעיתים, אנו נרצה למצוא היפותזה עם תכונות מסויימות, מעבר להיותה היפותזה עם שגיאה נמוכה. למשל, היפותזה שיהיה ניתן להעריך במהירות, או שתתפוס מעט מקום בזיכרון. שיטות מסויימות, למשל שיטות הלומדות מפרידים לינארים דלילים כמו AdaBoost, מייצרות היפותזות כאלו.

חלק III
למידת ייצוג

מה הלאה?

עד כה לא דנו בדרך בה הקלטים מיוצגים, כלומר, הנחנו שהמרחב X הינו נתון. סופסוף הגיע הזמן!

ליצוג של הקלטים יש השפעה רבה על הביצועים של ההיפותזה אותה נלמד. ככל, ייצוג טוב יותר, המדגיש את החלקים ה"חשובים" של הקלטים (ללא מידע "עודף"), יאפשר לימוד טוב יותר: אם נתון לנו יצוג כזה, יותר סביר שלהיפותזה פשוטה (כלומר כזו שהמגיעה ממחלקה קטנה ופשוטה) יהיו ביצועים טובים.

כיצד נייצר יצוג טוב? ובכן, דרך אחת היא להשתמש במומחיות ובידע מוקדם. כלומר, לתת למומחה לבנות יצוג "טוב" של הקלטים, לדוגמא:

- עבור קלטים טקסטואלים, הרבה פעמים Bag-of-Word מהווה יצוג טוב.
- עבור קלטים שהם קבצי שמע, הרבה פעמים יצוג טוב מבוסס על פירוק לתדרים.

אנו כמעט לא נדבר בקורס על יצוגים שהמתאימים לתחומים ספציפיים, שכן דיון בהם שייך לתחום הקונקרטי (זיהוי שפה, זיהוי דיבור, ראיה ממוחשבת, וכו') אליו הם שייכים. עם זאת, מלבד שימוש במומחיות, לעיתים ניתן להשתמש בקלטים עצמם על מנת **ללמוד** יצוג טוב.

כלומר, בהנתן הרבה קלטים:

$$x_1, ..., x_m \in X$$

אנו נרצה ללמוד מיפוי $\Psi : X \to X'$ כך שאם ניצג כל קלט $x \in X$ ע"י $\Psi(x) \in X'$, נקבל יצוג "טוב" של הקלטים. כלומר, X' יהיה מרחב פשוט יותר מהמרחב המקורי (ממימד נמוך / עם מספר קטן של נקודות / ...), וההעתקה Ψ תשמר את החלקים ה"מהותיים" של נקודות המדגם, ולא "תאבד מידע חשוב". השבוע ובשבוע הבא נלמד על דרכים ללמוד מיפויים כאלו.

נעיר מספר הערות לפני כן:

1. בלמידת יצוג, אנו מסתכלים על דוגמאות **לא מתויגות**. כלומר על איברים ב־X, ולא ב־$X \times Y$. הרבה פעמים, דוגמא לא מתויגת תהיה הרבה יותר זולה מדוגמא מתויגת:

   (א) בראיה ממוחשבת, ניתן למצוא ברשת בקלות מיליארדי תמונות. לעומת זאת, יקר יותר להשיג תמונות מתויגות.

   (ב) באופן דומה, בעיבוד שפה, יש המון טקסט ברשת ובמקומות אחרים, אך הרבה פחות טקסט מתויג.

   (ג) בביולוגיה, לעיתים, על מנת לקבל תיוג יש לבצע ניסוי.

   לאור זאת, אחת המוטיבציות ללמידת יצוג היא ללמוד גם מהדוגמאות הלא מתויגות, כלומר, להשתמש בהן על מנת לשפר את תהליך הלימוד. מכאן, לימוד יצוג לפעמים נקרא **למידה לא מפוקחת** (unsupervised learning), בשונה ממה שלמדנו עד כמה, הנכנס לקטגוריה של **למידה מפוקחת** (**supervised learning**).

2. בכל האלגוריתמים שנלמד, X יהיה $\mathbb{R}^n$, ונסמן ב־$d(x,y) = \|x-y\|$ את המרחק האוקלידי. כמו כן, כשנדבר על מדגם, הכוונה תהיה ל**מדגם לא מתוייג**

$$S = \{x^1, ..., x^m\} \subset \mathbb{R}^n$$

3. בדומה ללמידה מפוקחת, ניתן לדבר על מודל פורמאלי הדומה למודל PAC. אנו לא נעשה זאת, גם מפאת חוסר זמן, וגם בגלל שבניגוד ללמידה מפוקחת, עדיין אין מודל סטנדרטי.

ניתן לחלק את הגישות ללמידת יצוג לארבעה סוגים:

1. **טרנספורמציות פשוטות**. דרך פשוטה אך לעיתים אפקטיבית ללמוד יצוג היא ע"י הפעלת טרנספורמציה פשוטה (בדר"כ לינארית) על $\mathbb{R}^n$, הגורמת להתפלגות להיות "נקיה ומסודרת". דוגמא בסיסית היא **מרכוז וסטנדריזציה**. נקבע קואורדינטה $j \in [n]$. נסמן ב־

$$\mu_j = \frac{1}{m}\sum_{i=1}^{m} x_j^i, \;\; \sigma_j = \sqrt{\sum_{i=1}^{m} \left(x_j^i - u_j\right)^2}$$

את תוחלת וסטיית התקן (האמפירית) של הקואורדינטה ה־j. שימו לב שהגדלים הללו נלמדים מהמדגם הלא מתויג שבידינו. נגדיר:

$$\Psi(x) = \left(\frac{x_1 - \mu_1}{\sigma_1}, .., \frac{x_n - \mu_n}{\sigma_n} \right)$$

הפעלת Ψ על המדגם מייצרת מדגם בו לכל קואורדינטה יש תוחלת 0 וסטיית תקן 1. לכן, במובן מסויים, כל הקואורדינטות הומוגניות, ואין קואורדינטות שיותר "בולטות" מהאחרות.

2. **הצברה** (**Clustering**). בשימוש בהצברה עבור למידת יצוג, אנו מחפשים אוסף של $m \gg k$ נקודות (הנקראות **מרכזים**) $c^1, ..., c^k \in \mathbb{R}^n$ המיצגות טוב את המדגם. כלומר, לרב הדוגמאות x^i קיים מרכז c^j הקרוב ל־x^i. אוסף כזה של מרכזים מוביל ליצוג קומפקטי של X ע"י מספר סופי וקטן (k) של נקודות.

3. **הורדת מימד** (**Dimension Reduction**). בהורדת מימד אנו מחפשים תת מרחב לינארי $V \subset \mathbb{R}^n$ ממימד k, כאשר $k \ll d$, המיצג טוב את המדגם. למשל, מרחב V בו לרב הדוגמאות x^j קיים $v \in V$ הקרוב ל־x^j. מרחב V כזה מוביל ליצוג קומפקטי של X ע"י מרחב לינארי ממימד נמוך.

4. **למידת מילון** (**Dictionary Learning**). בלמידת מילון אנו מחפשים אוסף של $m \gg k$ נקודות (הנקראות **מילים**) $v_1, ..., v_k \in \mathbb{R}^n$ המיצגות טוב את המדגם במובן שרב הדוגמאות x^j הן בקירוב צירוף לינארי של מספר קטן של מילים.

אנו נלמד בעיקר הצברה, הורדת מימד ולמידת מילון.

12 הצברה - אלגוריתם ה־k-means

יהא נתון מדגם $S = \{x^1, ..., x^m\}$. אלגוריתמי k-means מנסים כאמור למצוא k מרכזים $C = \{c^1, ..., c^k\} \in \mathbb{R}^n$ הממזערים את הסכום (או, באופן שקול, ממוצע) ריבועי המרחקים של נקודות המדגם מהמרכז הקרוב ביותר אליהן. קונקרטית, לכל נקודה $x \in \mathbb{R}^n$ נסמן ב:

$$d(x, C) := \min_{1 \le j \le k} d\left(x, c^j\right)$$

את המרחב בין x לבין המרכז הקרוב ביותר ל־x. אלגוריתמי $k - means$ מנסים למזער את פונקצית המטרה:

$$L^{k-means}(C) := \sum_{i=1}^{m} d^2\left(x^i, C\right)$$

הבעיה החישובית של מציאת אוסף מרכזים C הממזער את $L^{k-means}(C)$ הינה בעיה קשה חישובית. לבעיה קיימים אלגוריתמי קירוב, אך אנו נלמד על יוריסטיקה פופולארית, שלעיתים נקראת בעצמה k-means (למעשה, וריאנט של היוריסטיקה שנלמד, הנקרא k-means++, הוא אלגוריתם קירוב עם יחס קירוב $O(\log(k))$). היוריסטיקה הנ"ל תחזיק אוסף של מרכזים:

$$C = \{c^1, ..., c^k\} \subset \mathbb{R}^n$$

כמו כן, חלוקה של המדגם ל־k תתי קבוצות:

$$S = C_1 \dot{\cup} ... \dot{\cup} C_k$$

הקבוצה (**צביר** \ **cluster**) C_j תכיל את הנקודות המתאימות למרכז c^j, כלומר, את אוסף נקודות המדגם שהמרכז הקרוב ביותר אליהן הוא c^j. היוריסטיקה תתבסס על שיטה הנקראית **Alternate Minimization** (עוד נפגוש אותה פעמיים בשלושת השבועות הקרובים): האלגוריתם יתחיל עם מרכזים אקראיים, ואז, בכל שלב, יבצע:

- **אופטימציה על החלוקה**: האלגוריתם יעדכן את החלוקה כך שלכל j, C_j תכיל את הנקודות המקיימות $d\left(x, c^j\right) = d(x, C)$.
- **אופטימיזציה על המרכזים**: האלגוריתם יעדכן את המרכזים כל שלכל j, c^j ימזער את סכום ריבועי המרחקים בינו לבין הנקודות ב־C_j.

כיצד יתבצע כל אחד מהשלבים?

ובכן, בהנתן המרכזים, ברור כיצד לעדכן את החלוקה - פשוט נשים ב־C_j את כל הנקודות שהמרכז הקרוב ביותר אליהן הוא c_j. מה לגבי עדכון המרכזים בהנתן החלוקה? על מנת לעדכן את המרכז ה־j עלינו למצוא $c^j \ni \mathbb{R}^n$ הממזער את:

$$\sum_{x \in C_j} \left\| c^j - x \right\|^2$$

למרבה המזל, לבעית האופטימיזציה הזו קיים פתרון סגור - c_j הוא פשוט ממוצע הנקודות ב־C_j (תרגיל). נסכם:

אלגוריתם 11 k-means

פרמטרים: מספר מרכזים k, מספר איטרציות T.
קלט: מדגם $S = \left\{x^1, ..., x^m\right\} \subset \mathbb{R}^n$

1. עבור על $j = 1, ..., k$:

   (א) בחר את c_j להיות דוגמא אקראית מתוך S (אתחול רנדומי לכל המרכזים)

2. עבור $t = 1, 2, ..., T$:

   (א) לכל $k \in [k]$, עדכן:

$$C_j = \left\{ x^i \left| d\left(x^i, c^j\right) = d\left(x^i, \left\{c^1, ..., c^k\right\}\right) \right. \right\}$$

   (ב) לכל $j \in [k]$, עדכן:

$$c^j = \frac{1}{|C_j|} \sum_{x^i \in C_j} x^i$$

3. החזר את $c^1, ..., c^k$

12.0.2 הערות והדגמות

בסיכומי הקורס מופיעות תמונות המראות דוגמא של הרצת האלגוריתם עם שני מרכזים ב־$\mathbb{R}^2$, עם שינוי קל ממה שאנחנו למדנו: בשלב האתחול בהרצה זו הרשו למרכזים לא להיות מהמדגם עצמו.

כמו כן, מופיעה דוגמא של הרצה על MNIST - מאגר מידע של תמונות של ספרות הכתובות בכתב יד. בסיכומי הקורס מופיעים המרכזים שהתקבלו מהרצת האלגוריתם על מאגר מידע זה - שלוש פעמים 9, פעם יחידה $1, 3, 2, 0, 8$ ופעמיים הספרה 6. נשים לב כי המרכזים שנמצאו אינם מייצגים מספיק טוב את המדגם - למשל, הספרות 5 ו־7 כלל לא מיוצגות. יכול להיות שאם היינו משתמשים ביותר מרכזים (נגיד, 100) כבר היינו מקבלים יצוג טוב. בהקשר זה, נעיר כי הרבה פעמים הדוגמאות עצמן מאוד מורכבות, ולא סביר שנוכל ליצגן באמצעות מספר קטן של מרכזים. במקרים כאלו עדיין יכול להיות מועיל לעשות הצברה על **חלקים של הדוגמאות**. למשל, אם הדוגמאות הן תמונות בגודל 100×100, ניתן לעשות הצברה על תמונות בגודל 10×10, המהוות ריבוע בתמונות המקוריות, ואז ניתן לקבל יצוג של התמונות המקוריות למשל באופן הבא: נחלק כל תמונה במדגם המקורי ל־100 תמונות בגודל 10×10, וניצג כל תת־תמונה באמצעות המרכז המתאים.

אתחול ראשוני ו־k-means++ אנו הצענו דרך מסויימת לאיתחול המרכזים (פשוט לבחור k דוגמאות באקראי). כפי שראינו בדוגמא, ישנן דרכים נוספות לאיתחול הנקודות. למשל, אלגוריתם ה־k-means++ זהה לאלגוריתם שהצגנו מעלה, מלבד האתחול, שיתבצע באופן הבא:

אלגוריתם 12 k-means++ initialization

פרמטרים: מספר מרכזים k.
קלט: מדגם $\mathbb{R}^n \supset \{x^1, ..., x^m\} = S$

1. אתחל $C = \phi$.
2. עבור $t = 1, ..., k$:
   (א) בחר דוגמא ב־S באקראי, כך שהסיכוי לבחור את x_i הינו $\frac{d^2(x_i,C)}{\sum_{j=1}^m d^2(x_j,C)}$
   (ב) הוסף את הדוגמא שנבחרה ל־C.
3. החזר את C.

נשים לב שבשלב ה־t, שטת האתחול הנ"ל תבחר בסיכוי גבוה יותר דוגמאות ללא מרכז קרוב. ניתן לראות שאם C הוא הפלט של האתחול, ו־C^* היא קבוצת k המרכזים האופטימלית, אז:

$$\mathbb{E}\left[L^{k-means}(C)\right] \leq O(\log(k)) L^{k-means}(C^*)$$

כאשר התוחלת היא על פני האקראיות הפנימית של האלגוריתם. לכן, אם משתמשים בשיטת האתחול הנ"ל, אנו מקבלים אלגוריתם קירוב עם יחס קירוב $O(\log(k))$.

12.1 אלגוריתמים נוספים ופונקציות מטרה נוספות

בנוסף, ישנם עשרות אלגוריתמי הצברה מעבר ל־k-means, ונראה חלק מהם בתרגול. כמו כן, קיימות פונקציות מטרה נוספות אותן אלגוריתמי ההצברה מנסים למזער כאשר הם מחפשים מרכזים, לדוגמא:

$$L^{k-medians}(C) := \sum_{i=1}^{m} d\left(x^i, C\right)$$
$$L^{k-center}(C) := \max_{i\in[m]} d\left(x^i, C\right)$$

12.2 הצברה מעבר ללמידת יצוג - מציאת חלוקה בעלת משמעות

אנו הצגנו אלגוריתמי הצברה בתור כלי המוצא יצוג טוב, ע"י קירוב כל הנקודות במדגם באמצעות מספר קטן של נקודות. דרך אחרת, אפילו יותר מקובלת, להסתכל על הצברה היא בתור כלי המוצא חלוקה בעלת משמעות של אוסף אובייקטים. מציאה של חלוקות כאלו שימושית כמעט בכל תחום, למשל:

- **ביולוגיה**: ניתן לעשות הצברה על מנת למצוא חלוקה לסוגים של צמחים \ בע"ח \ גנים \ חלבונים \ ...
- **רשתות חברתיות**: ניתן להשתמש בהצברה על מנת למצוא קהילות
- **פסיכולוגיה**: ניתן להשתמש בהצברה על מנת לחלק בני אדם לטיפוסים על סמך התכונות שלהם
- **עיבוד תמונה**: בהנתן תמונה, ניתן להשתמש בהצברה על מנת חלק את התמונה לאובייקטים

13 הורדת מימד

נתון מדגם $\mathbb{R}^n \supset \{x_1, ..., x_m\} = S$[23]. שיטות הורדת מימד מחפשות תת מרחב V ממימד $k \ll n$ והעתקה $\Psi : \mathbb{R}^n \to V$ המהווה יצוג טוב של המדגם. בהנתן יצוג כנ"ל, אנו יכולים ליצג כל דוגמא x ע"י k מספרים בלבד (המקדמים $\Psi(x)$ בהצגתו כצירוף לינארי לפי בסיס של V). יצוג קומפקטי כזה מאפשר לעבוד עם מחלקות עשירות יותר, ועשוי לשפר משמעותית את זמן האימון, ואת זמן הריצה של ההיפותזה הנלמדת.
אנו נלמד שתי שיטות להורדת מימד:

- הראשונה נקראת PCA ומוצאת ת"מ $V \subset \mathbb{R}^n$ שהוא הקרוב ביותר (במובן מסוים) לנקודות המדגם, ומגדירה את $\Psi : \mathbb{R}^n \to V$ להיות ההטלה האורתוגונלית על V.

[23] לנושא הזה היה עדכון בהרצאה הבאה - לכן אני משלבת כאן בין השניים.

- השיטה השניה שנלמד נקראת "הטלות מקריות", ובה אנו פשוט לוקחים העתקה לינארית אקראית $\Psi : \mathbb{R}^n \to \mathbb{R}^k$. למרבה הפלא, השיטה הנ"ל מובילה לעיתים לתוצאות טובות, ועל כן מהווה חלופה מהירה ל־PCA.

13.1 ניתוח גורמים ראשיים $(\mathbf{PCA})$

יהא נתון מדגם $S = \{x_1, ..., x_m\} \subset \mathbb{R}^n$. נסמן ב:

$$P_V : \mathbb{R}^n \to V$$

את **ההטלה האורתוגונלית על** V.
נזכר בכמה עובדות בסיסיות מאלגברה לינארית:

טענה 13.1

1. לכל $x \in \mathbb{R}^n$, $P_V(x)$ היא הנקודה ב־V הקרובה ביותר (לפי מרחק אוקלידי) ל־x, כלומר מתקיים:
$$d(x, P_V(x)) = \min_{v \in V} d(x, v) := d(x, V)$$
2. P_V היא העתקה לינארית.
3. אם $M_{n \times k} \ni U$ היא מטריצה שעמודותיה מהוות בסיס א"נ של V אזי:
$$P_V(x) = UU^T x$$

אלגוריתם ה־PCA מחפש בסיס למרחב V ממימד k הממזער את סכום (או, באופן שקול, ממוצע) ריבועי המרחקים של נקודות המדגם מ־V. כלומר, מרחב הממזער את:

$$L^{\mathrm{PCA}}(V) := \sum_{i=1}^{m} d^2(x_i, V) = \sum_{i=1}^{m} d^2(x_i, P_V(x_i))$$

דוגמא ויזואלית מופיעה בסיכומי ההרצאה.
למרבה הפלא, ניתן למצוא ביעילות תת מרחב V הממזער את ממוצע ריבועי המרחקים. הדרך בה עושים זאת מסתמכת על אלגברה לינארית, או באופן יותר קונקרטי, התורה של לכסון אורתוגונלי של מטריצות סימטריות. נסמן:

$$A = \sum_{i=1}^{m} x_i x_i^T \in M_{n \times n}$$

המטריצה A הינה מטריצה סימטרית מוגדרת חיובית למחצה[24] (תרגיל). מאלגברה לינארית, אנו יודעים שקיים בסיס אורתוגונלי של ו"ע $\mathbb{R}^n \ni u_1, ..., u_n$ עם ע"ע אי שליליים $\lambda_1 \geq ... \geq \lambda_n \geq 0$. מתברר ש:

משפט 13.2 $(\mathbf{PCA})$ המרחב הנפרש ע"י $u_1, ..., u_k$ ממזער את L^{PCA} על פני כל המרחבים ממימד $\geq k$.

ההוכחה מתבססת על אלגברה לינארית, ובאופן יותר קונקרטי, המשפט הספקטרלי, לפיו למטריצה סימטרית קיים לכסון אורתוגונלי. אנו נאמר כמה מילים על ההוכחה, אך לא נוכיח את המשפט באופן מלא (ראו תרגיל רשות, וכמו כן, פרק 23 בספר של שי ושי). נסכם:

אלגוריתם 13 PCA

קלט: מדגם $S = \{x_1, ..., x_m\} \subset \mathbb{R}^n$, מימד $1 \leq k$.

1. הגדר $A = \sum_{i=1}^{m} x_i x_i^T$
2. מצא ל־A בסיס א"נ של ו"ע $\mathbb{R}^n \ni u_1, ..., u_n$ עם ע"ע $\lambda_1 \geq ... \geq \lambda_n$
3. החזר את $u_1, ..., u_k$

[24] כלומר, מתקיים $A = A^T$ ובנוסף $x^T A x \geq 0$ לכל $x \in \mathbb{R}^n$.

13.1.1 הערות

מציאת בסיס א"נ על מנת להפעיל את האלגוריתם, עלינו למצוא ל־A בסיס א"נ. נעיר שקיימים אלגוריתמים יעילים סטנדרטיים העושים זאת.

מרכוז הרבה פעמים, מועיל לעשות מרכוז של המדגם לפני שמפעילים את PCA. כלומר, להחליף את $x_1,...,x_m$ ב־$(x_1-\mu),...,(x_m-\mu)$, כאשר:

$$\mu=\frac{1}{m}\sum_{i=1}^{m}x_i$$

הפעולה הנ"ל ממרכזת את המדגם, במובן שוקטור ה־0 הופך להיות המרכז (הממוצע) של הנקודות.

11.06.2015

המטריצה A ופרשנות גיאומטרית של הגורמים הראשיים הוקטורים[25] $u_1,...,u_n$ נקראים **הגורמים הראשיים** (**Principal Components**) של המדגם. הם מאופיינים באופן הבא:

- הוקטור u_1 מגדיר את המרחב החד מימדי (כלומר, קו) עם התכונה הבאה: אם מטילים את נקודות המדגם עליו, סכום ריבועי הנורמות של הנקודות המוטלות הוא הגדולה ביותר (על פני המרחבים ממימד 1).
- עבור $1<i$, u_i מגדיר את המרחב החד מימדי עם התכונה הבאה: אם מטילים את נקודות המדגם עליו, סכום ריבועי הנורמות של הנקודות המוטלות הוא הגדול ביותר, על פני כל המרחבים ממימד 1 הניצבים ל־$span\left\{u_1,..,u_{(i-1)}\right\}$ (בסיכומי הקורס - תמונה המדגימה את התכונה הזו).

האפיון הנ"ל נובע מהעובדה הבאה, שהוכחתה מושארת כתרגיל:

למה 13.3 תהא $A\ni M_{n\times n}$ מטריצה סימטרית. יהא $u_1,...,u_n$ בסיס א"נ של ו"ע של A עם ע"י $\lambda_1\geq...\geq\lambda_n$. אזי:

1. u_1 ממקסם את u^TAu על פני כל הוקטורים מנורמה 1.
2. לכל $1<i$, u_i ממקסם את u^TAu על פני כל הוקטורים מנורמה 1 הניצבים ל־$span\left\{u_1,...,u_{(i-1)}\right\}$.

בהקשר שלנו, $A=\sum_{i=1}^{m}x_ix_i^T$. לכן לכל וקטור יחידה $\mathbb{R}^n\ni u$ מתקיים:

$$u^TAu=\sum_{i=1}^{m}u^Tx_ix_i^Tu=\sum_{i=1}^{m}\left(u,x_i\right)^2$$

הביטוי הימני הינו סכום ריבועי ההטלות של הדוגמאות על $span\left\{u\right\}$. לכן, האפיון של הגורמים הראשיים נובע מהלמה הנ"ל.

שקילות של PCA למקסום סכום הנורמות של ההטלות, ומילה על ההוכחה יהא $\mathbb{R}^n\ni V$ ת"מ ממימד k. מתקיים:

$$\begin{aligned}L^{\mathrm{PCA}}\left(V\right) &= \sum_{i=1}^{m}\left\|x_i-P_V\left(x_i\right)\right\|^2\\ &= \sum_{i=1}^{m}\left\|x_i\right\|^2-2\left\langle x_i,P_V\left(x_i\right)\right\rangle+\left\|P_V\left(x_i\right)\right\|^2\\ &\overset{(*)}{=} \sum_{i=1}^{m}\left\|x_i\right\|^2-2\left\langle P_V\left(x_i\right),P_V\left(x_i\right)\right\rangle+\left\|P_V\left(x_i\right)\right\|^2\\ &= \sum_{i=1}^{m}\left\|x_i\right\|^2-\left\|P_V\left(x_i\right)\right\|^2\end{aligned}$$

כאשר $(*)$ נובע מכך שעבור הטלות $P:\mathbb{R}^n\to\mathbb{R}^n$:

$$\forall x,\ \left\langle x,P\left(x\right)\right\rangle=\left\langle P\left(x\right),P\left(x\right)\right\rangle$$

[25] מכיוון ולא נכחתי בשיעור זה, החומר נלקח ישירות מהסיכום והמצגת שהעלה מרצה הקורס, עם עריכות קטנות שלי.

נשים לב שהביטוי $\sum_{i=1}^{m}\|x_i\|^2$ אינו תלוי ב־V. לכן, מציאת V הממזער את $L^{\mathrm{PCA}}(A)$ שקולה למציאת V הממקסם את $\sum_{i=1}^{m}\|P_V(x_i)\|^2$, כלומר, מציאת מרחב V הממקסם את סכום הריבועים של ההטלות של נקודות המדגם עליו.

מהפסקה הקודמת נובע שעבור $k=1$, הבחירה האופטימלית היא $V=span\{u_1\}$. על מנת להוכיח את משפט ה־PCA, מראים שעבור k כלשהוא, הבחירה האופטימלית היא $V=span\{u_1,...,u_k\}$.

דוגמא הדגמה של הטלה באמצעות PCA של אוסף תמונות בשחור לבן מופיעה בסיכומי הקורס; היא מראה כי לאחר הורדת המימד באופן די דרסטי (מ־$\mathbb{R}^{2500}$ ל־$\mathbb{R}^{10}$) היצוג החדש עדיין טוב, ולא הרבה מידע אבד כתוצאה מהמעבר (התמונות נראות דומות).

13.2 הטלות מקריות

יהא נתון מדגם $\{x^1,...,x^m\}\subset\mathbb{R}^n$. דרך נאיבית למצוא יצוג של המדגם ע"י מרחב וקטורי ממימד $k\ll n$ היא לבחור מטריצה אקראית $W\in M_{k\times n}$, ולקחת בתור יצוג את ההעתקה (הלינארית) $\Psi:\mathbb{R}^n\to\mathbb{R}^k$ המוגדרת ע"י $\Psi(x)=Wx$.

השיטה הנ"ל נראית מאוד נאיבית - בבחירת היצוג אנו בכלל לא מסתכלים על המדגם!

למרות זאת, אנו נראה שבהסתברות גבוהה אנו מקבלים יצוג של המדגם המקורי, **ללא עיוות גדול** של המרחקים. כלומר, עבור כל i,j מתקיים:

$$d\left(x^i,x^j\right)\approx d\left(\Psi\left(x^i\right),\Psi\left(x^j\right)\right) \tag{1}$$

לכן, לעיתים, הטלות מקריות יכולות להוות תחליף מהיר ופשוט ל־PCA.

ניגש לתיאור יותר מפורט.

ראשית, הדרך בה נבחר את W היא ע"י כך שהרכיבים $\{W_{i,j}\}_{i\in[k],j\in[n]}$ יהיו משתנים מקריים בלתי תלויים המתפלגים נורמלית עם תוחלת 0 ושונות $\frac{1}{k}$. נזכיר ש:

- משתנה מקרי נורמלי עם תוחלת μ ושונות σ^2 הוא מ"מ הם פונקצית צפיפות:

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

- אם $X_1,...,X_n$ הם מ"מ נורמליים בלתי תלויים עם תוחלות $\mu_1,...,\mu_n$ ושונויות $\sigma_1^2,...,\sigma_n^2$, אז לכל $\mathbb{R}\ni\alpha_1,...,\alpha_n$ המשתנה המקרי:

$$\alpha_1X_1+...+\alpha_nX_n$$

הוא מ"מ נורמלי עם תוחלת $\alpha_1\mu_1+...+\alpha_n\mu_n$ ושונות $\alpha_1^2\sigma_1^2+...+\alpha_n^2\sigma_n^2$.

ניגש כעת להראות שמתקיים (10) בהסתברות גבוהה (על פני בחירת W). נראה זאת קודם עבור זוג בודד, ואח"כ נשתמש בחסם האיחוד על מנת לעבור לכל הזוגות.

למה 13.4 לכל זוג דוגמאות x^i,x^j:

$$\Pr_W\left(\left|\frac{d^2\left(\Psi\left(x^i\right),\Psi\left(x^j\right)\right)}{d^2\left(x^i,x^j\right)}-1\right|>\varepsilon\right)\le 2\cdot\exp\left(-\frac{k\varepsilon^2}{6}\right)$$

אנו נסתמך על הלמה הבאה (ראו הוכחה בפרק $B.7$) בספר, המהווה מעין אנלוג לחסם הופדינג, עבור ריבועים של מ"מ נורמליים:

למה 13.5 יהיו $Z_1,...,Z_k$ מ"מ נורמליים ב"ת ש"ה עם תוחלת 0. נסמן:

$$Z=\sum_{i=1}^{k}Z_i^2,\ \sigma^2=\mathbb{E}[Z]$$

אזי:

$$Pr\left(\left|\frac{Z}{\mathbb{E}[Z]}-1\right|>\varepsilon\right)\le 2\cdot\exp\left(-\frac{k\varepsilon^2}{6}\right)$$

נעבור להוכחת הלמה הקודמת: **הוכחה:** נסמן $x = x^i - x^j$. מסימון זה, מתקיים $d\left(x^i, x^j\right) = \|x\|$. כמו כן:

$$d\left(\Psi\left(x^i\right), \Psi\left(x^j\right)\right) = \left\|Wx^i - Wx^j\right\| = \|Wx\|$$

לכן, די להראות שמתקיים:

$$Pr\left(\left|\frac{\|\Psi(x)\|^2}{\|x\|^2} - 1\right| > \varepsilon\right) \leq 2 \cdot \exp\left(-\frac{k\varepsilon^2}{6}\right) \tag{2}$$

נכתוב:

$$\Psi(x) = \begin{pmatrix} \Psi_1(x) \\ \vdots \\ \Psi_k(x) \end{pmatrix} = \begin{pmatrix} W_{11}x_1 + ... + W_{1n}x_n \\ \vdots \\ W_{k1}x_1 + ... + W_{kn}x_n \end{pmatrix}$$

מכיוון ו־x קבוע, הקואורדינטות של $\Psi(x)$ הן ב"ת. כמו כן, כל אחת מהן היא צירוף לינארי של מ"מ נורמלים ב"ת, ולכן כל קואורדינטה היא מ"מ נורמלי עם תוחלת (בה"כ נביט בשורה הראשונה):

$$x_1\mathbb{E}[W_{11}] + ... + x_n\mathbb{E}[W_{1n}] = 0$$

ושונות:

$$var\left(\sum_{i=1}^n x_i W_{1i}\right) = \sum_{i=1}^n var\left(x_i W_{1i}\right) = \sum_{i=1}^n x_i^2 var\left(W_{1i}\right) = \sum_{i=1}^n x_i^2 \frac{1}{k} = \frac{\|x\|^2}{k}$$

דהיינו, $\Psi(x) = \begin{pmatrix} \Psi_1(x) \\ \vdots \\ \Psi_k(x) \end{pmatrix}$ הוא וקטור מקרי עם קואורדינטות ב"ת שכל אחת מתפלגת נורמלית עם תוחלת 0 ושונות $\frac{\|x\|^2}{k}$. בפרט:

$$\mathbb{E}\left[\|\Psi(x)\|^2\right] = \sum_{i=1}^k \mathbb{E}\left[\|\Psi_i(x)\|^2\right] = \sum_{i=1}^k \frac{\|x\|^2}{k} = \|x\|^2$$

ומלמת העזר נקבל כי (11) מתקיים, כנדרש.

■

כעת, מחסם האיחוד, נובע שכאשר $k \gg \log(m)$ מתקיים (10):

משפט 13.6 (ג'ונסון־לינדנשטראוס): נסמן $\varepsilon = \sqrt{\frac{6\log\left(\frac{m^2}{6}\right)}{k}}$. אזי, בהסתברות $1 - \delta \leq$ על פני הבחירה של W מתקיים:

$$\forall i, j \in [m], \ \left|\frac{d^2\left(\Psi\left(x^i\right), \Psi\left(x^j\right)\right)}{d^2\left(x^i, x^j\right)} - 1\right| \leq \varepsilon$$

14 הצצה ללמידת מילון

יהא נתון מדגם $S = \{x_1, ..., x_m\} \subset \mathbb{R}^n$. בלמידת מילון מחפשים $m \gg K$ נקודות (הנקראות **מילים**) $\in \mathbb{R}^n$ $v_1, ..., v_K$ המיצגות טוב את המדגם במובן שרב הדוגמאות הן, בקירוב, צירוף לינארי של מספר קטן, $k \ll K$, של מילים. כלומר, רב הדוגמאות מקיימות:

$$x_j \approx \sum_{i=1}^K \alpha_i^{(j)} v_i$$

עבור וקטור $\left(\alpha_1^{(j)},...,\alpha_K^{(j)}\right) \ni \mathbb{R}^K$ עם $k \geq$ קואורדינטות שאינן 0.

בדומה ל־k-means, רב האלגוריתמים הלומדים מילון מבוססים על Alternate Minimization: הם מחזיקים מילון $v_1,...,v_K$, וכמו כן מקדמים $\left(\alpha_1^{(j)},...,\alpha_K^{(j)}\right)$ לכל דוגמא. האלגוריתמים מתחילים עם בחירה אקראית של המילון, ואז, בכל שלב, מבצעים:

- **אופטימציה על המקדמים**: בהנתן המילון הנוכחי, לכל דוגמא x_j, מעדכנים את $\left(\alpha_1^{(j)},...,\alpha_K^{(j)}\right)$ כך שלמשל ימזער את:

$$\left\| x_j - \sum_{i=1}^{K} \alpha_i^{(j)} v_i \right\|^2$$

על פני כל הוקטורים $(\alpha_1,...,\alpha_K) \ni \mathbb{R}^K$ עם $k \geq$ קואורדינטות שאינן 0.

- **אופטימיזציה על המילון**: בהנתן המקדמים הנוכחיים, מעדכנים את המילון כך שלמשל ימזער את:

$$\sum_{j=1}^{m} \left\| x_j - \sum_{i=1}^{K} \alpha_i^{(j)} v_i \right\|^2$$

נעיר שבדומה ל־k-means, הבעיה של מציאת המילון האופטימלי הינה בעיה קשה. למעשה, הקושי שלה חמור אפילו יותר: ב־k-means, בהנתן המרכזים, קל לחשב את היצוג של וקטור חדש $x \in \mathbb{R}^n$ (היצוג הוא פשוט המרכז הקרוב ביותר). בלמידת מילון לעומת זאת, אפילו בהנתן המילון, קשה לחשב את היצוג של וקטור $x \in \mathbb{R}^n$ ביחס למילון. כלומר, קשה למצוא מקדמים $(\alpha_1,...,\alpha_K) \ni \mathbb{R}^K$ עם $k \geq$ קואורדינטות שאינן 0 הממזערים את $\left\| x_j - \sum_{i=1}^{K} \alpha_i^{(j)} v_i \right\|$.

לעובדה הנ"ל יש שתי השלכות:

- לאחר שלמדנו את המילון, בהנתן דוגמא חדשה $x \in \mathbb{R}^n$, היצוג שלה יחושב **באופן יוריסטי**. למשל, ע"י האלגוריתם החמדני המתחיל עם $(\alpha_1,...,\alpha_K) = (0,...,0)$, ואז מבצע k עדכונים, שבכל אחד מהם משנים את הערך של קואורדינטה בודדת, באופן אופטימלי מבין כל השינויים הללו.
- במהלך למידת מילון, בשלב האופטימיזציה של המקדמים, עלינו לחשב את היצוג של הדוגמאות לפי המילון הנוכחי. החישוב הנ"ל אף הוא יבוצע ע"י יוריסטיקה, בד"כ אותה יורידטיקה בה נשתמש בהמשך על מנת לחשב דוגמאות חדשות.

לסיום, נעיר שבניגוד לשלב של אופטימיזציה על המקדמים בהנתן המילון, ניתן לבצע ביעילות את השלב של אופטימיזציה על המילון בהנתן המקדמים, שכן מדובר בבעיה קמורה.

חלק IV
נושאים נוספים

15 למידת אונליין

18.06.2015

היום נלמד את הנושא האחרון שנלמד בצורה מסודרת[26]. נושא זה הוא סופר־חשוב שלא נספיק לתת לו מספיק תשומת לב.

עד כה - למידת Batch:

- הפרד ל־2\3 שלבים:
  - שלב האימון
  - שלב אפשרי של ולידציה
  - שלב טסט
- מניחים שקיימת התפלגות (לא ידועה) $\mathcal{D}$ מעל $X \times Y$ כך שגם מדגם האימון וגם מדגם הטסט נוצרים ממנה באופן בלתי תלוי.
- ה־setup הזה מתאים להרבה אפליקציות: זיהוי פרצופים\ספרות (שכפי שראינו בתרגילים), זיהוי טקסט...
- כאן יש לנו שפע דוגמאות מתוייגות לספק לאלגוריתם. ההנחה שמנחה אותנו היא ששום דבר לא משתנה יותר מידי - הדרך בה אנחנו מזהים פרצופים היום לא אמורה להשתנות בשנים הקרובות.

מודל זה פחות מתאים כשהסביבה משתנה, למשל בחיזוי מזג אוויר למשל, אנחנו רוצים כל יום תחזית,והתשובה תשפט לכל יום בצורה שונה, כי מזג האוויר משתנה. למטרות שכאלו, אנו נזקקים למודל שונה:

למידת אונליין

- הלומד מקבל החלטות באופן סדרתי, ומשלם "מחיר" (loss) בכל איטרציה שכזו (בה הוא מבצע החלטה), כאשר מחיר זה נקבע ע"י הסביבה.
- תהליך הלמידה מתואר כ"משחק חוזר" בין הלומד לבין הסביבה.
- בניגוד ל־setup הקודם, אין הנחה סטטיסטית על האופן בו מיוצרות הדוגמאות. בפרט, יתכן והסביבה "תתנהג" כיריב - הדוגמאות יבחרו באופן מסויים כדי "להפיל" את הלומד.

נפרמל את המודל:

15.1 Online classification

בשביל להתמודד עם הסיטואציה הנ"ל, נצטרך להגדיר מודל שונה מזה שהכרנו עד כה. נעשה זאת:

$$X - domain,\ Y = \{\pm 1\},\ \mathcal{H} \subseteq \{\pm 1\}^X$$

לצורך פישוט, אנו נתמקד במחלקת היפותזות סופית בגודל N. בספרות בדר"כ ההיפותזה נקראת expert - זאת מכיוון ובכל איטרציה t, הלומד מקבל את עצתם של "מומחים", ומשתמש בעצה זו על מנת להחליט על הערך הבא. באופן פורמלי:

הגדרת המודל משחק חוזר בין הלומד $\mathcal{A}$לסביבה $\mathcal{E}$, בו נתון מספר סיבובים T הידוע לסביבה וללומד. בכל איטרציה t:

- הסביבה בוחרת דוגמא $x_t \in X$ ומגלה אותה ללומד.
- הלומד (האלגוריתם $\mathcal{A}$) מנבא $\hat{y}_t \in Y$.
- הסביבה מחליטה על $y_t \in Y$ ומגלה אותה ללומד.
- ההפסד (העלות) של הלומד מוגדרת ע"י $\hat{l}_t = 1_{[\hat{y}_t \neq y_t]}$.

[26] למעשה, בגלל שביתת הסגל הזוטר, זו היתה ההרצאה האחרונה :(

ניסוח ראשון של המטרה של הלומד למזער את ההפסד המצטבר:

$$\hat{L}_T = \sum_{t=1}^{T} \hat{l}_t$$

אבחנה: ללא הנחות נוספות, הסביבה יכולה לכפות $\hat{L}_T = T$. כדי למנוע זאת, ננסה לנסח בשנית:

ניסוח שני של המטרה: למזער את **החרטה**.

הגדרה 15.1 החרטה מוגדרת להיות:

$$R_T = \hat{L}_T - \min_{h \in \mathcal{H}} L_T(h) \left(= \sum_{t=1}^{T} l_t(h) = \sum 1_{h(x_t) \neq y_t} \right)$$

כאן כבר משווים להחלטות קבועות - וזו נשמעת מטרה יותר ניתנת להשגה, אבל רגע, כל הרעיון היה להיות אדפטיביים, לא? נאמר על זה כמה מילים בסוף השיעור.

הגדרה 15.2 החרטה הממוצעת מוגדרת להיות:

$$\frac{1}{T} R_T$$

וכעת, להגדרת הבעיה עצמה:

הגדרה 15.3 בעיית online classification תקרא **למידה** אם קיים אלג' $\mathcal{A}$ כך שלכל סביבה $\mathcal{E}$, מתקיים:

$$\lim_{T \to \infty} \frac{1}{T} R_T = 0$$

במושג זה מובע הרעיון שאנו (הלומד) רוצים שהביצועים שלנו יהיו טובים כמו היפותזה קבועה במבט לאחור.

15.1.1 במקרה ה־realizable

- הסביבה חייבת לייצר דוגמאות כך שיש לפחות $h \ni \mathcal{H}$ עם $L_T(h) = 0$ (כלומר החרטה היא רק האיבר השני).
- לצורך הפשטות, $\mathcal{H} = \{h_1, ..., h_N\}$

נראה שהסביבה מאוד מוגבלת, ושהלומד יכול להצליח.
מאוד טבעי לעשות אדפטציה בשלב זה ל־ERM: להסתכל על ההיסטוריה, ולהחזיר היפוטזה שה־loss שלה הוא אפס. ואכן, קיימת כזו:

אלגוריתם 14 Consistent

נסמן את כל ההיפותזות העקביות בזמן t:

$$\mathcal{H}_t = \{ h \in \mathcal{H} | \, L_{t-1}(h) = 0 \}$$

אזי האלגוריתם מחזיר $\hat{h}_t \in \mathcal{H}_t$, כאשר $\hat{y}_t = \hat{h}_t(x_t)$.

טענה 15.4 האלגוריתם הנ"ל מבצע לכל היותר $N-1$ טעויות, כלומר החרטה שלו היא[27]:

$$R_T \leq N - 1$$

[27] ניתן להוכיח שהחסם הדוק, ע"י שרשרת דוגמאות שתיפול $N-1$ פעמים.

הוכחה: בכל פעם בו האלגוריתם טעה, הגודל של $\mathcal{H}_t$ קטן לפחות ב־1 (קיימת לפחות היפותזה אחת בו - זו זבה בחרנו, שבוחרת את הערך השגוי). כמו כן, מהנחת הפרידות, הסט $\mathcal{H}_t$ לעולם לא נהיה ריק (שכן יש בו לפחות היפותזה אחת נכונה תמיד). מאחר ו־$|\mathcal{H}| = N$, נקבל כי האלגוריתם לא יכול לטעות יותר מ־$N-1$ פעמים. ■

מכאן:

$$\frac{R_T}{T} \to 0$$

אם נתבונן בסט של ההיפותזות העקביות, באלגוריתם מחקנו אחת בכל שלב - אולי אפשר לצמצם יותר? אפשר! ואפילו יש אלגוריתם כזה:

אלגוריתם 15 Halving

נסמן את כל ההיפותזות העקביות בזמן t:

$$\mathcal{H}_t = \{h \in \mathcal{H} | \, L_{t-1}(h) = 0\}$$

אזי:

$$\hat{y}_t = \begin{cases} 1 & \sum_{h \in \mathcal{H}} h(x_t) \geq 0 \\ -1 & otherwise \end{cases}$$

נשים לב כי במקרה זה, בכל פעם שהאלגוריתם טועה, לפחות מחצית מההיפותזות ב־$\mathcal{H}_T$ מוסרות (ישירות מההגדרה), ולכן

$$R_T \leq \lfloor \log N \rfloor$$

15.1.2 במקרה הכללי (אגנוסטי)

במקרה זה, מכיוון ולא קיימת היפותזה נכונה ב־$\mathcal{H}$, כבר לא הגיוני להוריד היפותזות שטועות מ־$\mathcal{H}_t$ (שכן יתכן ונוותר בסוף ללא אפשרויות). מה נעשה אם כך?

אלגוריתם 16 Follow the Leader

לעשות ERM על סיבובים $1, ..., t-1$:

- בכל שלב בוחרים היפותזה הממזערת עד כה:

$$\hat{h}_t = \underset{h \in \mathcal{H}}{argmin} L_{t-1}(h)$$

- על פי ההיפותזה הנ"ל לבחור את התיוג:

$$\hat{y}_t = \hat{h}_t(x_t)$$

עם זאת, הביצועים של אלגוריתם זה לא מלהיבים. בפרט, הוא אינו מבטיח הקטנה של החרטה הממוצעת לאורך זמן. תוצאה שלילית זו גם מתרחשת עבור על אלגוריתם דטרמיניסטי. נראה זאת:

משפט 15.5 יהי $\mathcal{A}$ אלגוריתם דטרמיניסטי. נניח כי: $\mathcal{H} \supseteq \{h_+, h_-\}$. אזי קיימת סביבה שכופה:

$$\frac{R^T}{T} \geq \frac{1}{2} \Leftrightarrow R_T \geq \frac{T}{2}$$

הוכחה: נביט בסביבה הבוחרת:

$$y_t = \begin{cases} 1 & \hat{y}_t = -1 \\ -1 & \hat{y}_t = 1 \end{cases}$$

אזי $\hat{L}_T = T$. מהו ההפסד האופטימלי? נשים לב כי מכיוון ו־$\mathcal{H}$ מכילה את h_+, h_-, אזי בכל איטרציה אחת מהן, ורק אחת מהן, טועה. לכן:

$$minL_T(h) \leq min\{L_T(h_+), L_T(h_-)\} \leq \frac{T}{2}$$
$$\Rightarrow R_T \geq T - \frac{T}{2} \geq \frac{T}{2}$$

■

התוצאה הזו מאוד מאוד שלילית ־ דינו של כל אלגוריתם דטרמיניסטי סופו להכשל.

מסקנה 15.6 רנדומיות תעזור לנו ללמוד.

15.2 הוספת רנדומיות למודל

מה זה בכלל אומר אלגוריתם רנדומי, ואיך זה משפיע על המודל שהגדרנו?

אלגוריתם רנדומי

- האלגוריתם בוחר וקטור הסתברות $P_t = (P_t(1), ..., P_t(N))$, כאשר:

$$\hat{h}_t = h_i \ W.P. \ P_t(i), \ \hat{y}_t = \hat{h}_t(x_t)$$

- כאשר הסביבה בוחרת את y_t , היא יודעת רק את P_t.

נמטרך לשנות כאן את הדרך בה אנחנו מעריכים את ההפסד ־ שכן הוא רנדומי.

הגדרה 15.7 ההפסד מוגדר להיות:

$$\hat{l}_t = \sum_{i=1}^{N} P_t(i) \, l_t(h_i)$$

כלומר תוחלת ההפסד לפי וקטור ההסתברות P_t. כמו כן:

$$\hat{L}_T = \sum \hat{l}_t, \ R_T = \hat{L}_T - L_T, \ L_T = \min_{h \in \mathcal{H}} L_T(h)$$

נרצה להתחיל בלהבין מה המגבלות של המודל הזה.
נציג כעת אלגוריתם נוסף:

Multiplicative Weights $(Hedge, \ Weighted \ Majority)$ 15.2.1

נשים לב שזה דומה ל־AdaBoost, ולא בכדי ־ אפשר לקבל את AdaBoost ממנו, ויש לו הרבה מאוד שימושים. עוד על כך ־ הפניות בסיכום השיעור באתר.

הרעיון: נחזיק וקטור משקולות $w_t = (w_t(1), ..., w_t(N)) \ni \mathbb{R}_+^N$
בתחילה הוקטור מאותחל ל־$(1, ..., 1)$. בהנתן פרמטר $\eta > 0$, ישנן שתי דרכים פופולריות לעדכן את המשקלות, אנו נשתמש בשניה:

$$w_{t+1}(i) = w_t(i) \cdot e^{-\eta l_t(i)}$$
$$w_{t+1}(i) = w_t(i) \cdot (1 - \eta l_t(i))$$

אלגוריתם 17 Multiplicative Weights

קלט: מספר היפותזות N, פרמטר $\eta > 0$.

- אתחל $w_1 = (1, ..., 1) \in \mathbb{R}^n$
- לכל $t = 1, ..., T$:
  - קבל דוגמא x_t
  - עדכן $W_t = \sum_{i=1}^N w_t(i)$
  - $p_t = \frac{w_t}{W_t}$
  - בחר היפותזה i עם הסתברות $p_t(i)$
  - $\hat{y}_t \leftarrow h_i(x_t)$
  - קבל וקטור הפסד:

$$l_t = (l_t(i))_{i=1}^N = \left(1_{[h_i(x_t) \neq y_i]}\right)_{i=1}^N$$

  - עדכן את המשקלות:

$$\forall i \in [N] \;\; w_{t+1}(i) = w_t(i)(1 - \eta l_t(i))$$

משפט 15.8 האלגוריתם מעלה מבטיח:

$$R_T \leq \frac{\ln(N)}{\eta} + \eta T$$

ואם נניח $T \geq 4\ln(N)$, ונקבע:

$$\eta = \sqrt{\frac{\ln(N)}{T}} \leq \frac{1}{2}$$

יתקבל:

$$R_T \leq 2\sqrt{T \ln(N)} \Rightarrow \frac{R_T}{T} \leq \sqrt{\frac{2\ln(N)}{T}} \overset{t \to \infty}{\longrightarrow} 0$$

הוכחה: רעיון ההוכחה - נסתכל על $W_t = \sum_{i=1}^N w_t(i)$ כ"פוטנציאל", ונקשר אותו ל־$\hat{L_T}$וגם ל־L_T. בחלק השני נראה שאם L_T גדול, הפוטנציאל קטן, ואם הפוטנציאל קטן, L גדול[28].

[28] יתכן ויש כאן בעיה קטנה של סימונים שכן שיניתי את מה שהיה בשיעור כדי שיתאים לסיכום באתר הקורס - אבל במבט שני נראה שגם המרצה התבלבל קצת.

נתחיל בחלק הראשון:

$$W_{t+1}=\sum_{i=1}^{N}w_{t+1}\left(i\right)=\sum_{i=1}^{N}w_t\left(i\right)\cdot\left(1-\eta l_t\left(i\right)\right)=W_t-\eta\sum_{i-1}^{N}w_t\left(i\right)l_t\left(i\right)$$

$$=W_t-\eta W_t\overbrace{\sum P_t\left(i\right)\cdot l_t\left(i\right)}^{l_t}=W_t\left(1-\eta\hat{l}_t\right)\leq W_te^{-\eta\hat{l}_t}$$

$$\Rightarrow W_{t+1}\leq\overbrace{W_1}^{N}\cdot\prod_{i=1}^{T}e^{-\eta\hat{l}_t}=W_1e^{-\eta\hat{L}_T}$$

$$\Rightarrow\hat{L}_T\leq\frac{1}{\eta}\left(\ln N-\ln\left(W_{T+1}\right)\right)$$

נמשיך לחלק השני: נקבע על $h\in\mathcal{H}$. כעת:

$$W_{T+1}\geq w_{T+1}\left(i\right)=\overbrace{w_1}^{1}\left(i\right)\prod_{t=1}^{T}\left(1-\eta l_t\left(i\right)\right)$$

$$\Rightarrow\left(*\right)\ln\left(W_{t+1}\right)\geq\sum_{t=1}^{T}\ln\left(1-\eta l_t\left(i\right)\right)\overbrace{\geq}^{\eta\leq\frac{1}{2}}\sum_{t=1}^{T}\left(-\eta l_t\left(i\right)-\eta^2l_t^2\left(i\right)\right)$$

$$=-\eta\left(L_T\left(i\right)+\eta\overbrace{\sum l_t^2\left(i\right)}^{\leq1}\right)\geq-\eta\left(L_T\left(i\right)+\eta T\right)\left(*\right)$$

$$\Rightarrow\hat{L}\leq\frac{1}{\eta}\left(\ln\left(N\right)-\ln\left(W_{T+1}\right)\right)\overbrace{\leq}^{(*)}\frac{1}{\eta}\left(\ln\left(N\right)-\eta\left(\hat{L}_T\left(i\right)+\eta T\right)\right)$$

$$\Rightarrow\forall i\ L_T-\hat{L}_T\left(i\right)\leq\frac{\ln\left(N\right)}{\eta}+\eta T$$

$$\eta=\sqrt{\frac{\ln\left(N\right)}{T}}\Rightarrow R_T\leq2\sqrt{\ln\left(N\right)T}\Rightarrow\frac{R_T}{T}\leq2\sqrt{\frac{\ln\left(N\right)}{T}}$$

■

ההוכחה הזו חשובה שכן הרעיון הזה מופיע בהרבה אלגוריתמי קירוב.
מהו ההפסד המצטבר הכי גרוע שאפשרי שנקבל? הרנדומיות מונעת מאיתנו לקבל כבר T!

הערה 15.9 אמרנו שגם הסביבה וגם האלגוריתם יודעים מהו T. ובכן, טכנית הם לא יודעים, אבל אנחנו מניחים זאת כי קל להם מאוד לעקוף את הבעיה הזו - יהיו הערות בנושא זה ברשימות השיעור באתר.

15.3 הקשר בין Online ל־Batch

לא נוכיח את הקשר, רק נאמר מהו רעיון ההוכחה.

טענה 15.10 בעית אונליין קשה יותר מ־Batch. במילים אחרות, יש רדוקציה מבעיית Batch לאונליין.

אם $\left(\left(x_1,y_1\right),...,\left(x_T,y_T\right)\right)\sim\mathcal{D}^T$, $T=T\left(\varepsilon\right)$, והרצנו אלגוריתם אונליין שנתן חיזויים לפי $\hat{h}_1,...,\hat{h}_T\in\mathcal{H}$, האם ניתן ליצר היפותזה $\hat{h}$ כך ש:$\mathbb{E}\left[L_{\mathcal{D}}\left(\hat{h}\right)\right]\leq L_{\mathcal{D}}\left(h^*\right)+\varepsilon$? נטען שכן!

טענה 15.11 $\varepsilon\approx\frac{R_T}{T}$.

כלומר, אם יש אלגוריתם אונליין שמבטיח חרטה ממוצעת $\frac{R_T}{T}$, ורוצים דיוק ε. לפי $(*)$, $T\geq\frac{R_T}{\varepsilon}$, $\hat{h}=h_\varepsilon$ בהסתברות $\ldots\frac{1}{T}$

דוגמא: קיבלנו עם MW חרטה ממוצעת:

$$\varepsilon = \frac{\sqrt{\ln(N)}}{\sqrt{T}} \Rightarrow T \geq \frac{\ln N}{\varepsilon^2}$$

כלומר אם הצלחנו במשימת אונליין, אפשר לבחור את אחת ההיפותזות ש"סחבנו" במהלך הדרך באופן מקרי, ולהשתמש בה בשביל משימת Batch.