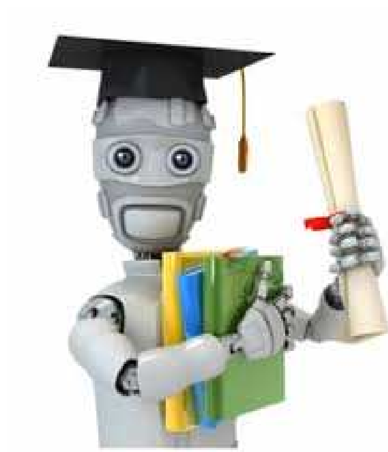
Introduction to Machine Learning (67577)

July 9, 2018



המסמך נכתב ע"י גלעד גרין, על בסיס הרצאות ותרגולי הקורס וספרי הקורס:

Understanding Machine Learning: From Theory to Algorithms

Elements of Statistical Learning

Reinforcement Learning: An Introduction

תוכן עניינים

חלק I

הקדמה

תהליך הלמידה עוסק בהפיכת ניסיון לידע או מומחיות. בהינתן עולם בעיה ($domain$) נרצה אלגוריתם ($learner$) אשר ילמד סט מידע ($trainset$) לספק פרדיקציה טובה עבור פריטים חדשים באותו עולם תוכן ($testset$). כאשר ניגש לבעיית למידה נרצה ללמוד את ה־$data$. נרצה להבין איך ה־$domain$ מתנהג, מהן התכונות ($features$) שנרצה ללמוד בעזרתן. מתוך כך נרצה להעריך איזה מודל מתאים ל־$domain$ ($model\, selection$) ולבסוף להעריך את טיב המודל ($model - assesment$) על מידע שכלל עוד לא ראינו ($validation\, set$).
במהלך הקורס נפגוש בכלים מתמטיים רבים. כאשר מדברים על כיצד המידע מתנהג למעשה מדובר בשאלה הסתברותית. כיצד המידע מתפלג על גבי ה־$features$ השונים והתיוגים ($labels$) השונים. כאשר נרצה ללמוד $domain$ מסויים נרצה מסווג שיטעה ($loss$) כמה שפחות והרי שמכאן כי זאת בעיית מינימיזציה אשר בהתאם למודלים השונים נפגוש חשבון אינפיטיסימלי ואלגברה לינארית.

מטרת הקורס ללמד איך לא להיות "$machine\, learning\, monkey$". כלומר נרצה להבין את המודלים השונים ולהחליט איזה מודל מתאים לנו במקרה הספציפי של ה־$domain$ עליו אנו עובדים. "להבין" בקורס זה משלב את ההבנה הפורמלית של משפטים וטענות, ואת פיתוח האינטואיציה עבור התהליכים השונים.

על אף שסיכום זה מבוסס על החומר שהועבר בהרצאות ובתרגולים, סדר הנושאים מנסה להתאים בין הסדר בספר הלימוד לבין סדר ההרצאות.

חלק II

ידע מוקדם

פרק זה עוסק ביסודות המתמטיים הנחוצים לקורס ומכיל את ההגדרות שהועברו בתרגולים בנושאי אלגברה לינארית, חשבון אינפיטיסימלי במרחבים ממימד גבוהה והסתברות.

1 אלגברה לינארית

1.1 נורמה ומכפלה פנימית סטנדרטית

הגדרה 1.1 נורמה הינה פונקציה $||\cdot|| \,:\, \mathbb{R}^m \to \mathbb{R}$ המקיימות את התכונות הבאות לכל $a \in \mathbb{R},\ u, v \in \mathbb{R}^m$:

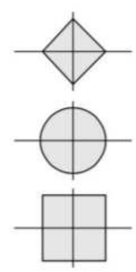
1. חיוביות - $||v|| \geq 0$ ואם $||v|| = 0$ אז $v = 0$.
2. הומוגניות חיובית - $||av|| = |a| \cdot ||v||$
3. אי שיוויון המשולש - $||u + v|| \leq ||u|| + ||v||$

הגדרה 1.2 כדור היחידה הינו $B = \{x \,|\, ||x|| \leq 1\}$. כאשר בהתאם לסוג הנורמה התוצאה שמתקבלת היא שונה:

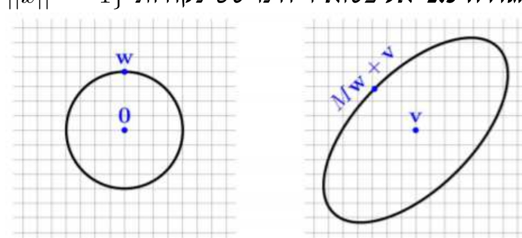
$$\|x\|_1 = \sum_{i=1}^{m} |x_i|,$$

$$\|x\|_2 = \left(\sum_{i=1}^{m} |x_i|^2\right)^{1/2} = \sqrt{x^* x},$$

$$\|x\|_\infty = \max_{1 \leq i \leq m} |x_i|,$$



הגדרה 1.3 אליפסואיד הינו סט נקודות $A = \{y \,|\, y = Mx + v \ \wedge\ ||x||^2 = 1\}$ כאשר M מטריצה הפיכה.



טענה 1.4 תהי U מטריצה אורתוגונלית אזי האליפסואיד המוגדר על ידי A שווה לאליפסואיד המוגדל על ידי AU.

הגדרה 1.5 מכפלה פנימית הינה פונקציה $\langle \cdot, \cdot \rangle : \mathbb{R}^m, \mathbb{R}^m \to \mathbb{R}$ המקיימת את התכונות הבאות: לכל $a \in \mathbb{R},\ x, y, z \in \mathbb{R}^m$

1. סימטריות - $\langle x,y\rangle = \langle y,x\rangle$

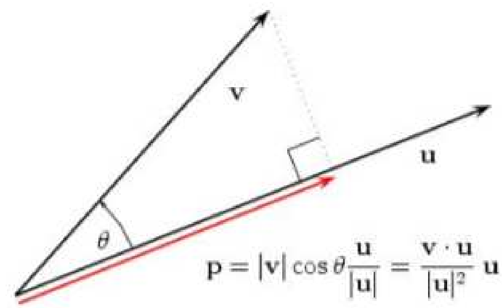
2. לינאריות - $\langle ax+z,y\rangle = a\langle x,y\rangle + \langle z,y\rangle$

3. חיוביות: $\langle x,x\rangle \geq 0$ ומתקיים $\langle x,x\rangle = 0 \Leftrightarrow x=0$

הגדרה 1.6 המכפלה הפנימית הסטנדרטית הינה: $\langle x,y\rangle = x^T \cdot y = \sum_{i=1}^{m} x_i \cdot y_i$.

טענה 1.7 עבור זוג וקטורים x,y מתקיים כי המכפלה הסקלרית שלהם היא $\langle x,y\rangle = ||x|| \cdot ||y|| \cdot cos\theta$ כאשר θ הזווית שבין הוקטורים.

הגדרה 1.8 ההטלה של וקטור v על וקטור u הינה הוקטור p עם אורך $||v|| \cos\theta$ בכיוון הוקטור u:



יהי $w \in \mathbb{R}^d$ אזי ה־$halfspace$ הינו הקבוצה $\{x | \langle w,x\rangle \geq b\}$ וה־$hyperplane$ (על מישור) הינו הקבוצה $\{x | \langle w,x\rangle = b\}$.

1.2 פירוקי EVD, SVD

משפט 1.9 <u>פירוק EVD</u> - תהי $B \in \mathbb{R}^{m\times m}$ סימטרית אזי קיימת U אורתוגונלית ו־λ אלכסונית כך ש־ $B = U\lambda U^T$ ולכל $u_i \in U$ וקטור עצמי של B עם ערך עצמי $\lambda_{i,i}$.

משפט 1.10 <u>פירוק SVD</u> - תהי $A \in \mathbb{R}^{d\times m}$ אזי קיימת $U \in \mathbb{R}^{d\times d}$,$V \in \mathbb{R}^{m\times m}$ אורתוגונליות ו־ $\Sigma \in \mathbb{R}^{d\times m}$ אלכסונית כך ש־ $A = U\Sigma V^T$. בנוסף מתקיים שאם $rank(A) = r$ אזי לכל $i \leq r$ הוקטורים v_i, u_i הינם הוקטורים הסינגולריים של A עם ערך סינגולרי $\sigma_i \in \Sigma_{i,i}$ ולכל $i > r$ אז $Av_i = 0,\ A^T u_i$.

טענה 1.11 <u>תכונות של SVD</u>:

1. σ_i^2הם ערכים סינגולריים של המטריצה XX^T.

2. u_i הם וקטורים סינגולריים ימניים המתאימים של XX^T.

3. v_i הם וקטורים סינגולריים שמאליים המתאימים של $X^T X$.

1.3 שימוש SVD בדחיסה

נניח כי $X \in \mathbb{R}^{d\times m}$ ורוצים לשערך אותה על ידי מטריצה מדרגה $k << d,m$. נרצה למצוא $\tilde{X} \in \mathbb{R}^{d\times m}$ שדרגתה k הממזערת את:

$$\tilde{r} = rank\left(X - \tilde{X}\right),\ ||X - \tilde{X}||_F = \sum_{i,j}\left(X_{i,j} - \tilde{X_{i,j}}\right)^2 = \sum_{i=1}^{\tilde{r}}\sigma_i\left(X - \tilde{X}\right)$$

כבעיית אופטימיזציה נוכל לכתוב בתור $||X - \tilde{X}||_F \underset{\tilde{X}:rank\left(\tilde{X}\right)\leq k}{argmin}$.

משפט 1.12 תהי $X \in \mathbb{R}^{d\times m}$ עם SVD $X = U\Sigma V^T$. נגדיר $\tilde{\Sigma} \in \mathbb{R}^{d\times m}$ אלכסונית בתור

$$\tilde{\Sigma}_{i,i} = \begin{cases}\Sigma_{i,i} & i < k\\ 0 & i \geq k\end{cases}$$

אזי הפתרון לבעיית הדחיסה הוא $\tilde{X} = U\tilde{\Sigma}V^T$.

2 חשבון רב מימדי

הגדרה 2.1 הנגזרת החלקית של הפונקציה $f:\mathbb{R}^n \to \mathbb{R}$ בנקודה $x \in \mathbb{R}^n$ ביחס ל־x_i היא:

$$\frac{\partial f\left(x\right))}{\partial x_i} = \lim_{a\to 0}\frac{f\left(x + a\cdot e_i\right) - f\left(x\right)}{a}$$

הגדרה 2.2 הגרדיאנט של פונקציה $f:\mathbb{R}^n \to \mathbb{R}$ בנקודה $x \in \mathbb{R}^n$ הוא:

$$\nabla f\left(x\right) = \begin{bmatrix}\frac{\partial f(x)}{\partial x_1}\\ \vdots\\ \frac{\partial f(x)}{\partial x_n}\end{bmatrix}$$

הערה 2.3 הגרדיאנט מספק תנאי הכרחי לנקודות קיצון של פונקציה. כלומר אם x נקודת קיצון והגרדיאנט בנקודה קיים אזי $\nabla f\left(x\right) = 0$.

הגדרה 2.4 תהי $f:\mathbb{R}^n \to \mathbb{R}$ ו־ $x_0 \in \mathbb{R}^n$ אזי הקירוב הלינארי של f סביב x_0 הוא $f\left(x_0\right) + \nabla f\left(x_0\right)\cdot\left(x - x_0\right)$. כלומר לכל $x \in \mathbb{R}^n$ מתקיים כי: $f\left(x\right) \approx f\left(x_0\right) + \nabla f\left(x_0\right)\cdot\left(x - x_0\right)$.

הגדרה 2.5 קו הגובה של $f:\mathbb{R}^n\to\mathbb{R}$ ביחס ל־$c\in\mathbb{R}$ הוא הקבוצה $\{x|f(x)=c\}$.

הגדרה 2.6 עבור פונקציה $f:\mathbb{R}^d\to\mathbb{R}^m$ היעקוביאן של הפונקציה הינה מטריצה $x\times d$ של כל הנגזרות החלקיות:

$$J_x(f)=\begin{bmatrix}\frac{\partial f_1(x)}{\partial x_1} & \cdots & \frac{\partial f_1(x)}{\partial x_d}\\ \vdots & & \vdots\\ \frac{\partial f_m(x)}{\partial x_1} & \cdots & \frac{\partial f_m(x)}{\partial x_d}\end{bmatrix}$$

<u>**דוגמה:**</u> נגדיר כדור ϵ סביב נקודה x_0: $B_\epsilon(x_0)=\{x_0+x\,|\,||x||_2=\epsilon\}$. על כן מהו הוקטור שממקסם/ממזער את הקירוב הלינארי של f סביב סביב x_0 על פני הכדור?

<u>**פתרון:**</u> נראה כי מתקיים:

$$f(x+x_0)\approx f(x_0)+\nabla f(x_0)^T x=\underbrace{||\nabla f(x_0)||_2}_{doesn't\,depend\,on\,x}\cdot\underbrace{||x||_2}_{\epsilon}\cdot cos\theta$$

ולכן מזעור/מקסום הקירוב הלינארי שקול למיקסום/מזעור $cos\theta$ כאשר זוהי הזווית בין x לגרדיאנט של f בנקודה x_0.
על כן נקבל כי עבור:

- $\theta=0^o$ נקבל את הוקטור הממקסם $x=\epsilon\frac{\nabla f(x_0)}{||\nabla f(x_0)||_2}$
- $\theta=180^o$ נקבל את הוקטור הממזער $x=-\epsilon\frac{\nabla f(x_0)}{||\nabla f(x_0)||_2}$

3 הסתברות

3.1 אי שוויונות

משפט 3.1 אי שיוויון מרקוב ־ יהי X משתנה מקרי (מ"מ) אי שלילי בעל תוחלת סופית אזי לכל $0<a\in\mathbb{R}$ מתקיים כי:

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

משפט 3.2 אי שיוויון צ'בישב ־ יהי X מ"מ בעל תוחלת ושונות סופיות אזי לכל $0<a\in\mathbb{R}$ מתקיים כי:

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

משפט 3.3 אי שיוויון הופדינג - יהיו $X_1, ..., X_m$ מ"מ בלתי תלויים (ב"ת) וחסומים כך שלכל $i \in [m]$ מתקיים $a_i \leq X_i \leq b_i$.
יהי $\bar{X} = \frac{1}{m}\sum_{i=1}^{m} X_i$ אזי

$$\mathbb{P}\left[\left|\bar{X} - \mathbb{E}\left[\bar{X}\right]\right| \geq \epsilon\right] \leq 2exp\left(\frac{-2m^2\epsilon^2}{\sum_{i=1}^{m}\left(b_i - a_i\right)^2}\right)$$

3.2 הסתברות רב מימדית

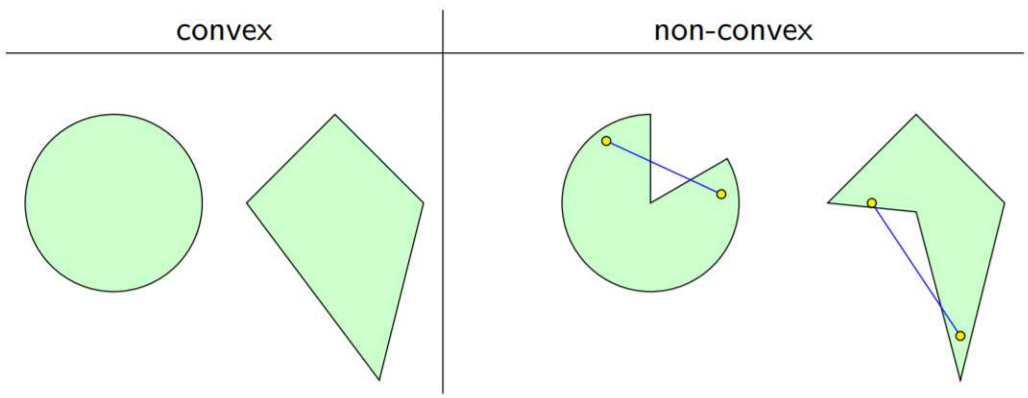
להשלים

4 $Convexity,\, Lipschitzness\, and\, Smoothness$

4.1 $Convexity$

4.1.1 קבוצות קמורות

הגדרה 4.1 יהי V מרחב וקטור. $C \subseteq B$ נקראת קמורה אם לכל $u, v \in C$ ו־$\alpha \in [0,1]$ מתקיים $\alpha u + (1-\alpha) v \in C$



הגדרה 4.2 עבור קבוצת נקודות $\{v_i \,|\, i \in I\}$ הנקודה $\sum_{i \in I} \alpha_i v_i$, כאשר המקדמים $\{\alpha_i > 0 \,|\, i \in I\}$ חיוביים ונסכמים ל־1, נקראת צירוף קמור של ה־v_i.

דוגמה: כדורי יחידה הינם קבוצה קמורה.

הוכחה: יהי $B=\{v\in V\,|\,||v||=1\}$. יהי $u,v\in B$ ו־$\alpha\in[0,1]$ אזי:

$$\underbrace{\begin{aligned}||\alpha u+(1-\alpha)v|| &\leq ||\alpha u||+||(1-\alpha)v||\\ &= \alpha||u||+(1-\alpha)||v|| \leq \alpha+1-\alpha=1\end{aligned}}$$

$$\Downarrow$$

$$\alpha u+(1-\alpha)v\in B$$

משפט 4.3 כל הקבוצות הבאות הן קמורות:

1. חיתוך קבוצות קמורות הוא קמור. $C=\bigcap_{i\in I}C_i$ עבור $\{C_i\,|\,i\in I\}$ אוסף קבוצות קמורות.
2. סכום וקטורי $C_1+C_2=\{c_1+c_2\,|\,c_1\in C_1,\,c_2\in C_2\}$ עבור C_1,C_2 קמורות.
3. הקבוצה $\lambda C=\{\lambda c\,|\,c\in C\}$ עבור C קמורה.

הוכחה: (מקרה 2) יהיו $u,v\in C_1+C_2$. נראה שלכל $\alpha\in[0,1]$ מתקיים $\alpha u+(1-\alpha)v\in C_1+C_2$. נראה כי מתקיים $u=a_1+a_2,\,v=b_1+b_2$ כך ש־$a_1,b_1\in C_1$ ו־$a_2,b_2\in C_2$. כעת:

$$\underbrace{\begin{aligned}\alpha u+(1-\alpha)v &= \alpha(a_1+a_2)+(1-\alpha)(b_1+b_1)\\ &= \underbrace{\alpha a_1+(1-\alpha)b_1}_{\in C_1}+\underbrace{\alpha a_2+(1-\alpha)b_2}_{\in C_2}\end{aligned}}$$

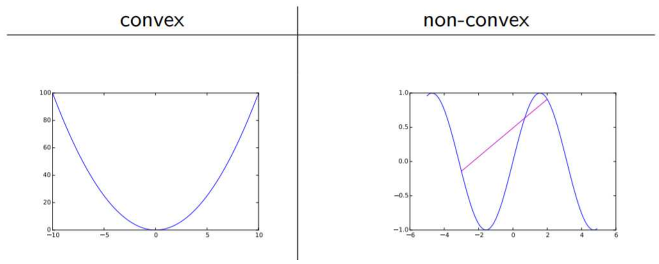
$$\Downarrow$$

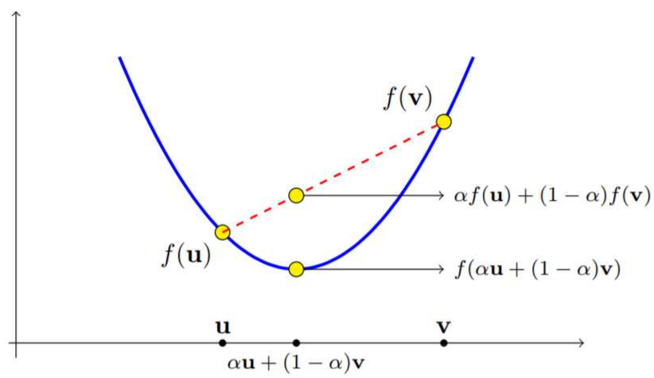
$$\alpha u+(1-\alpha)v\in C_1+C_2$$

4.1.2 פונקציות קמורות

הגדרה 4.4 יהי V מרחב וקטור ו־C קבוצה קמורה. פונקציה $f:C\to\mathbb{R}$ תקרא קמורה אם לכל $u.v\in C$ ו־$\lambda\in[0,1]$ מתקיים:

$$f(\lambda u+(1-\lambda)v)\leq\lambda f(u)+(1-\lambda)f(v)$$





הגדרה 4.5 ה־$epigraph$ של פונקציה היא הקבוצה $epigraph\left(f\right)=\left\{ \left(x,\beta\right)\mid f\left(x\right)\leq\beta\right\}$

משפט 4.6 פונקציה f הינה קמורה $\Longleftrightarrow$ $ephgraph\left(f\right)$ קבוצה קמורה

דוגמה: נורמה $||\cdot||$ היא פונקציה קמורה. לכל $u,v\in\mathbb{R}^{d}$ ול־$\lambda\in\left[0,1\right]$ מתקיים:

$$||\lambda u+\left(1-\alpha\right)\lambda||\overset{tri.\,inequality}{\leq}||\lambda u||+||\left(1-\lambda\right)v||=\lambda\left||u\right||+\left(1-\lambda\right)\left||v\right||$$

דוגמה: ההעתקה האפינית $f\left(u\right)=\left\langle w,u\right\rangle +b$ הינה פונקציה קמורה.

משפט 4.7 הפונקציה $g\left(u\right)$ הינה פונקציה קמורה בכל המקרים הבאים:

1. סכום ממושקל חיובית של פונקציות קמורות: $g\left(u\right)=\sum_{i\in I}\gamma_{i}f_{i}\left(u\right)$ עבור $\left\{ \gamma_{i}\mid i\in I\right\}$ חיוביות ו־$\left\{ f_{i}\mid i\in I\right\}$ קמורות.
2. הרכבה של פונקציה קמורה על אפינית: $g\left(u\right)=f\left(Au+b\right)$ עבור f קמורה ו־$A\in\mathbb{R}^{m\times n},\ b\in\mathbb{R}^{m}$
3. sup או max של אוסף פונקציות קמורות: $g\left(u\right)=\underset{i\in I}{sup}\left\{ f_{i}\left(u\right)\right\}$ עובר $\left\{ f_{i}\mid i\in I\right\}$ קמורות.

הוכחה: (מקרה 3) נשים לב כי $\forall i \in I\ \ (u,t) \in epigraph\,(f_i) \iff \forall i \in I\ f_i\,(u) \le t \iff \underset{i \in I}{sup} f_i\,(u) \le t \iff (u,t) \in epigraph\,(g)$. ומכאן כי $epigraph\,(g) = \bigcap_{i\in I} epigraph\,(f_i)$. היות ו־$f_i$ קמורות אזי חיתוך קמורות הוא קמור ולכן גם g קמורה.

4.2 ליפשיציות

הגדרה 4.8 תהי $C \subset \mathbb{R}^d$. פונקציה $f : \mathbb{R}^d \to \mathbb{R}^k$ הינה ρ־ליפשיצית מעל C אם $\forall w_1, w_2 \in C$ מתקיים $||f\,(w_1) - f\,(w_2)|| \le \rho\,||w_1 - w_2||$

באופן אינטואיטיבי פונקציה ליפשיצית מוגבלת במהירות ההשתנות שלה

דוגמה: פונקציה לינארית $f : \mathbb{R}^d \to \mathbb{R}$ המוגדרת על ידי $f\,(w) = \langle v, w\rangle + b$ עבור $v \in \mathbb{R}^d$ הינה $||v||$־ליפשיצית. בעזרת אי שיוויון קושי־שוורץ נקבל:

$$|f\,(w_1) - f\,(w_2)| = |\langle v, w_1 - w_2\rangle| \le ||v|| \cdot ||w_1 - w_2||$$

טענה 4.9 תהי $f\,(x) = g_1\,(g_2\,(x))$ כאשר g_1 הינה ρ_1־לפשיצית ו־g_2 הינה הינה ρ_2־לפשיצית אזי f הינה $\rho_1\rho_2$־לפשיצית

4.3 חלקות

הגדרה 4.10 פונקציה דיפרנציאבילית $f : \mathbb{R}^d \to \mathbb{R}$ הינה β־חלקה אם הגרדיאנט שלה הינו β־ליפשיצי.
כלומר $\forall v, u\ \ ||\nabla f\,(v) - \nabla f\,(u)|| \le \beta\,||v - w||$

טענה 4.11 תהי $f\,(w) = g\,(\langle w, x\rangle + b)$ כאשר g פונקציה β־חלקה ו־$x \in \mathbb{R}^d$ אזי f הינה $\beta\,||x||^2$־חלקה. (הוכחה עמוד 163 בספר הלימוד)

טענה 4.12 תהי פונקציה קמורה ו־β־חלקה אזי $||\nabla f\,(w)||^2 \le 2\beta f\,(w)$. פונקציה המקיימת תכונה זו נקראת גם פונקציה $self - bounded$.

חלק III

תיאוריה כללית

5 יסודות בלמידה

5.1 סוגי למידה

- $\underline{Batch\, vs.\, Online}$ - בסוג למידה $batch$ נקבל $training\, set$ מתוייג ו- $test\, set$ שאיננו מתוייג. נלמד על סט האימון ונספק $estimation$ עבור סט הבדיקה. בלמידה $online$ בכל פעם נקבל $sample$, נעדכן את המודל בהתאם ל-$sample$ החדש ונחזיר פרדיקציה.
- $\underline{Supervised\, vs.\, Unsupervides}$ - כאשר ה-$trainset$ מתוייג בתיוגי "אמת" נכנה את הלמידה $supervised$. עבור $trainset$ שאיננו מתוייג נכנה את הלמידה $unsupervised$.

5.2 המודל של הלמידה הסטטיסטית

הקלט שניתן ללומד:

1. $Domain\, set$ הינו סט ארביטררי $\mathcal{X}$ של מידע המכיל דוגמיות המכונות $samples$. כל $x\in\mathcal{X}$ מייצג $feature\, vector$.
2. $Label\, Set$ הינו סט התיוגים האפשריים $\mathcal{Y}$ עבור המידע ב-$\mathcal{X}$.
3. $Training\, Set$ הינו אוסף $S=\{(x_i,y_i)\}_{i=1}^{m}\subseteq\mathcal{X}\times\mathcal{Y}$ כאשר $\forall i\in[m]\ x_i\in\mathcal{X},\ y_i\in\mathcal{Y}$.

בתהליך הלמידה:

1. נרצה שהלומד יתאמן על המידה ויחזיר $Prediction\, Rule$ - פונקציה מסט המידע לסט התיוגים. $h:\mathcal{X}\to\mathcal{Y}$. נסמן את פלט האלגוריתם $\mathcal{A}(S)$
2. בהינתן דומיין $\mathcal{X}$ וסט תיוגים $\mathcal{Y}$ מחלקת היפותזות $\mathcal{H}$ הינה קבוצת כל הפונקציות $\mathcal{H}=\{h:\mathcal{X}\to\mathcal{Y}\}$.

כאשר אנו דנים ב־$domain$ מסויים אנו מניחים כי הדגימות בו נוצרו מעל התפלגות כלשהי של המידע, נסמן התפלגות זו ב־$\mathcal{D}$ ותויג נכונה על ידי פונקציה כלשהי f. עבור חוק הפרדיקציה שהחזיר הלומד נרצה למדוד את הטעות של חוק הפרדיקציה h. נגדיר את הטעות של מסווג, ה־$true\,error$ או ה־$generalization\,error$ על ידי:

$$L_{\mathcal{D},f}\left(h\right)\overset{def.}{=}\mathbb{P}_{x\sim\mathcal{D}}[h\left(x\right)\neq f\left(x\right)]\overset{def.}{=}\mathcal{D}\left(\left\{x:h\left(x\right)\neq f\left(x\right)\right\}\right)$$

בהמשך נראה כי הגדרה זו משתנה מעט. היות והלומד איננו יודע את $\mathcal{D},f$ נוכל רק לבדוק את ה־$training\,error$. נקרא גם $empirical\,error$ או $empirical\,risk$:

$$L_{S}\left(h\right)\overset{def.}{=}\frac{\left|i\in\left[m\right]\,:\,h\left(x_{i}\right)\neq y_{i}\right|}{m}$$

נשים לב שבהגדרה זו אנו מניחים שמשקל כל טעות שווה ־ התפלגות אחידה.

5.3 $ERM,\,Realizibility\,and\,i.i.d$

כאשר תיארנו את הרעיון המנחה בלמידה רצינו כי הלומד ידע לספק פרדיקציה טובה עבור המידע. ההגדרה של "טובה" איננה חד משמעית אולם לרוב מכילה גם אינפורמציה עבור ה"איבוד" של חוק הפרדיקציה שנחזיר. כלומר החזרת אמת־מידה לטעויות הפרדיקציה שביצע האלגוריתם.

5.3.1 $Loss\,functions$

קיימות פונקציות איבוד רבות בעלות תכונות שונות והמשמשות אלגוריתמים שונים:

- $\underline{0-1\,Loss}$ ־ פונקציית איבוד אשר נותנת 1 עבור טעות ו־0 להצלחה. בהינתן סט מתוייג $S=\left\{(x_{i},y_{i})\right\}_{i=1}^{m}$ נגדיר:

$$L_{S}^{0-1}\left(h\right)=\sum_{i=1}^{m}\mathbb{1}_{h\left(x_{i}\right)\neq y_{i}}$$

- $\underline{Absolute\,Value\,Loss}$ ־ פונקציית איבוד הסוכמת את מרחקי הפרדיקציות מערכי האמת:

$$L_{S,f}=\sum_{\left(x,y\right)\in S}\left|y-f\left(x\right)\right|$$

- $\underline{Least\,Squares\,Loss}$ ־ פונקציית איבוד הסוכמת את ריבועי הפרשי המרחקים בין הפרדיקציות לערכי האמת:

$$L_{S,f}=\sum_{\left(x,y\right)\in S}\left(y-f\left(x\right)\right)^{2}$$

5.3.2 עיקרון ERM

אלגוריתם למידה המממש את עיקרון ה$ERM - Empirical\,Risk\,Minimization$ מחזיר היפותזה הממזערת את שגיאת האימון. עבור $f:\mathcal{X}\to\mathcal{Y}$ פונקציית תיוג המתייגת נכונה את $\mathcal{X}$:

$$ERM_{\mathcal{H}}\left(S\right)\in\underset{h\in\mathcal{H}}{argmin}L_{S}\left(h\right)=\underset{h\in\mathcal{H}}{argmin}\frac{1}{m}\left|\left\{ x_{i}\in S\,|\,h\left(x_{i}\right)\neq f\left(x_{i}\right)\right\} \right|$$

5.3.3 $The\,realizability\,assumption$

הגדרה 5.1 עבור מחלקת היפותזות $\mathcal{H}$ הנחת ה־$realizability$ אומרת כי קיימת $h^{*}\in\mathcal{H}$ כך שמתקיים כי $L_{\mathcal{D},f}\left(h^{*}\right)=0$. מכאן גם כי לכל $S\subseteq\mathcal{X}$ שנדגם בהתאם ל־$\mathcal{D}$ מתקיים $L_{S}\left(h^{*}\right)=0$.

5.3.4 $Independently\,and\,Identically\,Distributed-i.i.d$

כאשר נבנה סט אימון מתוך $\mathcal{X}$ אשר בעל פילוג $\mathcal{D}$ נבחר את ה־$samples$ באופן מקרי ונניח כי הם בלתי תלויים אחד בשני. נסמן זאת בתור $S\sim\mathcal{D}^{m}$ אשר אומר למעשה כי $x_{1},...,x_{m}$ הם מ"מ ב"ת וש"ה מתוך הפילוג $\mathcal{D}$.

5.4 $No\,Free\,Lunch$

משפט 5.2 יהי אלגוריתם למידה $\mathcal{A}$ המתייג בינארית וביחס לפונקציית האיבוד $0-1$ מעל דומיין $\mathcal{X}$. יהי $m<\frac{\left|\mathcal{X}\right|}{2}$ מספר ה־$samples$ בסט האימון S. אזי קיימת התפלגות $\mathcal{D}$ מעל $\mathcal{X}\times\left\{ 0,1\right\}$ כך ש:

1. קיימת $f:\mathcal{X}\to\left\{ 0,1\right\}$ עבורה $L_{\mathcal{D}}\left(f\right)=0$.
2. בהסתברות לפחות $\frac{1}{7}$ מעל בחירה של $S\sim\mathcal{D}^{m}$ נקבל כי $L_{\mathcal{D}}\left(\mathcal{A}\left(S\right)\right)\geq\frac{1}{8}$.

כלומר, לכל $learner$ תהיה משימת למידה עליה הוא יכשל.

הערה 5.3 עבור אלגוריתם המתייג אקראי נקבל כי האיבוד שווה ל$\frac{1}{2}$. מכאן כי המשפט אומר כי לא נוכל לעשות טוב יותר מניחוש אקראי.

5.5 $Classification\,Errors$

5.5.1 $Accuracy$

הדרך הפשוטה ביותר לבדוק את טיב הסיווג הוא על ידי מדד זה. עבורו פשוט נסכום את סך הטעויות בסיווג שביצע המסווג.

$$\sum_{i=1}^{m}\mathbb{1}_{y_{i}\neq\hat{f}\left(x_{i}\right)}$$

5.5.2 סוגי שגיאות

כאשר אנו מתעסקים עם בעיות $classification$ מתקבלות שתי סוגי שגיאות: אמרנו a כאשר למעשה היה b או אמרנו b כאשר למעשה היה a. לשתי טעויות אלה אופי שונה בהתאם לסוג הבעיה שאנו מנסים לפתור ועלינו להחליט איזה סיווג הינו בעל אפקט ואיזה איננו בעל אפקט. נניח כי אנו מסווגים $data$ מסויים ל"מעניין" ו"לא מעניין" ונניח כי איננו יכולים להסכים לטעויות רבות שבו אמרו עם דוגמאות שהן "לא מעניינות" כאשר הן למעשה כן. על כן נסמן תיוג $y=-1$ בתור "מעניין" ו־$y=1$ בתור "לא מעניין" אזי עבור $\hat{y}$ תוצאת הסיווג על ידי המסווג נקבל:

	$y_i=-1$	$y_i=1$
$\hat{y}_i=-1$	·	Type-II error
$\hat{y}_i=1$	Type-I error	·

על כן

- $Type-I\ error$ - אמרנו "יש אפקט" כאשר "אין אפקט". טעות זו הינה פחות חשובה. בדוגמה למעשה אמרו כי הדוגמה "מעניינת" כאשר למעשה היא הייתה "לא מעניינת" והגדרנו כי נרצה כמה שפחות לטעות על המקרים שבהם טענו כי הדוגמה "לא מעניינת"
- $Type-II\ error$ - אמרנו "אין אפקט" כאשר למעשה "יש אפקט". טעות זו יותר משמעותית. בדוגמה למעשה אמרנו כי דוגמה "לא מעניינת" כאשר היא הייתה "מעניינת" - סוג הטעות שרצינו להפחית.

כעת כשנסתכל שוב על הטבלה נוכל לומר כי:

	$y_i=-1$	$y_i=1$
$\hat{y}_i=-1$	true negatie	false negative
$\hat{y}_i=1$	false positive	true positive

5.5.3 $Precision/Recall$

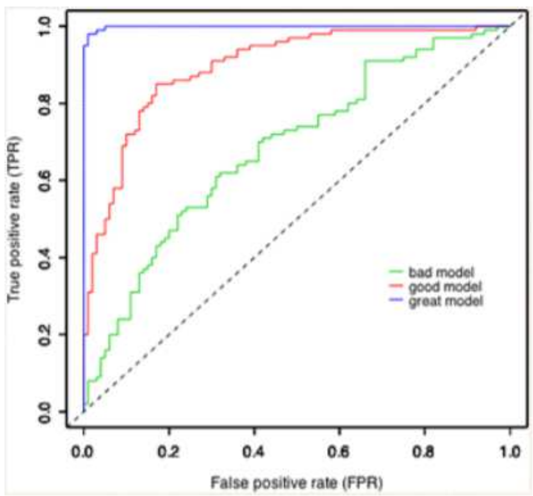
נסמן P את מספר הדוגמאות ה־$positive$ ו־N מספר הדוגמאות ה־$negative$ אזי:

- $Error\ Rate$ - $(FP+FN)\,/\,(P+N)$
- $Accuracy$ - $(TP+TN)\,/\,(P+N)$
- $Precision$ - $TP/\,(TP+FP)$
- $Recall\ \ (True\,positive\,rate)$ - TP/P

- TN/N - $Specificity$
- FP/N - $False\, positive\, rate$

5.6 עקומת ROC

עקומה זו משמשת מניתוח ה־$tradeoff$ בין ה־$sensitivity$ (ה־$true\, positive\, rate,\, recall$) של המודל לבין ה־$specificity$ של המודל.



6 $PAC\ and\ Agnostic\ PAC$

בלמידה הסטטיסטית עבור $\mathcal{X}\ domain$, קבוצת תגיות $\mathcal{Y}$ ומחלקת היפותזות $\mathcal{H}$ ישנו פילוג $\mathcal{D}$ על פני $\mathcal{X}$ אשר איננו ידוע ל־$learner$. ממשפט ה־$No\,Free\,Lunch$ לא נוכל לצפות לקבל כפלט מהאלגוריתם $\mathcal{A}\left(S\right)$ פרדיקציה אשר ללא איבוד על הפילוג הלא ידוע $\mathcal{D}$. על כן נרצה להיות מסוגלים להחזיר פרדיקציה שמצליחה "טוב מספיק" בהסתברות "גבוהה מספיק". כעת נבטא באופן פורמלי רצון זה.

6.1 $PAC\ Model$

הגדרה 6.1 מחלקת היפותזות $\mathcal{H}$ הינה $PAC\ learnable$ אם קיים אלגוריתם $\mathcal{A}$ ופונקציה $m_{\mathcal{H}}:\left(0,1\right)^{2}\rightarrow\mathbb{N}$ עם התכונות הבאות: לכל $\epsilon,\delta\in\left(0,1\right)$, לכל פילוג $\mathcal{D}$ מעל $\mathcal{X}$ ולכל $f\in\mathcal{H}$, אם מתקיימת הנחת ה־$realizability$ ביחס ל־ $\mathcal{H},\mathcal{D},f$ כאשר נריץ את $\mathcal{A}$ על $m\geq m_{\mathcal{H}}\left(\epsilon,\delta\right)$ דוגמאות $i.i.d$ מפילוג $\mathcal{D}$ שתוייגו על ידי f אזי האלגוריתם החזיר היפותזה $h\in\mathcal{H}$ אשר:

$$\mathbb{P}_{S|_{x}\sim\mathcal{D}^{m}}\left[L_{\mathcal{D},f}\left(h\right)\leq\epsilon\right]\geq1-\delta$$

או במילים אחרות ההסתברות שהאלגוריתם יחזיר היפותזה בעלת איבוד על כל $\mathcal{D}$ שגבוה מהסף שקבענו (ϵ) קטן מ־δ. ϵ מכונה ה־$accuracy$ ו־δ מכונה ה־$confidence$.

6.2 $Sample\ Complexity$

הפונקציה $m_{\mathcal{H}}$ מכונה ה־$sample\ complexity$ של מחלקת ההיפותזות והיא מגלמת את הקשר בין ה־$accuracy$ וה־$confidence$ שברצוננו להשיג למספר הדוגמאות ($samples$) עליהם צריך להתאמן האלגוריתם. לרוב נרצה לקבוע ϵ,δ בהתאם לדרישות עבור פתרון הבעיה הלימודית ופונקציה זו למעשה נותנת חסם תחתון למספר הדוגמאות עליהם נתאמן.

טענה 6.2 יהיו $\epsilon,\delta\in\left(0,1\right)$. אם $m\geq\left\lceil \frac{log\left(|\mathcal{H}|/\delta\right)}{\epsilon}\right\rceil$ אזי לכל $\mathcal{D},f$ בהסתברות לפחות $1-\delta$ מתקיים

$$L_{\mathcal{D},f}\left(ERM_{\mathcal{H}}\left(S\right)\right)\leq\epsilon$$

.

מסקנה 6.3 עבור $\mathcal{H}$ מחלקת היפותזות סופית אזי $\mathcal{H}$ הינה $PAC\ learnable$ עם $m_{\mathcal{H}}\left(\epsilon,\delta\right)\leq\left\lceil \frac{log\left(|\mathcal{H}|/\delta\right)}{\epsilon}\right\rceil$.

הוכחה לטענה ניתן למצוא בספר $Understanding\ Machine\ Learning$ בעמודים $39-40$.

6.3 $Agnostic\,PAC$

כפי שראינו עד כה ב־$PAC\,learning$ אנו מניחים את קיומה של פונקציית תיוג נכונה f שבמחלקת ההיפותזות. הנחה זו מגבילה אותנו שכן יתכן כי לא קיימת פונקציה כזו כלל או שהיא איננה במחלקת ההיפותזות. על כן נרצה להרחיב את המודל להתמודד עם מצב זה.

6.3.1 $Data\,Generating\,Distribution$

הנחת ה־$realizability$ למעשה קבעה כי f זו מתייגת נכון בהסתברות 1 כל $\mathcal{X}\sim\mathcal{D}$. על מנת להשתחרר מהגבלה זו נגדיר את הפילוג על מרחב התיוגים בנוסף למרחב הדוגמאות: $\mathcal{D}\sim\mathcal{Z}=\mathcal{Y}\times\mathcal{X}$. נכנה את $\mathcal{D}$ בתוך ה־$joint\,distribution$ מעל נקודות ב$domain$ ותיוגים כאשר:

- $\mathcal{D}_{\mathcal{X}}$ הינה ההתפלגות מעל הנקודות ב־$domain$ ומכונה $marginal\,distribution$.
- $\mathcal{D}\left(\left(x,y\right)|x\right)$ הינה ההסתברות המותנית לתיוגים עבור נקודות ב־$domain$.

נשים לב כי אילו מחלקת היפותזות $\mathcal{H}$ הינה $Agnostic\,PAC\,learnable$ (אשר נראה הגדרה מלאה בהמשך) ונניח כעת כי אכן מתקיימת הנחת ה־$realizability$ אזי למעשה הפונקציה המתייגת נכונה בהסתברות 1 תספק את ה־ $y\in\mathcal{Y}$ הנכון ואת יתר התיוגים בהסתברות 0. על כן נוכל להסתכל על PAC כמקרה פרטי של $Agnostic\,PAC$.

6.3.2 $True\,error/Empirical\,Risk$

עבור $\mathcal{Z}=\mathcal{Y}\times\mathcal{X}$ אזי פונקציית האיבוד מוגדרת בתוך $l:\mathcal{H}\times\mathcal{Z}\rightarrow\mathbb{R}_{+}$ ועל כן ה־$risk\,function$ הינו האיבוד המצופה ממסווג $h\in\mathcal{H}$ ביחס לפילוג $\mathcal{D}$ מעל $\mathcal{Z}$:

$$L_{\mathcal{D}}\left(h\right)=\mathbb{E}_{\ddagger\sim\mathcal{D}}\left[l\left(h,z\right)\right]$$

ונגדיר את ה־$empirical\,risk$ מעל $S=\left(z_{1},...,z_{m}\right)\subseteq\mathcal{Z}^{m}$ בתור:

$$L_{S}\left(h\right)=\frac{1}{m}\sum_{i=1}^{m}l\left(h,z_{i}\right)$$

הגדרת ה־ERM, שכן הינה על סט האימון, נשארת זהה:

$$ERM_{\mathcal{H}}\left(S\right)\in\underset{h\in\mathcal{H}}{argmin}L_{S}\left(h\right)=\underset{h\in\mathcal{H}}{argmin}\frac{1}{m}\left|\left\{x_{i}\in S\,|\,h\left(x_{i}\right)\neq y_{i}\right\}\right|$$

כעת נראה את ההגדרה הפורמלית של $Agnostic\,PAC$:

הגדרה 6.4 נאמר כי מחלקת היפותזות $\mathcal{H}$ הינה $Agnostic\,PAC\,learnable$ אם קיימים פונקציה $m_{\mathcal{H}}\,:\,(0,1)^2\,\rightarrow\,\mathbb{N}$ ואלגוריתם למידה $\mathcal{A}$ עם התכונה: שלכל $\epsilon,\delta\in(0,1)$ ולכל התפלגות $\mathcal{D}$ מעל $\mathcal{X}\times\mathcal{Y}$ כאשר מריצים את $\mathcal{A}$ על $m\geq m_{\mathcal{H}}(\epsilon,\delta)$ דוגמאות $i.i.d$ מפילוג $\mathcal{D}$אזי האלגוריתם מחזיר היפותזה h כך ש:

$$\mathcal{D}^m\left(\{S\in\mathcal{Z}^m\,:\,L_{\mathcal{D}}(h)\leq min_{h'\in\mathcal{H}}L_{\mathcal{D}}(h')+\epsilon\}\right)\geq 1-\delta$$

6.4 דוגמה - $Coin\,Prediction$

כבעיית חימום לסיווג נתעסק בשערוך תוצאת הטלת מטבע. בהינתן משתנה מקרי z המייצג הטלת מטבע עם סיכוי הצלחה p נרצה ללמוד מהו p. המטבע הינו $z\sim Ber(p)$.

יהיו $z_1,...,z_m\sim\mathcal{D}_z$ מ"מ ש"ה ב"ת (כלומר $i.i.d$) סט אימון S אשר ההתפלגות שלו $\mathcal{D}_z^m$. אם כך הקלט לבעיית הלמידה $\mathcal{A}$ הינו S (אקראי שנבחר בהתאם ל־$\mathcal{D}^m$ והפלט, הפרדיקציה, הינה $\mathcal{A}(S)$ ונסמן $\hat{p}$.

היות ואופן התפלגות המידע אינו ידוע לנו נרה להכניס פרמטר $accuracy$ למערכת ($\epsilon>0$) ונרשה ל־$\hat{p}$ לסטות מ־p בלכל היותר ϵ. בנוסף, יתכן כי סט האימון איננו מייצג את ההתפלגות הכללית ולכן נכניס $confidence$ למערכת ($\delta>0$) אשר יבטיח כי

$$\mathbb{P}\left(\{S\,:\,|\hat{p}-p|>\epsilon\}\right)<\delta$$

על כן באופן פורמלי אלגוריתם למידה $\mathcal{A}$ עבור $coin\,prediction$ הינו $f:\cup_{m=1}^{\infty}\{0,1\}^m\rightarrow[0,1]$ המקיימת את ההבאים: קיימת פונקציה $m:(0,1)^2\rightarrow\mathbb{N}$ כל שלכל $\epsilon,\delta\in(0,1)$ ולכל התפלגות $\mathcal{D}$ המתאימה ל־ $p\in[0,1]$, כאשר נריץ את $\mathcal{A}$ על $m\geq m_{\mathcal{A}}(\epsilon,\delta)$ דוגמאות iid על פי $\mathcal{D}^m$ האלגוריתם יחזיר $\hat{p}$ כך שבהסתברות לפחות $1-\delta$ נקבל כי $|\hat{p}-p|\leq\epsilon$.

בעיה מטילים מטבע בעל הסתברות p לקבלת H m פעמים. רוצים לשערך את p.

פתרון מרחב ההסתברות הינו $\Omega=\{H,T\}^m$ ו־ $D^m=\frac{1}{2^m}$ (כלומר יוניפורמי). נגדיר $z_1,...,z_m$ כאשר $z_i=\mathbb{1}_{w_i=H}$. האלגוריתם יקבל כקלט סט ערכים של m הטלות והפלט הינו $\bar{z}=\frac{1}{m}\sum_{i=1}^m z_i$. נשים לב כי $\mathbb{E}[\bar{z}]=p$ ומאי שיוויון הופדינג נקבל כי

$$\mathcal{D}^m\left[|\bar{z}-p|\geq\epsilon\right]\leq 2exp\left(-2m\epsilon^2\right)$$

נרצה כי ביטוי זה יהיה קטן מ־δ ולכן:

$$2exp\left(-2m\epsilon^2\right)\leq\delta\;\Leftrightarrow\;-2m\epsilon^2\leq ln\left(\frac{\delta}{2}\right)\;\Leftrightarrow\;\frac{1}{2\epsilon^2}ln\left(\frac{2}{\delta}\right)\leq m$$

7 $VC - Dimension$

על אף שראינו כי סופיות של מחלקת היפותזות $\mathcal{H}$ הינו תנאי מספיק ללמידה הוא איננו תנאי הכרחי. כדי לספק תנאי רחב יותר גם עבור מחלקות אינסופיות נזכר ברעיון ההוכחה של $No\,Free\,Lunch$. במקרה המשפט הראינו שללא הגבלת מחלקת ההיפותזות לכל אלגוריתם, באמצעות שיטת היריב נוכל ליצור התפלגות עליה אלגוריתם למידה אחד ייכשל ואחר יצליח.
בשביל לעשות זאת השתמשנו בתת קבוצה סופית של הדומיין $C \subset \mathcal{X}$. שיטת היריב החזירה אלגוריתם שעבור קבוצה סופית זו בחרה בפונקציית תיוג שתיכשל על קלט זה.

הגדרה 7.1 תהי $\mathcal{H}$ מחלקת היפותזות מ־$\mathcal{X}$ ל־$\{0,1\}$ ותהי $C = \{c_1, ..., c_m\} \subset \mathcal{X}$. נאמר כי ה־$restriction$ של $\mathcal{H}$ על C הינה תת קבוצת הפונקציות מ־C ל־$\{0,1\}$ שניתן לקבל מ־$\mathcal{H}$:

$$\mathcal{H}_C = \{h(c_1), ..., h(c_m) : h \in \mathcal{H}\}$$

הגדרה 7.2 נאמר כי מחלקת היפותזות $\mathcal{H}$ מנתצת ($shatters$) קבוצה וסופית $C \subset \mathcal{X}$ אם ה־$restriction$ של $\mathcal{H}$ על C הינו סט <u>כל הפונקציות</u> מ־C ל־$\{0,1\}$. כלומר $|\mathcal{H}_C| = 2^{|C|}$.

הגדרה 7.3 מימד ה־VC ($VC - Dimension$) של מחלקת היפותזות $\mathcal{H}$, מסומן ב־$VCdim(\mathcal{H})$, הינו הגודל המקסימלי של $C \subset \mathcal{X}$ אשר $\mathcal{H}$ מנתצת את C.

משפט 7.4 תהי מחלקת היפותזות $\mathcal{H}$ בעלת מימד VC אינסופי אזי $\mathcal{H}$ איננה $PAC\,learnable$.

לצורך הוכחת המשפט קיימת טענת עזר (6.4 בספר $understanding\,machine\,learning$, עמוד 69) שאם $\mathcal{H}$ מנתצת סט מגודל $2m$ לא נוכל ללמוד את $\mathcal{H}$ באמצעות m דוגמאות בלבד.

8 $Uniform\,Convergence$

נראה תנאי, $uniform\,convergence$, אשר הינו תנאי מספיק ללמידה. ראינו כי ERM מחזיר היפותזה הממזערת שגיאה על האימון ובתקווה שכך גם על $\mathcal{D}$ הכללי. על כן מספיק לבקש שה־$empirical\,risk$ של כל $h\in\mathcal{H}$ מקרב מספיק ל־$true\,error$.

הגדרה 8.1 סט אימון S הינו ϵ־מייצג $(\epsilon-representative)$ (ביחס לדומיין $\mathcal{Z}$, מחלקת היפותזות $\mathcal{H}$, פונקציית איבוד l והתפלגות $\mathcal{D}$) אם

$$\forall h\in\mathcal{H}\ \ |L_S(h)-L_{\mathcal{D}}(h)|\leq\epsilon$$

למה 8.2 יהי סט אימון S אשר הינו $\frac{\epsilon}{2}$־מייצג אזי לכל פלט של $ERM_{\mathcal{X}}(S)$ מתקיים כי:

$$h_S\in\underset{h\in\mathcal{H}}{argmin}\,L_S(h)\quad L_D(h_S)\leq\underset{h\in\mathcal{H}}{min}L_{\mathcal{D}}(h)+\epsilon$$

הוכחה: תהי $h\in\mathcal{H}$ אזי:

$$L_{\mathcal{D}}(h_s)\overset{\frac{\epsilon}{2}-rep.}{\leq}L_S(h_s)+\frac{\epsilon}{2}\overset{min\,of\,group}{\leq}L_S(h)+\frac{\epsilon}{2}\overset{\frac{\epsilon}{2}-rep.}{\leq}L_{\mathcal{D}}(h)+\frac{\epsilon}{2}+\frac{\epsilon}{2}=L_{\mathcal{D}}(h)+\epsilon$$

■

על כן נסיק כי עבור סט $\frac{\epsilon}{2}$־מייצג עיקרון ה־ERM יחזיר היפותזה טובה.

הגדרה 8.3 מחלקת היפותזות $\mathcal{H}$ הינה בעלת התכונה $uniform\,convergence$ אם קיימת פונקציה $m_{\mathcal{H}}^{UC}:(0,1)^2\to\mathbb{N}$ כך שלכל $\epsilon,\delta\in(0,1)$ ולכל התפלגות $\mathcal{D}$ מעל $\mathcal{Z}$ אז

$$\mathcal{D}^m\left(\{S\in\mathcal{Z}^m\ :\ S\,is\,\epsilon-representative\}\right)\geq1-\delta$$

מסקנה 8.4 אם $\mathcal{H}$ הינה $uniform\,convergence$ עם $m_{\mathcal{H}}^{UC}$ אזי $\mathcal{H}$ הינה $agnostic\,PAC\,learnable$ עם $sample\,complexity$ של $m_{\mathcal{H}}\leq m_{\mathcal{H}}^{UC}\left(\frac{\epsilon}{2},\delta\right)$ ופרדיגמת $ERM_{\mathcal{H}}$ הינה $agnostic\,PAC\,learner$ טובה עבור $\mathcal{H}$.

טענה 8.5 תהי $\mathcal{H}$ מחלקת היפותזות סופית ופונקציית איבוד l חסומה ב־$[0,1]$ אזי $\mathcal{H}$ הינה $agnostic\,PAC\,learnable$ באמצעות $ERM_{\mathcal{H}}$ ו־$sample\,complexity$:

$$m_{\mathcal{H}}(\epsilon,\delta)\leq\left\lceil\frac{2log\left(\frac{2|\mathcal{H}|}{\delta}\right)}{\epsilon^2}\right\rceil$$

9 המשפט היסודי של הלמידה הסטטיסטית

ישנן שתי גרסאות שונות למשפט. הראשונה מספקת חסמים עבור פונקציית ה־$sample\,complexity$ והשנייה מספקת תנאים שקולים ללמידה.

משפט 9.1 תהי $\mathcal{H}$ מחלקת היפותזות מדומיין $\mathcal{X}$ ותיוגים $\{0,1\}$ ותהי L^{0-1} פונקציית האיבוד. נניח כי $VCdim\left(\mathcal{H}\right)=d<\infty$ אזי קיימים קבועים אוניברסליים C_1,C_2 כך שמתקיים:

1. $\mathcal{H}$ **הינה** $PAC\,learnable$ **עם**

$$C_1\cdot\frac{d+log\left(\frac{1}{\delta}\right)}{\epsilon}\leq m_{\mathcal{H}}\left(\epsilon,\delta\right)\leq C_2\cdot\frac{d\cdot log\left(\frac{1}{\epsilon}\right)+log\left(\frac{1}{\delta}\right)}{\epsilon}$$

2. $\mathcal{H}$ הינה $agnostic\,PAC\,learnable$ עם

$$C_1\cdot\frac{d+log\left(\frac{1}{\delta}\right)}{\epsilon^2}\leq m_{\mathcal{H}}\left(\epsilon,\delta\right)\leq C_2\cdot\frac{d+log\left(\frac{1}{\delta}\right)}{\epsilon^2}$$

3. $\mathcal{H}$ בעלת תכונת ה־$uniform\,convergence$ עם

$$C_1\cdot\frac{d+log\left(\frac{1}{\delta}\right)}{\epsilon^2}\leq m_{\mathcal{H}}^{UC}\left(\epsilon,\delta\right)\leq C_2\cdot\frac{d+log\left(\frac{1}{\delta}\right)}{\epsilon^2}$$

($*$) רק המקרה הראשון הוצג בכיתה.

משפט 9.2 תהי $\mathcal{H}$ מחלקת היפותזות מדומיין $\mathcal{X}$ ותיוגים $\{0,1\}$ ותהי L^{0-1} פונקציית האיבוד. התנאים הבאים שקולים:

1. $\mathcal{H}$ הינה $uniform\,convergence$.
2. כל היפותזת ERM הינה $agnostic\,PAC\,learner$ טובה עבור $\mathcal{H}$.
3. $\mathcal{H}$ הינה $agnostic\,PAC\,learnable$.
4. $\mathcal{H}$ הינה $PAC\,learnable$.
5. כל היפותזת ERM הינה $PAC\,learner$ טובה עבור $\mathcal{H}$
6. $VCdim\left(\mathcal{H}\right)<\infty$.

10 הבנת $Bias - Complexity\, Tradeoff$

נראה כי כאשר בוחרים מחלקת היפותזות לבעיית למידה נרצה למעשה לנסות להשיג 2 דברים:

1. נרצה שמחלקה זו תכיל פונקציית תיוג נכונה או לכל הפחות שהפונקציה בעלת האיבוד הנמוך ביותר במחלקה אכן נמוך מספיק לצורכינו. על כן נרצה להגדיל את המחלקה.

2. לא נוכל לבחור מחלקה רחבה מדי שכן אחרת מ־$No\, Free\, Lunch$ נקבל כי לא נצליח ללמוד את הדומיין. על כן נרצה להקטין את המחלקה.

10.1 $Error\, Decomposition$

כדי לראות כיצד להשיג 2 מטרות אלה נפרק את השגיאה של לומד $ERM_{\mathcal{H}}$ לשני מרכיבים. עבור h_S היפותזת $ERM_{\mathcal{H}}$ נאמר:

$$L_{\mathcal{D}}\left(h_S\right) = \underbrace{\epsilon_{app}}_{bias} + \underbrace{\epsilon_{est}}_{variance\, as\, in\, complexity}$$
$$\epsilon_{app} = min_{h\in\mathcal{H}} L_{\mathcal{D}}\left(h\right)$$
$$\epsilon_{est} = L_{\mathcal{D}}\left(h_S\right) - \epsilon_{app}$$

כאשר:

1. <u>$The\ Approximation\ Error$</u> ־ השגיאה המינימלית שיכולה להתקבל על ידי פונקציה במחלקה. למעשה שגיאה זו מייצגת את כמות ה"סיכון" בהגבלתנו את מחלקת ההיפותזות (כמה עשירה המחלקה). כלומר ה־$inductive\, bias$ שקיים במערכת. שגיאה זו למעשה איננה תלויה בגודל סט האימון אלא רק במחלקת ההיפותזות. במקרה שמתקיימת הנחת ה$realizability$ שגיאה זו הינה 0. באופן כללי $bias$ מתייחס להעדפה של דבר אחד על פני אחר כתוצאה מדעה מוקדמת.

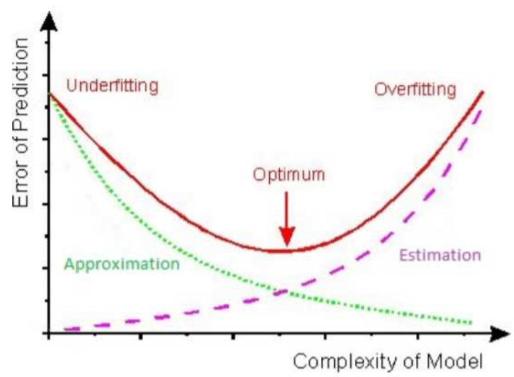
2. <u>$The\ Estimation\ Error$</u> ־ המרחק בין שגיאת ה־$approximation$ לבין השגיאה שמתקבלת על ידי הפרדיקציה של ה־$ERM_{\mathcal{H}}$. שגיאה זו נובעת מכך שה־$empirical\, risk$ הינה רק שערוך של השגיאה האמיתית על $\mathcal{D}$. איכות שגיאה זו תלויה בגודל סט האימון ובגודל (או $VCdim$) מחלקת ההיפותזות. נוכל להתייחס לגודל של $\mathcal{H}$ כמדד ל־$complexity$ שלה.

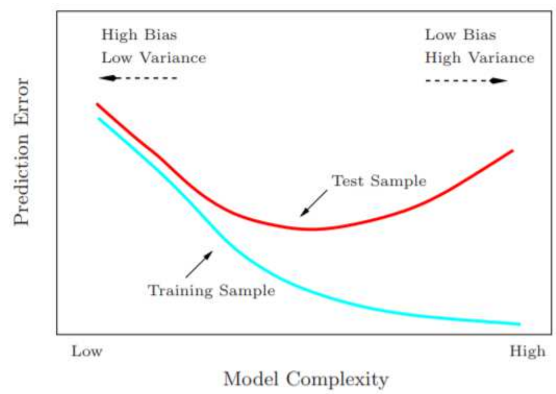
היות ובאופן כללי נרצה לצמצם את השגיאה הכוללת אנו ניצבים בפני $tradeoff$ בין שני אלה. עבור מחלקה עשירה מאוד נוריד את שגיאת ה־$approximation$ אבל נסתכן בהגדלת שגיאת ה־$estimation$ ונסתכן ב־$overfitting$ (הליכה אחרי הרעש האינרנטי בדגימות). עבור מחלקה מצומצמת נוריד את שגיאת ה־$estimation$ אבל נסתכן בהגדלת שגיאת ה־$approximation$ וב־$underfotting$.

10.2 $Tradeoff$

ראינו כי ה־$bais$ מודד את המרחק בין החיזוי ה"ממוצע" לאמת וה־$variance$ מודד כמה וריאבילית הפרדיקציה.

- יותר מידע $\Leftarrow$ פחות $variance$.
- מודל יותר $complicated$ $\Leftarrow$ יותר $variance$.
- יותר רעש σ^2 $\Leftarrow$ יותר $variance$.

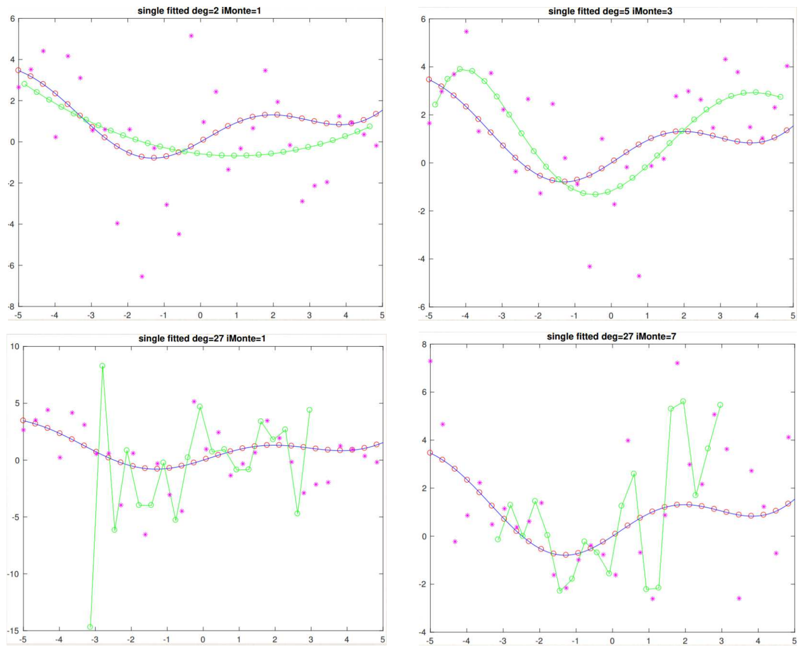




ברמת האינטואיציה עבור מודל פשוט יהיה לנו $bias$ גבוה ו־$variance$ נמוך מכיוון מוחלקת ההיפותזות פשוטה ונקבל שגיאה גבוהה. עבור מודל מורכב נקבל $bias$ נמוך ו־$variance$ גבוהה מכיוון ו"נלך אחרי הרעש" שבדגימות ונקבל שגיאה גבוהה. על כן לכל מודל נצטרך להבין מהם המנופים שנוכל לשחק בהם כדי למצוא את נקודת המינימום שבסט ה־$test$.

10.3 דוגמה עבור פולינומים

בתמונות הבאות: הנקודות הירוקות ־ החיזוי, הנקודות אדומות ־ ערך אמת, כוכביות ־ דוגמאות האימון כאשר ההבדל בינן לבין האדומות נובע מהכנסת רכיב הרעש, הקו הירוק ־ המסווג, הקו הכחול המסווג ה"מושלם".



מה שאנו רואים זה שפולינומים ממעלה נמוכה לא מספקים פרדיקציה טובה מספיק ($underfitting$) בעוד שבמעלות גבוהות המסווג "עונה כל כך טוב" על הדוגמאות שלמעשה הוא מסווג את הרעש ולא את ה־$data$ שמאחורי הרעש. המסווג מאוד "גמיש" והוא "רץ אחרי הרעש". מקרה זה מתאר $overfitting$.

חלק IV

תיאוריה לאלגוריתמים

בחלק זה נראה אוסף אלגוריתמים הפותרים בעיות למידה. חלק מהאלגוריתמים מספקים פתרון לעולם תוכן מסויים (לדוגמה רגרסיה לינארית) בעוד שאחרים הינם אלגוריתמים לשיפור תכונה כללית מסוימת (כמו $boosting$).

11 מסווגים לינאריים

הגדרה 11.1 מחלקת הפונקציות האיפיניות הינה $L_d = \left\{h_{w,b} : w \in \mathbb{R}^d,\, b \in \mathbb{R}\right\}$ כאשר $h_{w,b}(x) = \langle w, x\rangle + b = \left(\sum_{i=1}^{d} w_i x_i\right) + b$. נוח לסמן בתור $L_d = \left\{x \mapsto \langle w, x\rangle + b \,:\, w \in \mathbb{R}^d, b \in \mathbb{R}\right\}$.

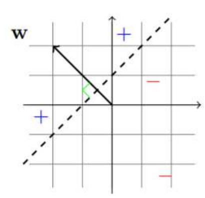
במקרים רבים נוח לשלב את b, הנקרא גם ה־$bias$, לתוך w בקואורדינטה ה־0 ולהוסיף לכל $x \in \mathcal{X}$ קואורדינטה ראשונה עם הערך 1. כלומר $w' = (b, w_1, ..., w_d) \in \mathbb{R}^{d+1},\ x' = (1, x_1, ..., x_d) \in \mathbb{R}^{d+1}$ ומכאן כי: $h_{w,b}(x) = \langle w, x\rangle + b = \langle w', x'\rangle$.

11.1 חצאי מרחב ($Half spaces$)

נתבונן במחלקת ההיפותזות של מפרידים לינאריים עבור בעיות קלסיפיקציה ($|\mathcal{Y}| = 2$ לרוב $\mathcal{Y} = \{\pm 1\}$ או $\mathcal{Y} = \{0,1\}$):

$$HS_d = sign \circ L_d = \{x \mapsto sign\left(h_{w.b}(x)\right) \,:\, h_{w,b} \in L_d\}$$

במקרה של $d = 2$ היפותזה המוחזרת מהמחלקה מגדירה על־מישור מפריד אשר מאונך ל־w. דגימות (ערכי קלט x) מעל העל־מישור יקבלו $+$, ודגימות מתחת יקבלו $-$.



11.1.1 פתרון ERM עבור $half spaces$ באמצעות $Perceptron$

אלגוריתם ה־$Perceptron$ עובד באופן איטרטיבי ליצירת סדרת וקטורים $w^{(1)}, w^{(2)}, ...$ כאשר בכל איטרציה t האלגוריתם מוצא $sample$ x_i אשר $w^{(t)}$ טעה ($sign(\langle w, x_i\rangle) \neq y_i$) ומעדכן את $w^{(t)}$ על ידי הוספת $x_i y_i$. אנו מקבלים:

$$y_i \left\langle w^{(t+1)}, x_i \right\rangle = y_i \left\langle w^{(t)} + y_i x_i, x_i \right\rangle = y_i \left\langle w^{(t)}, x_i \right\rangle + ||x||^2$$

כלומר כאשר מעדכנים את הוקטור w מכוונים אותו מעט לכיוון ה־$sample$ ה־i

```
Batch Perceptron

input: A training set (x_1, y_1), ..., (x_m, y_m)
initialize: w^(1) = (0, ..., 0)
  for t = 1, 2, ...
    if (∃ i  s.t.  y_i⟨w^(t), x_i⟩ ≤ 0) then
      w^(t+1) = w^(t) + y_i x_i
    else
      output w^(t)
```

משפט 11.2 יהיו $S=\{(x_i,y_i)\}_{i=1}^{m}$ ברי הפרדה לינארית, $B=min\left\{||w||\ :\ \forall i\in[m]\ y_i\left\langle w,x_i\right\rangle\geq 1\right\}$ ו־$R=\underset{i}{max}\,||x_i||$ אזי האלגוריתם יעצור אחרי לכל היות $(RB)^2$ איטרציות ובסופן $\forall i\in[m]\ \ y_i\left\langle w^{(t)},x_i\right\rangle>0$.

הערה 11.3 אלגוריתם $Perceptron$ בהכרח יתכנס, אולם קצב ההתכנסות תלוי ב־B שבמקרים מסוימים עשוי להיות אקספוננציאלי ב־d.

11.1.2 פתרון ERM עבור $halfspaces$ באמצעות תכנון לינארי

עבור המקרה של חצאי־מרחב הומוגניים (נוכל להרחיב ללא הומוגניים באופן דומה למתואר בתחילת הפרק) בהינתן סט אימון S אזי היפותזת ERM הינה וקטור $w\in\mathbb{R}^d$ אשר:

$$\forall\ i\in[m]\quad sign\left(\left\langle w,x_i\right\rangle\right)=y_i\quad\Longleftrightarrow\quad y_i\cdot\left\langle w,x_i\right\rangle>0$$

נציג זאת בצורה סטנדרטית של תכנון לינארי. תחת הנחת הריאלזביליות קיים w^* המקיים את כל התנאים אזי נוכל להניח כי

$$\forall\ i\in[m]\quad y_i\left\langle w,x_i\right\rangle\geq 1$$

אזי נגדיר

$$\gamma=\underset{i}{min}\left\{y_i\left\langle w,x_i\right\rangle\right\}$$

ועבור $\bar{w}=\frac{1}{\gamma}w^*$ מתקיים:

$$\forall\ i\in[m]\ \ y_i\left\langle\bar{w},x_i\right\rangle=\frac{y_i\left\langle w^*,x_i\right\rangle}{\gamma}\geq 1$$

ובצורה מטריציונית נקבל:

$$y_i\left\langle w,x_i\right\rangle=y_ix_i^Tw$$

$$\begin{bmatrix}- & y_1x_1^T & -\\ \\ - & y_mx_m^T & -\end{bmatrix}w\geq 1_m$$

11.1.3 $VCdim$ של $halfspaces$

משפט 11.4 ה־$VCdim$ של מחלקת ההיפותזות של ה־$halfspaces$ ההומוגניים מעל $\mathbb{R}^d$ הוא d.

הוכחה: ■

1. תהי $\mathcal{X} \supset C = \{e_1, ..., e_d\}$ קבוצת וקטורי הבסיס הסטנדרטי ($(e_i)_j = \mathbb{1}_{i=j}$). נראה כי לכל תיוג $y_1, ..., y_d \in \{-1,1\}^d$ נבחר $w = (y_1, ..., y_d)^T$ ואז $\forall i \in [d] \;\; \langle w, e_i \rangle = y_i$. על כן $\mathcal{H}$ מנתצת את C ומתקיים $VCdim\left(\mathcal{H}\right) \geq d$.

2. תהי $\mathcal{X} \supset C = \{x_1, ..., x_{d+1}\}$. היות ו־$dim\left(\mathbb{R}^d\right) = d$ קבוצה זו תלויה לינארית ולכן קיימים $a_1, ..., a_{d+1} \in \mathbb{R}$ לא כולם אפס כך ש: $\sum a_i x_i = 0$. נסמן $I = \{i \,:\, a_i > 0\}\,,\; J = \{\, i \,:\, a_i < 0\}$. בהכרח אחת הקבוצות לא ריקה.

- אם שתי הקבוצות לא ריקות אזי מתקיים $\sum\limits_{i \in I} a_i x_i = \sum\limits_{j \in J} |a_j|\, x_j$. נניח בשלילה כי $\mathcal{H}$ מנתצת את C ($|C| = d+1$) אזי קיים w כך ש: $\forall i \in I \;\; \langle w, x_i \rangle > 0 \;\; \wedge \forall j \in J \;\; \langle w, x_j \rangle < 0$. מכאן נובע כי:

$$0 < \sum_{i \in I} a_i \langle w, x_i \rangle = \left\langle w, \sum_{i \in I} a_i x_i \right\rangle = \left\langle w, \sum_{j \in J} |a_j|\, x_j \right\rangle = \sum_{j \in J} |a_i| \langle w, x_j \rangle < 0$$

בסתירה.

- אם אחת הקבוצות ריקה, נניח בה"כ J, נקבל בהתאם:

$$0 < \sum_{i \in I} a_i \langle w, x_i \rangle = \left\langle w, \sum_{i \in I} a_i x_i \right\rangle = \left\langle w, \sum_{j \in J} |a_j|\, x_j \right\rangle = \langle w, 0 \rangle = 0$$

בסתירה.

משפט 11.5 חה־$VCdim$ של מחלקת ההיפותזות של ה־$halfspaces$ הלא הומוגניים מעל $\mathbb{R}^d$ הוא $d+1$.

הוכחה: תחילה נוודא ניתוץ של הקבוצה $\mathcal{X} \supset C = \{0, e_1, ..., e_d\}$ כשם שהוכחנו מקודם. לאחר מכן, באופן זהה להוכחה לעיל, נראה שלא ניתן לנתץ קבוצה $\mathcal{X} \supset C = \{x_1, ..., x_{d+2}\}$. נסיק כי $VCdim\left(HS_d\right) = d+1$ כנדרש. ■

11.2 רגרסיה לינארית

11.2.1 פתרון הבעיה

נציג תחילה עולם בעיה שנרצה ללמוד ונראה כיצד בעזרת המונחים השונים נתן אלגוריתם למידה עבור הבעיה. נניח שאנו סוחרי נדלן וברצוננו לשערך באיזה מחיר נוכל למכור נכס מסויים. עבור נכס ספציפי נוכל להתייחס לגודלו, למספר החדרים, למספר חדרי השינה, האם בבניין או בית פרטי, באיזה קומה, האם יש נוף, מצב הנכס ופרמטרים רבים נוספים. כל אלה הם ה־$features$ שבעזרתם ננסה לשערך את המחיר $label$.

על כן בעולם בעיה זו ה־$domain$ שלנו הוא נכסים, ה־$features$ הם התכונות שתיארנו וה־$labels$ הם מחיר הנכס. נשים לב שהתיוגים חיים ב־$\mathbb{R}$ ־ ולכן מדובר פה בבעיית למידה של רגרסיה. באופן פורמלי: $\mathcal{X} = \mathbb{R}^d,\ \mathcal{Y} = \mathbb{R}$ ואנו מחפשים $f : \mathcal{X} \to \mathcal{Y}$.

כעת נרצה לתאר את מחלקת ההיפותזות. בהתאם לעיקרון ה־$no\, free\, lunch$ עלינו להניח דבר מה על מחלקת ההיפותזות. בצורת למידה זו אנו מניחים כי התיוגים $\mathcal{Y}$, בדוגמה הנ"ל מחירי הנכסים, הינם לינאריים ב־$features$ אזי החלקת ההיפותזות הינה מהצורה:

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

בנוסף נניח כי מתקיימת $the\, realizability\, assumption$ אזי קיימת $\hat{f} : \mathcal{X} \to \mathcal{Y}$ אשר מתייגת נכונה את הדגימות ונניח כי $f \in \mathcal{H}_{reg}$. בהגדרה זו כל $feature$ מקבל משקל w_i וקיים w_0 אשר הוא ה־$intercept$ ־ ההסטה של הישרייה מראשית הצירים. למידת בעיה זו משמע מציאת הוקטור $w \in \mathbb{R}^{d+1}$.

כעת נבין כיצד נראה ה־$data$ עליו אנו עובדים. תחילה לכל דגימה ב־$\mathcal{X}$ נראה כי היא למעשה וקטור $(x_1, ..., x_d) \in \mathbb{R}^d$. היות וקיים ה־$intercept$ נבחר לתאר את הדגימות במימד גבוה יותר: $(1, x_1, ..., x_d) \in \mathbb{R}^{d+1}$. לכן עבור m דגימות ($samples$) ו־d $features$ נוכל לארגן את סט האימון במטריצה:

$$X = \begin{bmatrix} | & & | \\ x_1 & \dots & x_m \\ | & & | \end{bmatrix},\ x_i \in \mathbb{R}^{d+1}$$

הגדרה 11.6 תהי $X \in \mathbb{R}^{d \times m}$ ופירוק SVD $X = U\Sigma V^T$ אזי ה־ $Moore–Penrose\, pseudo\, inverse$ של x הוא:

$$X^{\dagger} = V\Sigma^{\dagger}U^T \quad \Sigma^{\dagger}_{i,i} = \begin{cases} (\Sigma_{i,i})^{-1} & \Sigma_{i,i} \neq 0 \\ 0 & \Sigma_{i,i} = 0 \end{cases}$$

טענה 11.7 לימוד בעיית הרגרסיה הלינארית שקול למציאת פתרון w למערכת המשוואות $X^T w = y \iff XX^T w = Xy$.

הוכחה: נראה כי בהתאם למטריצה XX^T נוכל לחלק למקרים עבורם נמצא את הפתרון למערכת המשוואות:
תחילה נניח כי XX^T הפיכה ונוכיח כי קיים למערכת המשוואות פתרון יחיד וחילוצו: $\hat{w} = \left(XX^T\right)^{-1} Xy = \left(X^{\dagger}\right)^T y$.
היותר ו־ XX^T הפיכה קיימת לה מטריצה הפוכה. על כן $XX^T w = Xy \;\Leftrightarrow\; w = \left(XX^T\right)^{-1} Xy$. יהי פירוק SVD

של X: $X=U\Sigma V^T$. היות ו־XX^T הפיכה קיים לה פירוק EVD. נוכל לבחור כי המטריצה האורתונורמלית תהיה U מפירוק ה־SVD של X ולכן $\left(XX^T\right)^{-1}=UD^{-1}U^T$. מכאן:

$$\begin{aligned}\left(XX^T\right)^{-1}X &= \left(UD^{-1}U^T\right)\left(U\Sigma V^T\right)\\ &= UD^{-1}U^TU\Sigma V^T\\ &= UD^{-1}\Sigma V^T\\ &\overset{(*)}{=} U\Sigma^\dagger V^T\\ &= \left(V\left(\Sigma^\dagger\right)^TU^T\right)^T\\ &= \left(V\Sigma^\dagger U^T\right)^T\\ &= \left(X^\dagger\right)^T\end{aligned}$$

כאשר $(*)$ בגלל שלכל $i\neq j$ מתקיים כי $\left(D^{-1}\Sigma\right)_{i,j}=0$ וכאשר $i=j$ אזי

$$\left(D^{-1}\Sigma\right)_{i,i}=\sum_k\left(D^{-1}_{i,k}\cdot\Sigma_{k,i}\right)=D^{-1}_{i,i}\Sigma_{i,i}=\frac{\sigma_i}{\sigma_i^2}=\sigma_i^{-1}$$

ולכן פתרון מערכת המשוואות הוא $\hat{w}=\left(X^\dagger\right)^Ty$.

כעת נניח כי XX^T איננה הפיכה ונוכיח כי קיימים אינסוף פתרונות והמינימלי והוא: $||\hat{w}||=min\left\{||w||\ :\ X^Tw=y\right\}$.
XX^T איננה הפיכה אזי $dim\left(Ker\left(XX^T\right)\right)\neq 0$. אזי מתקיים גם כי $dim\left(X^T\right)\neq 0\ \wedge\ dim\left(X\right)\neq 0$. היות והגרעין איננו טריוויאלי למערכת המשוואות $X^Tw=y$ קיימים אינסוף פתרון או אין פתרון כלל. יהי $y\perp Ker\left(X\right)$ אזי $y\in Ker\left(X\right)^\perp$ ולכן $y\in Im\left(X^T\right)$. מכאן כי קיים $w\in\mathbb{R}^d$ עבורו מתקיים $X^Tw=y$ ונובע כי קיימים אינסוף פתרונות למערכת המשוואות.
על מנת לחלץ את הפתרון המינימלי יהיו $X=U\Sigma V^T,\ XX^T=UDU^T$ פירוקי $SVD,\,EVD$ של המטריצות בהתאמה אזי:

$$\begin{gathered}X^Tw=y\\ XX^Tw=Xy\\ UDU^Tw=U\Sigma V^Ty\\ \begin{bmatrix}d_1&&&\\&\ddots&&\\&&d_r&\\&&&0\end{bmatrix}U^Tw=\begin{bmatrix}\sigma_1&&&\\&\ddots&&\\&&\sigma_r&\\&&&0\cdots\quad 0\end{bmatrix}V^Ty\\ D^\dagger DU^Tw=D^\dagger\Sigma V^Ty\end{gathered}$$

כעת נשים לב כי:

$$D^{\dagger}D=\begin{bmatrix} d_1^{-1} & & & \\ & \ddots & & \\ & & d_r^{-1} & \\ & & & 0 \end{bmatrix}\begin{bmatrix} d_1 & & & \\ & \ddots & & \\ & & d_r & \\ & & & 0 \end{bmatrix}=\begin{bmatrix} 1 & & & \\ & \ddots & & \\ & & 1 & \\ & & & 0 \end{bmatrix}$$

$$D^{\dagger}\Sigma=\begin{bmatrix} d_1^{-1} & & & \\ & \ddots & & \\ & & d_r^{-1} & \\ & & & 0 \end{bmatrix}\begin{bmatrix} \sigma_1 & & & \\ & \ddots & & \\ & & \sigma_r & \\ & & & 0\cdots \quad 0 \end{bmatrix}=\begin{bmatrix} \sigma_1^{-1} & & & \\ & \ddots & & \\ & & \sigma_r^{-1} & \\ & & & 0\cdots \quad 0 \end{bmatrix}=\left(\Sigma^{\dagger}\right)^T$$

לכן נקבל כי:

$$U^Tw=\left(\Sigma^{\dagger}\right)^T V^Ty$$
$$\hat{w}=U\left(\Sigma^{\dagger}\right)^T V^Ty$$

■

מסקנה 11.8 למשוואות הנורמליות $XX^Tw=Xy$ בהכרח קיים פתרון (יחיד או אינסוף)

11.2.2 קיומו של רעש

כעת לאחר שראינו כיצד למצוא פתרון ללמידת רגרסיה לינארית נזכר כי בסט האימון שלנו קיים רעש ולכן למעשה הוא מהצורה $S=\{(x_i,f(x_i)+z_i)\}_{i=1}^m$ כאשר $z_1,...,z_m \overset{i.i.d}{\sim}\left(0,\sigma^2\right)$. כלומר מ"מ ב"ת ש"ה בעלי תוחלת 0 ושונות σ^2. על כן כאשר נחפש את הפתרון נחפש $X^Tw+z=y$ ומכאן כי $y\notin Im\left(X^T\right)$ ולמערכת ההומוגנית אין פתרון.

על כן בהתאם לעיקרון ERM נתבונן בפונקציית האיבוד $Least\,squares$. נראה כי מתקיים:

$$\sum_{(x_i,y_i)\in X,y}\left(y_i-X_i^Tw\right)^2=\left|\left|y-X^Tw\right|\right|^2=\left\langle y-X^Tw,y-X^Tw\right\rangle$$

התוצאה של $y_i - X_i^T w$ נקראת ה־$the\, i-th\, residual$ ומכאן במרה זה ה־ERM הינו $Residual\, Sum\, of\, Squares$:

$$
\begin{gathered}
RSS\left(w\right) = \left|\left|y - X^T w\right|\right|^2 \\
\Downarrow \\
\overbrace{\frac{\partial}{\partial w_i} RSS\left(w\right) = -2\sum_{i=1}^{m}\left(x_i\right)\cdot\left(y_i - X_i^T w\right) = 0} \\
\Downarrow \\
\nabla RSS\left(w\right) = -2X\left(y - X^T w\right) = 0 \\
\Downarrow \\
X\left(y - X^T w\right) = 0 \\
X^T y = X X^T w
\end{gathered}
$$

ומכאן כי אסטרטגיה זו אכן ממזערת את הטעות ומקיימת את עיקרון ה־ERM כנדרש.

11.2.3 $Numerically\, Stable$

נראה כי עולם הבעיה שתיארנו הינו ב־$\mathbb{R}^d$. היות וברצוננו לבטא כל זאת במחשב, ומחשב מסוגל לשמור רק מספר סופי (מצומצם) של ערכים אחרי הנקודה העשרונית ($floating\, point$) נרצה להתאים את המודל כך שיתמודד טוב יותר עם חוסר דיוק זה. לצורך ההבנה יתכן כי המחשב אינו יבדיל בין 0.00000 לבין 0.00001 ומכאן תנבא ההשפעה על תוצאת האלגוריתם. לכן עבור $\epsilon > 0$ שנבחר נגדיר את ה־ $Moore\text{–}Penrose\, pseudo\, inverse$ בתור:

$$
\Sigma_{i,i}^{\dagger,\epsilon} = \begin{cases} \left(\Sigma_{i,i}\right)^{-1} & \Sigma_{i,i} > \epsilon \\ 0 & \Sigma_{i,i} \le \epsilon \end{cases}
$$

11.2.4 $Bias - Variance\, Decomposition$

אחרי שאימנו מסווג $\hat{w}$ יהיו $\tilde{x_1}, ..., \tilde{x_k}$ דגימות חדשות אשר התיוגים האמתיים שלהן $\tilde{y_1}, ...\tilde{y_k}$. סיווגן במסווג מתבצע על ידי $\hat{\tilde{y}} = \tilde{X}^T \hat{w}$. על מנת לבדוק את איכות הסיווג:

$$
R\left(\tilde{y}, \hat{w}\right) = \mathbb{E}\left(\sum_{i=1}^{k}\left(\tilde{y_i} - \tilde{x_i}^T \hat{w}\right)^2\right) = \mathbb{E}\left|\left|\tilde{y} - \tilde{X}^T \hat{w}\right|\right|^2
$$

כאשר מקור המקריות (ובגלל זה תוחלת) הוא הרעש שבתיוגים y ־ ומכאן גם הרעש ב$\hat{w}$. כעת כאשר נבדוק את ביטוי זה נקבל:

$$\begin{aligned}\mathbb{E}\left|\left|\tilde{y}-\tilde{X}^{T}\hat{w}\right|\right|^{2} &= \mathbb{E}\left|\left|\tilde{y}-\hat{\tilde{y}}\right|\right|^{2}\\ &= \mathbb{E}\left|\left|\tilde{y}-\mathbb{E}\hat{\tilde{y}}+\mathbb{E}\hat{\tilde{y}}-\hat{\tilde{y}}\right|\right|^{2}\\ &= \mathbb{E}\left|\left|\tilde{y}-\mathbb{E}\hat{\tilde{y}}\right|\right|^{2}+\mathbb{E}\left|\left|\mathbb{E}\hat{\tilde{y}}-\hat{\tilde{y}}\right|\right|^{2}+2\mathbb{E}\left\langle \tilde{y}-\mathbb{E}\hat{\tilde{y}},\mathbb{E}\hat{\tilde{y}}-\hat{\tilde{y}}\right\rangle \\ &= \left|\left|\tilde{y}-\mathbb{E}\hat{\tilde{y}}\right|\right|^{2}+\sum_{i=1}^{m}Var\left(\hat{\tilde{y}}_{i}\right)\end{aligned}$$

אזי פירקנו את ה־$risk$ ל־$bias$ (משמאל) ול־$variance$ (מימין).

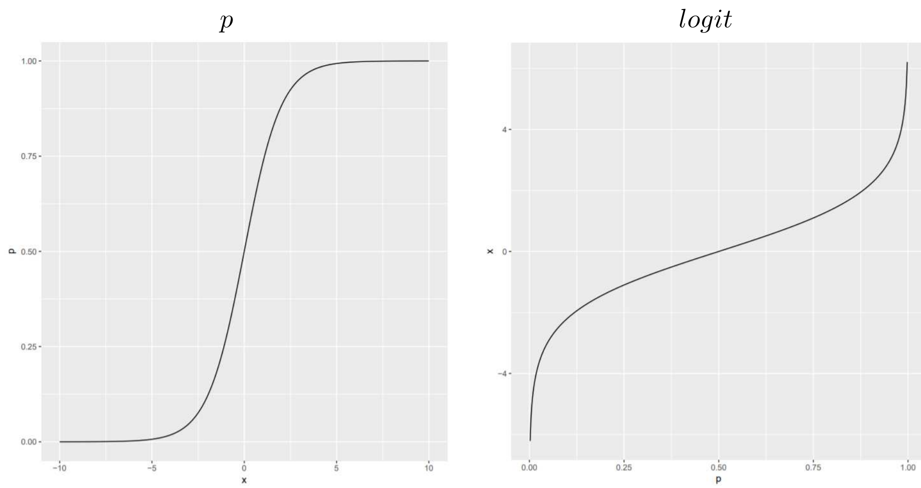
11.3 רגרסיה לוגיסטית

11.3.1 אינטואיציה

מודל הרגרסיה הלוגיסטית נובע מהרצון לשערך הסתברויות של סיווגים שונים (ב־$classification$ או $multiclassification$) באמצעות פונקציות לינאריות. במילים אחרות את ה־$likelihood$ של דגימה x להיות מסווגת k. במקרה של $classification$ אשר התיוגים הם $\mathcal{Y}=\{0,1\}$ אזי נשאל $\mathbb{P}\left(Y_i\mid X_i=x_i\right)$ כאשר היות ומדובר בקלסיפיקציה אזי עבור $X_i=x_i$ נקבל כי $Y_i|X_i=x_i\sim Ber\left(p_i\right)$ ונרצה טרנספורמציה מתאימה מהפונקציה הלינארית ל־$[0,1]$.

פונקציות קבות משמשות לרגרסיה הלוגיסטית כאשר מחלקה זו היא של פונקציות סיגמואידיות (מקבלות צורת S) החסומות ב־$[0,1]$. אנו נתבונן בפונקציית ה־$log-odds$ או ה־$logit$ $logit\left(p\right)=log\left(\frac{p}{1-p}\right)$. כעת כשנסתכל על הפונקציה ההופכית שלה נקבל: $p\left(x\right)=\frac{e^{x}}{1+e^{x}}$ וזוהי הפונקציה הלוגיסטית. תחת ההנחה כי הפונקציה $logit$ לינארית בתיוגים נקבל כי:

$$\begin{gathered}logit\left(y_i\right)=ax_i+b\\ \Downarrow\\ y_i=\frac{e^{ax_i+b}}{1+e^{ax_i+b}}\end{gathered}$$



נראה כי כאשר ערכי x גדולים מאוד אזי p שואפת לתיוג 1 וכאשר ערכי x קטנים אזי p שואפת לתיוג -1.

11.3.2 הגדרה פורמלית

כעת במקום להניח כי $\mathbb{E}\left[Y_i \mid X_i = x_i\right]$ לינארית ב־x נניח כי לינארית ב־$g\left(x\right)$, כאשר g תהיה הפונקציה הלוגיסטית, ולכן:

$$logit\left(p_i\right) = \beta_0 + x_i^T\beta,\ \ Y_i \sim Ber\left(p_i\right)$$

$$\Downarrow$$

$$log\left(\frac{\mathbb{P}\left(Y_i = 1|X_i = x_i\right)}{\mathbb{P}\left(Y_i = 0|X_i = x_i\right)}\right) = \beta_0 + x_i^T\beta$$

כדי להגדיר את פונקציית ה־$likelihood$:

$$p_i = \frac{exp\left(\beta_0 + x_i^T\beta\right)}{1 + exp\left(\beta_0 + x_i^T\beta\right)}$$

ואז ה־$likelihood$ ב־β_0, β על כל הדוגמאות היא:

$$L\left(\beta_0, \beta\right) = \prod_{i=1}^{n} p_i^{y_i}\left(1 - p_i\right)^{1-y_i}$$

פונקציית ה־$log - likelihood$ היא:

$$\ell\left(\beta_0, \beta\right) = \sum_{i=1}^{n} y_i\left(\beta_0 + x_i^T\beta\right) - \sum_{i=1}^{n} log\left(1 + exp\left(\beta_0 + x_i^T\beta\right)\right)$$

וכעת על מנת לסווג נרצה למצוא את הסיווג שהכי $likely$. כלומר נחפש את הסיווג אשר $\underset{\beta_0,\beta}{argmax}\,\ell\left(\beta_0, \beta\right)$. בשונה מרגרסיה לינארית אין ביטוי סגור לחישוב זה אבל התוצאה נמצאת בתוך הקמור ($convex$) של β_0, β.

11.3.3 מקרה $multiclassification$

כאשר נרצה לתייג באוסף תיוגים למשל $\mathcal{Y}=\{1,...,K\}$ עבור $k\in\mathbb{N}$ אזי יהיו $\overline{\beta_1},...,\overline{\beta_{K-1}},\beta_1,...,\beta_{K-1}\in\mathbb{R}^d$:

$$\begin{gathered}log\frac{\mathbb{P}(G=1\mid X=x)}{\mathbb{P}(G=K\mid X=x)}=\overline{\beta_1}+\beta_1^Tx\\ \vdots\\ log\frac{\mathbb{P}(G=K-1\mid X=x)}{\mathbb{P}(G=K\mid X=x)}=\overline{\beta_{K-1}}+\beta_{K-1}^Tx\end{gathered}$$

12 $Bayes\,Classifiers$

12.1 $Bayes\,Optimal$

בהינתן התפלגות $\mathcal{D}$ מעל $\mathcal{X}\times\{0,1\}$ פונקציית התיוג האופטימלית, הנקראת גם $Bayes\,Optimal$, הינה

$$f_{\mathcal{D}}\left(x\right)=\begin{cases}1 & \mathbb{P}\left[y=1|x\right]\geq\frac{1}{2}\\ 0 & else\end{cases}$$

טענה 12.1 מסווג ה־$optimal\,Bayes$, $f_{\mathcal{D}}$, הינו האופטימלי. כלומר $L_{\mathcal{D}}\left(f_{\mathcal{D}}\right)\leq L_{\mathcal{D}}\left(g\right)$ $\forall\mathcal{X}\to\{0,1\}$.

הוכחה: יהי $x\in\mathcal{X}$ ותהי p ההסתברות של x להיות מתוייג $1\in\mathcal{Y}$. לפי הגדרה מתקיים כי האיבוד של $h\in\mathcal{H}$:

$$\begin{aligned}L_{\mathcal{D}}\left(h\right) &= \mathbb{E}_{(x,y)\sim D}\left[\mathbb{1}_{h(x)\neq y}\right]\\ &= \mathbb{E}_{x\sim D}\left[\mathbb{E}_{y\sim D_{\mathcal{Y}|x}}\left[\mathbb{1}_{h(x)\neq y}|X=x\right]\right]\end{aligned}$$

בפרט עבור $f_{\mathcal{D}}$ מתקיים כי ההסתברות של $\mathbb{1}_{f_D(x)\neq y}|X=x$ היא:

$$\begin{aligned}\mathbb{P}\left(\mathbb{1}_{f_D(x)\neq y}|X=x\right) &= \mathbb{1}_{p\geq\frac{1}{2}}\cdot\mathbb{P}\left(Y=1|X=x\right)+\mathbb{1}_{p<\frac{1}{2}}\cdot\mathbb{P}\left(Y=0|X=x\right)\\ &= \mathbb{1}_{p\geq\frac{1}{2}}\cdot\left(1-p\right)+\mathbb{1}_{p<\frac{1}{2}}\cdot p\\ &= min\left(p,1-p\right)\end{aligned}$$

מצד שני לכל $h \in \mathcal{H}$ ההסתברות של $\mathbb{1}_{f_D(x)\neq y}|X=x$ היא:

$$\begin{aligned}\mathbb{P}\left(\mathbb{1}_{h(x)\neq y}|X=x\right) &= \mathbb{P}\left(h(x)=0|X=x\right)\cdot\mathbb{P}\left(Y=1|X=x\right)+\mathbb{P}\left(h(x)=1|X=x\right)\cdot\mathbb{P}\left(Y=0|X=x\right)\\ &= \mathbb{P}\left(h(x)=0|X=x\right)\cdot(1-p)+\mathbb{P}\left(h(x)=1|X=x\right)\cdot p\\ &\geq \underbrace{\left(\mathbb{P}\left(h(x)=0|X=x\right)+\mathbb{P}\left(h(x)=1|X=x\right)\right)}_{1} min\left(p,1-p\right)\end{aligned}$$

על כן $\forall h \in \mathcal{H}\ L_{\mathcal{D}}\left(f_D\right)\leq L_{\mathcal{D}}\left(h\right)$ ומכאן האופטימליות כנדרש. ■

12.2 $Naive\,Bayes$

היות ואיננו יודעים את $\mathcal{D}$ איננו יכולים להשתמש ב־$optimal\,Bayes$. קיימות מגוון שיטות המאפשרות לשערך צפיפות של דגימות ובכך לקבל מושג יותר טוב לגבי $\mathcal{D}$. כאשר מימד ה־$feature\,space$ גבוה שיטות אלה יקרות ועל כן נניח "הנחה נאיבית" כי ה־$features$ השונים בלתי תלויים. את כלל ה־$optimal\,Bayes$ נוכל לכתוב בתור $h_{\mathcal{D}}\left(x\right)=\underset{y=\pm1}{argmax}p_{\mathcal{D}}\left(y|x\right)$. ובעזרת נוסחת בייס:

$$h_{\mathcal{D}}\left(x\right)=\underset{y=\pm1}{argmax}\frac{p_{\mathcal{D}}\left(x|y\right)p_{\mathcal{D}}\left(y\right)}{p_{\mathcal{D}}\left(x\right)}=\underset{y=\pm1}{argmax}p_{\mathcal{D}}\left(x|y\right)p_{\mathcal{D}}\left(y\right)$$

וכעת תחת ההנחה כי ה־$features$ בלתי תלויים נקבל כי

$$p\left(x|y\right)=\prod_{i=1}^{m}p_{\mathcal{D}}\left(x_i|y\right)$$

לכן מספיק שנדע את $p_{\mathcal{D}}\left(x|y\right),p_{\mathcal{D}}\left(y\right)$.

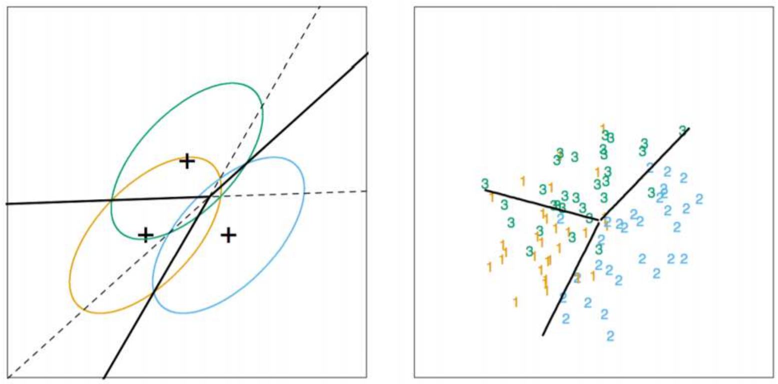
12.3 $Linear\,Discriminant\,Analysis$

כמו בדוגמה של $Naive\,Bayes$ נרצה לתייג ל־$y\in\{0,1\}$ על בסיס $x=\left(x_1,...,x_d\right)\in\mathcal{X}$ כאשר הפעם ההנחה עבור $\mathcal{D}$ היא $p_{\mathcal{D}}\left(x|y\right)=\mathcal{D}\left[x|Y=y\right]=\mathcal{N}\left(\mu_y,\Sigma\right)$ כאשר $\mathcal{N}$ משתנה מקרי מעל $\mathbb{R}^d$ המתפלג נורמלי עם תוחלת μ_y ושונות Σ.

הערה 12.2 פונקציית הצפיפות של משתנה שכזה היא:

$$\mathbb{P}\left(X=x|Y=y\right)=\frac{1}{\sqrt{\left(2\pi\right)^2 det\left(\Sigma\right)}}exp\left(-\frac{1}{2}\left(x-\mu\right)^T\Sigma^{-1}\left(x-\mu\right)\right)$$

כלומר אנחנו מניחים כי ההתפלגות של x בהינתן התפלגות התגיות היא גאוסיאן. על כן כאשר הגאוסיאנים רחוקים אחד מהשני הסיווג פשוט. כאשר הם קרובים ויש חפיפה הסיכוי לטעות גדל. קווי הגובה של פונקציה זו הם אליפסואידים והיות ומטריצת השונות זהה בין תיוגים שונים "אזורי התיוג" לכל תיוג (השטח במרחב שבתוכו נקבל את התיוג) הם אליפסואידים שווים בצורתם כאשר ההבדל הוא מיקום המרכז.



טענה 12.3 בהינתן $\mathcal{D}$ המקיימת את ההנחה לעיל ונסמן $\mathcal{D}(Y=k)=\pi_k$ אזי ה־$bayes\ optimal$ מתקבל על ידי $\underset{y=\pm 1}{argmax}\,\delta_y(x)$ כאשר:

$$\delta_y(x)=x^T\Sigma^{-1}\mu_y-\frac{1}{2}\mu_y^T\Sigma^{-1}\mu+log(\pi_y)$$

הוכחה: תהי $x\in\mathbb{R}^d$ אזי

$$\begin{aligned}
h_{\mathcal{D}}(x) &= \underset{y}{argmax}\,\mathcal{D}(x|y)\,\mathcal{D}(y)\\
&= \underset{y}{argmax}\left(det(2\pi\Sigma)^{-0.5}\,\pi_y exp\left(\tfrac{1}{2}(x-\mu_y)^T\Sigma^{-1}(x-\mu_y)\right)\right)\\
&= \underset{y}{argmax}\left(log(\pi_y)\,+\,\tfrac{1}{2}(x-\mu_y)^T\Sigma^{-1}(x-\mu_y)\right)\\
&= \underset{y}{argmax}\left(log(\pi_y)\,+\,x^T\Sigma^{-1}\mu_y-\tfrac{1}{2}\mu_y^T\Sigma^{-1}\mu_y\right)\\
&= \underset{y}{argmax}\left(log(\pi_y)\,+\,\delta_y(x)\right)
\end{aligned}$$

■

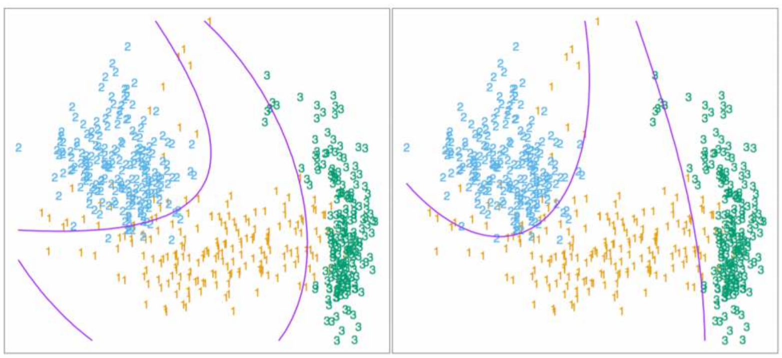
כעת כדי לממש זאת בהינתן $S=\{(x_i,y_i)\}_{i=1}^{m}$ נצטרך להעריך את Σ, π_y, μ_y. נסמן $m_y=\#\{Y=y\}$ (מספר הדוגמאות המתוייגות y עבור $y\in\mathcal{Y}$), $\hat{\mu}_y=\frac{\left(\sum_{i:y_i=y} x_i\right)}{m_y}$, $\pi_y=\frac{m_y}{m}$ ולבסוף

$$\hat{\sum}=\frac{1}{m-2}\sum_{j=\pm1}\sum_{i:y_i=j}\left(\sum\left(x_i-\hat{\mu}_j\right)\left(x_i-\hat{\mu}_j\right)^T\right)$$

נשים לב כי נוכל להציג כעת את δ בתור $\delta_y(x)=a_y+b_y^T x$ ומכאן כי גבול הסיווג הינו לינארי ב־x. כלומר ה־$Bayes\,optimal\,predictor$ במקרה זה הינו $half\,space$.

12.4 $Quadratic\,Discriminant\,Analysis$

ב־LDA הנחנו כי לכל תיוג מטריצת השונות בהתפלגות הנורמלית זהה. כעת נניח כי $\forall y\in\mathcal{Y}$ מטריצת השונות היא Σ_y (מקרה פרטי הוא ה־LDA שבו זהה). במקרה זה קיבלנו את ה־QDA.



13 $Boosting$

$Boosting$ הינה פרדיגמה אלגוריתמית שבאה לעזור עם שתי בעיות. הראשונה היא ה־$bias-complexity\,tradeoff$ והשניה היא ה־$computational\,complexity$ של הלמידה. ראינו תחת ב־$bias-complexity\,tradeoff$ כי שגיאת המסווג ניתנת לפירוק ל־$approximation\,error$ ול־$estimation\,error$ וכי כיצד כאשר מחלקת היפותזות עשירה יותר אז חל שיפור ב־$approximation\,error$ אבל קיימת עליה ב־$estimation\,error$. ב־$boosting$ נרצה לקבל מעט יותר שליטה על $tradeoff$ זה. במקרה של $computational\,complexity$ נראה כי מימוש עיקרון ה־ERM במקרים רבים הינו בעייתי חישובית. שימוש ב־$boosting$ מאפשר לשמר ביצועים של $weak\,learners$ אשר נגדיר בפרק זה.

13.1 $Weak\,Learnability$

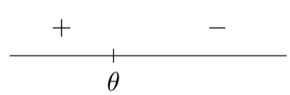
הגדרה 13.1 מחלקת היפותזות $\mathcal{H}$ הינה $(\epsilon,\delta)-weak-learnable$ אם קיים אלגוריתם למידה $\mathcal{A}$ וסט אימון מגודל m כך שלכל הסתברות $\mathcal{D}$ מעל $\mathcal{X}$ ולכל $f\in\mathcal{H}$ מתקיים:

$$\mathbb{P}_{S\sim\mathcal{D}^m}\left(L_{\mathcal{D},f}\left(\mathcal{A}\left(S\right)\right)\leq\epsilon\right)\geq1-\delta$$

הערה 13.2 נשים לב כי בשונה מהגדרת ה־PAC (שנקרא לצורת למידה זו גם למידה חזקה, $strong\,learning$) ב־$weak-learning$ על האלגוריתם לספק את התנאים ל־ϵ,δ ספציפיים ולא $\forall\epsilon,\delta\in(0,1)$.

הערה 13.3 עבור $\delta=0$ למעשה עלינו לספק אלגוריתם שבהסתברות 1 בעל דיוק ϵ ועבור $\delta=1$ תמיד נעמוד בתנאי. כאשר $\epsilon=1$ תמיד נצליח לספק אלגוריתם שבכל הסתברות עומד בתנאי הדיוק וכאשר $\epsilon=0$ למעשה נצטרך אלגוריתם עם איבוד אפס וראינו כי לא קיים כזה. לבסוף עבור $\delta=0,\epsilon=\frac{1}{2}$ מדובר ב־$random$ ולכן נניח לומד חדש בהכרח מקיים $\epsilon<\frac{1}{2}$.

דוגמה־מנחה: תהי $\mathcal{B}=\left\{x\mapsto sign\left(x-\theta\right)\cdot b\,:\,\theta\in\mathbb{R},\,b\left\{\pm1\right\}\right\}$ מחלקת ההיפותזות של $Decision\,Stamps$ (באופן כללי $decision\,stamps$ הינם עץ החלטה בעל רמה אחת $\left\{x\in\mathbb{R}^d\mapsto sign\left(x_j-\theta\right)\cdot b\,:\,\theta\in\mathbb{R},\,b\in\left\{\pm1\right\},\,j\in[d]\right\}$)



כעת נגדיר $\mathcal{X}=\mathbb{R}$ ומחלקת היפותזות $\mathcal{H}$ להיות מחלקת ה־$3-piece\,classifiers$. קל לראות כי לא נוכל ללמוד PAC את המחלקה הזאת בעזרת $\mathcal{B}$ אולם נראה כיצד נוכל ללמוד אותה $weak-learning$ בעזרת $\mathcal{B}$. מחלקה זו היא למעשה סיווג לפי שלושה חלקים:

טענה: קיים $m \in \mathbb{N}$ כך שעבור אלגוריתם הממשש $ERM_{\mathcal{B}}$ האלגוריתם הינו $\left(\frac{5}{12},\frac{1}{2}\right)-weak-learner$ עבור $\mathcal{H}$.

הוכחה: נתחיל מטענת עזר:

למה: לכל $\mathcal{D}$ מעל $\mathcal{X}$ ולכל $f\in\mathcal{H}$ קיימת $h\in\mathcal{B}$ המקיימת $L_{\mathcal{D},f}\left(f\right)\le\frac{1}{3}$.

הוכחה: עבור $x\in\mathcal{X}$ נסמן $\mathbb{P}\left(x\le\theta_1\right)=\mathcal{D}_1,\,\mathbb{P}\left(\theta_1<x\le\theta_2\right)=\mathcal{D}_2,\,\mathbb{P}\left(x>\theta_2\right)=\mathcal{D}_3$. מתקיים $\mathcal{D}_1+\mathcal{D}_2+\mathcal{D}_3=1$ וכן קיים $i\in[3]$ כך ש־$\mathcal{D}_i\le\frac{1}{3}$. נניח בלי הגבלת הכלליות כי $i=1$ אזי אזי נגדיר $h\left(x\right)=\begin{cases}+ & x>\theta_2\\ - & x\le\theta_2\end{cases}$ ולמעשה האלגוריתם טועה רק על $\mathcal{D}_1$ ולכן: $L_{\mathcal{D},f}\left(f\right)=\mathcal{D}_1\le\frac{1}{3}$ כנדרש.

כעת היות ואנו יודעים את ה־$VCdim\left(\mathcal{B}\right)$ ובפרט סופי נוכל, בעזרת המשפט המרכזי של הלמידה הסטטיסטית עבור $\delta=\frac{1}{2}$ נקבל שגיאה של $\frac{1}{3}+\frac{1}{12}=\frac{5}{12}$ אשר נבחר בתור ϵ (נשים לב כי $\frac{1}{2}<\frac{5}{12}$) ועל כן קיים אלגוריתם $\left(\frac{5}{12},\frac{1}{2}\right)-weak-learner$ עבור $\mathcal{H}$ כנדרש.

13.2 $Boosting\,Confidence$

יהי $\mathcal{A}$ אלגוריתם $\left(\epsilon_0,\delta_0\right)-weak-learner$ למחלקת היפותזות $\mathcal{H}$ הדורש m_0 דגימות. נראה כיצד נוכל ללמוד את $\mathcal{H}$ בדיוק של $\epsilon_0+\epsilon$ ובטחון δ לכל $\epsilon,\delta\in\left(0,1\right)$:

1. נריץ את $\mathcal{A}$ על $k=\left\lceil\frac{log(2/\delta)}{log(1/\delta_0)}\right\rceil$ סטים $i.i.d$ כל אחד בגודל m_0 לקבלת $h_1,...,h_k$.
2. ניקח סט ולידציה V כך ש: $\left|V\right|\ge\frac{2log(4k/\delta)}{\epsilon^2}$ ונחזיר $\hat{h}\in\underset{h_i}{argmin}L_V\left(h_i\right)$

טענה 13.4 $\mathbb{P}\left(L_{\mathcal{D}}\left(\hat{h}\right)\le\epsilon_0+\epsilon\right)\ge1-\delta$

הוכחה: תחילה נחסום את ההסתברות לשגיאה גדולה מ־ϵ_0:

$$\begin{aligned}\mathbb{P}\left[\underset{i}{min}L_{\mathcal{D}}\left(h_i\right)>\epsilon_0\right] & = & \mathbb{P}\left[\forall i\;L_{\mathcal{D}}\left(h_i\right)>\epsilon_0\right]\\ & \overset{(i.i.d)}{=} & \prod_{i=1}^{k}\mathbb{P}\left[L_{\mathcal{D}}\left(h_i\right)>\epsilon_0\right]\\ & \overset{weak-learner}{\le} & \left(\delta_0\right)^k\\ & \overset{choosing\,k}{\le} & \delta/2\end{aligned}$$

כלומר $\mathbb{P}\left[\underset{i}{min}L_{\mathcal{D}}\left(h_i\right)\le\epsilon_0\right]\ge1-\frac{\delta}{2}$. היות איננו יודעים את $\mathcal{D}$ אנו משתמשים ב־L_V כמדד שיערוך ל־$L_{\mathcal{D}}$. תהי פונקציית האיבוד ℓ בעלת טווח של $[0,1]$ אזי מאי שיוויון הופדינג נקבל כי בהסתברות לפחות $1-\delta$:

$$\left|L_V\left(h\right)-L_{\mathcal{D}}\left(h\right)\right|\le\sqrt{\frac{log\left(2/\delta\right)}{2m_0}}$$

מתהליך הולידציה אנו מקבלים

$$\mathbb{P}\left[L_{\mathcal{D}}\left(\hat{h}\right)>\underset{i}{min}L_{\mathcal{D}}\left(h_i\right)+\epsilon\right]\le\delta/2$$

כעת באמצעות חסם האיחוד נקבל את מה שהיה להוכיח. ■

תכונות־הפרוצדורה:

1. נדרשנו למספר רב של דגימות: $m_0 \cdot \left\lceil \frac{log(2/\delta)}{log(1/\delta_0)} \right\rceil + \frac{2log(4k/\delta)}{\epsilon^2}$
2. זמן הריצה הינו: $\left\lceil \frac{log(2/\delta)}{log(1/\delta_0)} \right\rceil \cdot \left(O\left(WL\right) + \frac{2log(4k/\delta)}{\epsilon^2} \right)$ כאשר WL הינו האלגוריתם הלומד החלש.
3. הגדלנו את הביטחון אך שילמנו בדיוק: במקום ϵ_0 כעת $\epsilon_0 + \epsilon$.

13.2.1 $Boosting\, Confidence$ בתוחלת

נשים לב כי בהינתן לומד $\mathcal{A}$ המקיים:

$$\underset{S\sim\mathcal{D}^m}{\mathbb{E}}\left[L_{\mathcal{D}}\left(\mathcal{A}\left(S\right)\right)\right] \leq \underset{h\in\mathcal{H}}{min} L_{\mathcal{D}} + \epsilon$$

נסמן $\theta = L_{\mathcal{D}}\left(\mathcal{A}\left(S\right)\right) - \underset{h\in\mathcal{H}}{min} L_{\mathcal{D}}$ אזי $\underset{S\sim\mathcal{D}^m}{\mathbb{E}}\left[\theta\right] \leq \epsilon$. θ הינו משתנה מקרי אי שלילי אזי מאי שיוויון מרקוב:

$$\mathbb{P}\left[\theta \geq 2\epsilon\right] \leq \frac{\mathbb{E}\left[\theta\right]}{2\epsilon} \leq \frac{1}{2}$$

וקיבלנו כי $\mathcal{A}$ הינו $\left(2\epsilon, \frac{1}{2}\right) - weak - learner$ עליו נוכל לעשות $boosting$ ולשפרו.

13.3 $Boosting\ Accuracy$ באמצעות $AdaBoost$

עבור $boosting\, accuracy$ נראה את אלגוריתם $Adaptive\, boosting\, (AdaBoost)$ אשר מספק היפותזה עם $empirical\, risk$ נמוך. בהינתן קלט $S = \{(x_i, y_i)\}_{i=1}^m$ כך ש־$y_i = f\left(x_i\right)$ עבור $f \in \mathcal{H}$ כלשהי, האלגוריתם רץ מספר איטרציות כאשר בכל איטרציה מגדיר התפלגות $\mathcal{D}^{(t)}$ מעל S, מעביר את ההתפלגות ־WL ונותן ל־h_t שחזרה מה־WL משקל פרופורציונלי לשגיאה. באופן אינטואיטיבי השיטה מאלצת את הלומד להתמקד על הדוגמאות ה"קשות" יותר.

AdaBoost

input:
training set $S = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$
weak learner WL
number of rounds T
initialize $\mathbf{D}^{(1)} = (\frac{1}{m}, \ldots, \frac{1}{m})$.
for $t = 1, \ldots, T$:
invoke weak learner $h_t = \mathrm{WL}(\mathbf{D}^{(t)}, S)$
compute $\epsilon_t = \sum_{i=1}^m D_i^{(t)} \mathbb{1}_{[y_i \neq h_t(\mathbf{x}_i)]}$
let $w_t = \frac{1}{2} \log\left(\frac{1}{\epsilon_t} - 1\right)$
update $D_i^{(t+1)} = \frac{D_i^{(t)} \exp(-w_t y_i h_t(\mathbf{x}_i))}{\sum_{j=1}^m D_j^{(t)} \exp(-w_t y_j h_t(\mathbf{x}_j))}$ for all $i = 1, \ldots, m$
output the hypothesis $h_s(\mathbf{x}) = \mathrm{sign}\left(\sum_{t=1}^T w_t h_t(\mathbf{x})\right)$.

כלומר, בהינתן כקלט סט אימון מתוייג, אלגוריתם למידה חלש ו־T מספר איטרציות האלגוריתם:

0) מאתחל התפלגות אחידה $D^{(1)} \sim Unif\left([m]\right)$ מעל S.

1) לכל איטרציה t:

1.1) האלגוריתם קורא ל־WL על סט האימון S וההתפלגות $\mathcal{D}^{(t)}$. ה־WL מחזיר היפותזה מתאימה h_t עבור ההתפלגות וסט האימון. נבחין כי מההגדרה של WL האלגוריתם, עבור ϵ_0, δ_0 מחזיר היפותזה מתאימה לכל התפלגות, ולכן בפרט עבור $\mathcal{D}^{(t)}$.

1.2) נחשב את שגיאת ההכללה $\epsilon_t = L_{\mathcal{D}^{(t)}}\left(h_t\right) = \sum_{i=1}^{m} \mathcal{D}_i^{(t)} \mathbb{1}_{y_i \neq h_t(x_i)}$ - כלומר מספר הדוגמאות שה־WL טעה עליהם עבור התפלגות $\mathcal{D}^{(t)}$.

1.3) נגדיר $w_t = \frac{1}{2} log\left(\frac{1}{\epsilon_t} - 1\right)$ אשר ישמש כמשקולת עבור h_t בבניית ההתפלגות הבאה. נבחין כי בהינתן WL לומד חלש אנו מניחים כי $\epsilon_t < \frac{1}{2}$ (שכן אחרת לא שיפרנו מעל $random$) ועל כן w_t מספר חיובי. המשקולת הינה פרופורציונלית הפוכה לשגיאה - דוגמאות עליהן הצליח המסווג יקבלו משקל נמוך יותר ודוגמאות עליהן טעה המסווג יקבלו משקל גבוה יותר. כך נכריח את המסווג, באיטרציה הבאה, להצליח טוב יותר על הטעויות.

1.4) נבנה התפלגות חדשה $\mathcal{D}^{(t+1)}$ באופן הבא: לכל דגימה $i \in [m]$ נגדיר את התפלגותה על ידי $\mathcal{D}_i^{(t+1)} = \frac{\mathcal{D}_i^{(t)} exp(-w_t y_i h_t(x_i))}{\sum_j \mathcal{D}_j^{(t)} exp(-w_t y_j h_t(x_j))}$. כאשר ההיפותזה מצליחה על דגימה מסוימת אזי $y_i h_t\left(x_i\right) > 0$ ועל כן היות ומוכפל ב־$-w_t$ נקבל כי משקל הדגימה יקטן. כאשר ההיפותזה טועה על דגימה מסוימת אזי $y_i h_t\left(x_i\right) < 0$ ולכן כאשר מוכפל ב־$-w_t$ נקבל כי משקל הדגימה יגדל.

2) לבסוף נחזיר את סימן הסיווג (התיוג ± 1) על בסיס סכום משוקלל של כל ההיפותזות כפול משקלן.

טענה 13.5 שגיאת היפותזה h_t ביחס להתפלגות $\mathcal{D}^{(t+1)}$ היא $\frac{1}{2}$

הוכחה:

$$\sum_{i=1}^{m}\mathcal{D}_i^{(t+1)}\mathbb{1}_{y_i\neq h_t(x_i)} = \frac{\sum_{i=1}^{m}\mathcal{D}_i^{(t)}e^{-w_ty_ih_t(x_i)}\mathbb{1}_{y_i\neq h_t(x_i)}}{\sum_{j=1}^{m}\mathcal{D}_j^{(t)}e^{-w_ty_jh_t(x_j)}} = \frac{e^{w_t}\epsilon_t}{e^{w_t}\epsilon_t+e^{-w_t}(1-\epsilon_t)}$$

$$= \frac{\epsilon_t}{\epsilon_t+e^{-2w_t}(1-\epsilon_t)} = \frac{\epsilon_t}{\epsilon_t+\frac{\epsilon_t}{1-\epsilon_t}(1-\epsilon_t)} = \frac{1}{2}$$

■

טענה 13.6 יהי WL אלגוריתם $\left(\frac{1}{2}-\gamma,\delta\right)-weak-learner$ אזי $\mathbb{P}\left(L_S\left(h_s\right)\leq exp\left(-2\gamma^2T\right)\right)\geq 1-\delta T$

הערה 13.7 לכל $\epsilon>0$ ו־$\gamma\in\left(0,\frac{1}{2}\right)$ אם $T\geq\frac{log(1/\epsilon)}{w\gamma^2}$ אזי $AdaBoost$ יחזיר היפותזה h_s עם $L_S\left(h_s\right)\leq\epsilon$.

הערה 13.8 זמן הריצה של האלגוריתם הינו $T\left(O\left(m\right)+O\left(WL\right)\right)$ - בכל איטרציה $t\in[T]$ האלגוריתם מריץ את ה־WL על S ($O\left(WL\right)$), מחשב את ϵ_t על סמך S ($O\left(m\right)$), מחשב את w_t ($O\left(1\right)$) ומעדכן את ההתפלגות ($O\left(m\right)$).

דוגמה־מנחה כפי שראינו אלגוריתם ה־$AdaBoost$ למעשה מחזיר הרכבה ($ensamble/composition$) של מסווגים חלשים. תהי מחלקת ההיפותזות $\mathcal{B}=\left\{x\mapsto sign\left(x-\theta\right)\cdot b\,:\,\theta\in\mathbb{R},\,b\left\{\pm1\right\}\right\}$ של $Decision\,Stamps$ (שראינו בתחילת הפרק) אשר WL יחזיר היפותזה ממחלקה זו. אזי אלגוריתם $AdaBoost$ מחזיר היפותזה מהמחלקה הבאה:

$$L\left(\mathcal{B},T\right)=\left\{x\mapsto sign\left(\sum_{t=1}^{T}w_th_t\left(x\right)\right)\mid w\in\mathbb{R}^T,\,\forall t\;h_t\in\mathcal{B}\right\}$$

כלומר כל $h\in L\left(\mathcal{B},T\right)$ מכיל T היפותזות בסיסיות ב־$\mathcal{B}$ הממושקלות לפי וקטור $w\in\mathbb{R}^T$. הפרדיקציה של h על דגימה x למעשה קודם יוצרת וקטור

$$\psi\left(x\right)=\left(h_1\left(x\right),...,h_T\left(x\right)\right)\in\mathbb{R}^T$$

אשר ה־$halfspace$ שמגדיר w מפריד אותו. על כן נוכל להגדיר את מחלקת ההיפותזות בתור:

$$L\left(\mathcal{B},T\right)=\left\{x\mapsto sign\left(\left\langle w,\psi\left(x\right)\right\rangle\right)\mid w\in\mathbb{R}^T,\;\left|\left|w\right|\right|_0\leq T\right\},\;\left|\left|w\right|\right|_0=\left|\left\{i:w_i\neq0\right\}\right|$$

13.3.1 $Bias - complexity\, tradeoff$

טענה 13.9 $VCdim\left(L\left(\mathcal{B},T\right)\right)\leq\tilde{O}\left(T\cdot VCdim\left(\mathcal{B}\right)\right)$

מסקנה 13.10 $AdaBoost$ יחזיר היפותזה h_s אשר בהסתברות של לפחות $1-\delta$: ($\tilde{O}$ - עד כדי פקטורים לוגריתמים)

$$L_{\mathcal{D}}\left(h_s\right)\leq L_S\left(h_s\right)+\tilde{O}\left(\sqrt{\frac{T\cdot VCdim\left(\mathcal{B}\right)+log\left(1/\delta\right)}{m}}\right)$$

נובע מכך:

- ה־$estimation\, error$ של $AdaBoost$ גדל לינארית ב־T.
- ה־$empirical\, risk$ של $AdaBoost$ קטן עם T.
- באופן כללי ניתן להשתמש ב־T להקטנת ה־$approximation\, error$ של $L\left(\mathcal{B},T\right)$ ועל כן T יכול לשמש כמנוף ב־$tradeoff$ בין $bias$ ו־$complexity$.

להסבר יותר מעמיק ניתן לקרוא בספר הקורס בעמוד 137.

14 $Model\,Selection\,and\,Validation$

כאשר ראינו בפרק הקודם את אלגוריתם $AdaBoost$ (וכפי שנראה בדוגמאות נוספות תחת $regulization, decision\,trees, Soft-SVM$) ראינו כיצד פרמטר T שולט ב־$bias-complexity\,tradeoff$. כיצד נבחר T זה מבין $\{\mathcal{H}_T\,:\,T\in\mathbb{N}\}$?

14.1 $Validation\,and\,k-Cross\,Validation$

בתהליך ה־$Validation$ נרצה להצליח לשערך כמה טוב המסווג שלנו. שיטה אחת, כפי שעשינו ב־$AdaBoost$ ($Hold\,out\,set$), היא להחזיק סט דגימות נוסף לולידציה על המסווגים שקיבלנו ובעזרתו לבחור את האופטימלי.

אחת השיטות הנפוצות כדי להעריך $prediction\,error$ הינה $Cross-Validation$. באופן טבעי, אם היה לנו מספיק $data$ היינו שומרים חלק בצד בתור $validation\,set$ כדי לבדוק את טיב המודל שנבחר. היות ו־$data$ איננו בלתי מוגבל נרצה להשתמש במה שיש גם ל־$train$ וגם ל־$validation$. עבור $K\in\mathbb{N}$ כאשר נדבר על $K-fold\,cross-validation$ נחלק את הדגימות ל־K חלקים שווים באופן מקרי. נאמן מודל על סמך $K-1$ מקטעים ונבדוק על המקטע ה־K. כל פעם נבחר את המקטע בדיקה (ולידציה) להיות מקטע אחר ולבסוף נמצע את כל התוצאות.

1	2	3	4	5
Train	Train	Validation	Train	Train

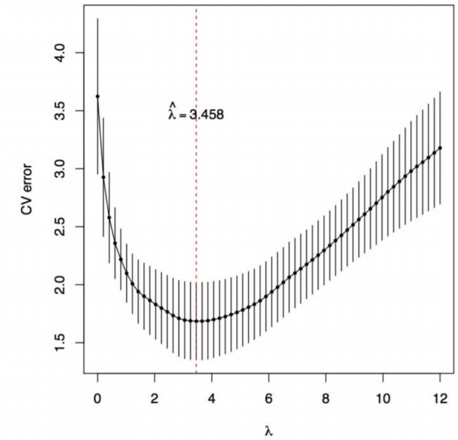
באופן יותר פורמלי תהי $k:[N]\to[K]$ הממפה דגימה ל־$partition$ באופן מקרי. נסמן $\hat{f}^{-k}(x)$ ההיפוזתה שנבחרה על ה־$data$ ללא ה־$partition$ ה־k. אזי שיערוך השגיאה באמצעות ה־$cross\,validation$ הינו:

$$CV\left(\hat{f}\right)=\frac{1}{N}\sum_{i=1}^{N}L\left(y_i,\hat{f}^{-k(i)}\left(x_i\right)\right)$$

אם אנו עוסקים במודל בעל פרמטר כיול α כמו למשל פרמטר הרגולריזציה בעצי החלטה או $Soft\,SVM$ או כדוגמת פרמטר k ב־KNN (ראו בהמשך) אז אנו למעשה עוסקים במשפחה של מחלקות היפותזות $\mathcal{H}_\alpha$. נסמן $\hat{f}^{-k}(x,\alpha)$ המודל ה־α שנבחר עבור הפרמטר רגולריזציה α שנבחר על ה־$data$ ללא ה־$partition$ה־k. אזי שיערוך השגיאה באמצעות ה־$cross\,validation$ הינו:

$$CV\left(\hat{f},\alpha\right)=\frac{1}{N}\sum_{i=1}^{N}L\left(y_i,\hat{f}^{-k(i)}\left(x_i,\alpha\right)\right)$$

במקרה שכזה נוכל לשערך את טעות ה־CV כתלות בפרמטר α (λ בגרף) עבור ערכים שונים ולקבל עקומה שתעזור לנו לקבוע מה ה־λ המתאים לבעיה.



Pseudocode

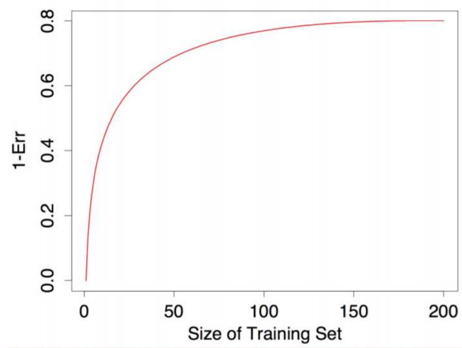
k-Fold Cross Validation for Model Selection

input:
  training set $S = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$
  set of parameter values Θ
  learning algorithm A
  integer k
partition S into $S_1, S_2, \ldots, S_k$
foreach $\theta \in \Theta$
  for $i = 1 \ldots k$
    $h_{i,\theta} = A(S \setminus S_i; \theta)$
  $\text{error}(\theta) = \frac{1}{k} \sum_{i=1}^{k} L_{S_i}(h_{i,\theta})$
output
  $\theta^\star = \text{argmin}_\theta \left[\text{error}(\theta)\right]$
  $h_{\theta^\star} = A(S; \theta^\star)$

14.1.1 בחירת ערך ה־k עבור $k-fold$

כאשר $k=1$ כל ה־$data$ הזמין משמש לאימון ולמעשה אין $cross\,validation$. כאשר $k=2$ (המכונה גם $split-sample$) רק מחצית המידע משמש ל־$train$ ולמעשה השגיאה שנקבל עם CV תהיה גדולה מבלי. באופן כללי עבור k קטנים אנחנו עלולים להתאמן על $datasets$ קטנים מדי ונקבל כי ה־$CV\,error$ הוא $biased\,upwards$ - משערכים שגיאה גדולה ממה שנוכל להשיג בפועל. עבור ערכי k גדולים ההבדל בין קבוצות המדגם באימונים השונים נמוך. על כן תהיה קורלציה גבוהה יותר בין תוצאות אימונים שונים ולמעשה נגדיל את ה־$variance$. ככלל אצבע, על מספיק $data$, $k=5,10$ נותנים ביצועים טובים (ניתן לקרוא את $Breiman\,and\,Spector$, 2012 או $Kohavi$, 1995).

היות ובחירת k משפיעה גם על גודל סט האימון נרצה לראות עבור הדומיין הספציפי מהו גודל סט האימון הנחוץ. נוכל לצייר עקומה שתעיד על ההצלחה כתלות בגודל ה־$trainingset$. נראה כי עבור $N = 200$ או $N = 150$ ההצלחה (או 1 פחות הטעות) כמעט לא משנה ועל כן נוכל לבחור k שבו נאפשר פחות דגימות בסט האימון, נגדיל את השונות בין הסטים, נקטין את ה־$variance$ וזאת מבלי לאבד כמעט באיכות המסווג שיבחר.



14.1.2 איך לא בוחרים k

נניח כי אנו חוקרים בעיית למידה מסויימת ומנסים למצוא מודל מתאים. אנחנו מחפשים סט $predictors$ טובים במחלקה שלנו (ונניח כי המחלקה עשירה). לאחר שמצאנו סט שכזה המראה קורלציה טובה עם התיוגים נמצא מסווג על בסיסן. כעת נשתמש ב־$cross\,validation$ כדי להעריך את ערך פרמטר הכיול ולהעריך את ה־$prediction\,error$ הסופי.

מקרה זה מתאר מקרה שבו הכנסנו אופטימיות למודל. סט ההיפותזות שהתבססנו עליהן נבחר על סמך ביצועיהן על כל ה־$data$ ועל כן התוצאות שקיבלנו בסופו של תהליך (ה־$CV\ error$) הוא $biased$ לטובת היפותזות אלה ולמעשה אנחנו מנבאים טעות נמוכה מדי. הדרך הנכונה לפעול היא:

1. נפצל את הדגימות ל־K קבוצות ($folds$) באופן רנדומלי.
2. לכל $k = 1, ..., K$:

(א) נמצא תת קבוצה של $predictors$ בעלות קורלציה טובה בין תוצאותיהן לתיוגים. בשלב זה נשתמש בכל ה־$samples$ למעט אלה ב־$fold$ ה־k.

(ב) נמצא מסווג טוב על בסיס תת קבוצה זו בהסתכל על כל ה־$samples$ למעט אלה ב־$fold$ ה־k.

(ג) נבדוק את טיב המסווג על הדגימות ב־$fold$ ה־k.

14.2 $Train - Validation - Test\ split$

בדרך כלל אלגוריתמים הכוללים $Validation$ הם מהצורה:

1. $Training\, Stage$ - מציאת ההיפותזה האופטימלית מכל מודל:

$$\begin{aligned} For\, each\, \theta \in \Theta: \quad & Draw\, new\, S \\ & h_\theta = \mathcal{A}\left(S, \theta\right) \end{aligned}$$

2. $Validation\, Stage$ - מציאת היפותזה אופטימלית מעל כל סט המודלים:

$$\begin{aligned} & Sample\, new\, set\, V \\ & Choose\, h^* = h_{\theta^*}: \quad \theta^* = \underset{\theta \in \Theta}{argmin} L_V\left(h_\theta\right) \end{aligned}$$

3. $Test\, Stage$ - שיערוך טיב ההיפותזה שנבחרה:

$$\begin{aligned} & Sample\, new\, set\, T \\ & Estimate\, error\, by \quad L_T\left(h^*\right) \end{aligned}$$

15 $Regularization$

כאשר אנו באים לענות על בעיית למידה וקיים לנו דומיין $\mathcal{X}$ על $feature\, space$ כלשהו אנו מעוניינים למצוא היפותזה הפותרת טוב את הבעיה. ראינו ב־$bias - complexity\, tradeoff$ שנרצה למצוא היפותזה אשר מידת המורכבות שלה לא נמוכה מדי ולא גבוהה מדי. בפרק זה נראה כלים העוזרים לנו למצוא היפותזה שכזו. בנוסף בבעיות למידה רבות אנו נתקלים במצבים שבהם מספר ה־$feautres$ גבוה מאוד ($d \sim m$ או אפילו $d > m$). כאשר מספר ה־$features$ רב, היכולת ליצור מודלים מתאימים טוב יתר ל־$data$ (ומכאן גם מורכבים מאוד) היא רבה ועל כן נרצה להיות ערים וזהירים מכך. נרצה גם להשתמש רק ב־$features$ שעוזרים לנבא הכי טוב את התיוגים מתוך כלל ה־$features$, ולהתעלם מ־$features$ שמידת התרומה שלהם לפרדיקציה היא נמוכה. לזאת קוראים $feature\, selection$ ונתעסק גם בזאת מעט בפרק זה.

15.1 עיקרון $RLM - Regularized\, Loss\, Minimization$

ראינו כי בעיקרון ה־ERM נחזיר היפותזה אשר מביאה למינימום את ה־$empirical\, risk$. במקרים רבים שימוש ב־ERM פשוטו כמשמעו יביא לקבלת היפותזה מורכבת אשר עונה טוב מאוד על סט האימון (מצליחה לתפוס את הנקודות ולמעשה "הולכת אחרי הרעש") אך זו תצליח פחות טוב על סט הבדיקה. במקרה זה נסבול מ־$variance$ גבוה ו־$overfitting$. נרצה להתאים את כללי הלמידה באופן כזה שיגבילו את מורכבות ההיפותזה המוחזרת ($regulaized$).

הגדרה 15.1 פונקציית רגולריזציה ($regulaizer$) הינה מיפוי $\mathcal{R} : \mathcal{H} \to \mathbb{R}$ כאשר עיקרון ה־RLM מחזיר היפותזה המביאה למינימום את $\sum\limits_{i=1}^{m} \left(L\left(y_i, h\left(x_i\right)\right) + \lambda \mathcal{R}\left(h\right)\right)$ עבור $\lambda \geq 0$.

λ משמש בתור ה־$complexity\, parameter$. באופן אינטואיטיבי מורכבות ההיפותזה נמדדת על ידי הערך שמוחזר מ־$\mathcal{R}$ והאלגוריתם אם כך יחפש איזון ($tradeoff$) באמצעות λ בין $empirical\, risk$ נמוך לבין מודלים פשוטים. עבור $\lambda = 0$ נקבל את עיקרון ה־ERM ועבור λ גדול, היות ונחפש מינימום, נקבל רק מודלים פשוטים. כעת נראה מספר מימושים אפשריים של עיקרון זה ונעמוד על ההבדלים ביניהם. במבט כללי ההבדל ביניהם הוא השימוש בנורמות שונות ($||\cdot||_0,\ ||\cdot||_1,\ ||\cdot||_2$ כאשר בפועל "נורמת 0" איננה באמת נורמה) בפונקציית הרגולריזציה. נתבסס על רגרסיה לינארית כדוגמה מנחה על מנת להבין את המימושים השונים.

15.1.1 $Best\, Subset\, Regression$

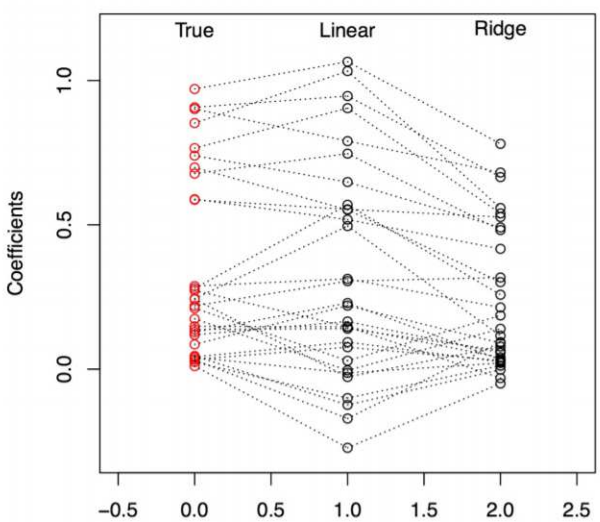
נגדיר $\mathcal{R}\left(w\right) = \#\left\{i\ :\ w_i \neq 0\right\}$ ונראה לפונקציה זו ה־$sparsity$ של w. נסמן גם ב־$||w||_0$. כלומר באופן אינטואיטיבי $regulaizer$ זה יבקש לצמצם את מספר ה־$features$ אשר המודל משתמש בהם: נחפש מינימום גם על גורם זה ־ כלומר היפותזה המכילה פחות מקדמים שהם לא אפס ־ כלומר היפותזה שבה יש יותר $features$ שלא בשימוש. במקרה של רגרסיה לינארית שבה $\mathcal{H} = \left\{x \mapsto \langle x, w \rangle\right\}$ נקבל שנרצה מינימום של:

$$\left|\left|y - X^T w\right|\right|_2^2 + \lambda \left|\left|w\right|\right|_0$$

למרות שנראה כי הגדרה זו היה בדיוק מה שאנו מחפשים כדרך לומר כמה פשוט/מורכב נרצה את המודל שלנו, נשים לב שלמעשה ניתן לעשות רדוקציה מבעיה זו לבעיית מציאת תת קבוצה אופטימלית אשר הינה בעיה $NP-hard$. לכן לא קיים אלגוריתם יעיל לפתרון בעיה זו. בכל זאת, למספר $d\,features$ נמוך (בערך עד כדי 40) קיימות ספריות שנותנות מענה.

15.1.2 $Ridge\,Regression$

ה־$Ridge\,Regression$מכווץ את המקדמים של הוקטור ברגרסיה על ידי אכיפת עלות לגודלם. בגרף הבא נתבונן בערכי המקדמים ל־$features$ השונים על $data$ שיצרנו ואנו יודעים מהם הערכים האמיתיים של המקדמים. נראה כי כאשר ביצענו את הרגרסיה הלינארית כל $feature$ קיבל כעת מקדם כלשהו כאשר גם הפיזור ($variance$) גדולה יותר וגם כאשר נתבונן ב־$features$ שהמקדמים שלהם הם 0 או כמעט 0 נראה כי חלקם קיבלו ערכים שונים מכך אשר למעשה פוגע בפרדיקציה. כאשר נשווה זאת עם תוצאת ה־$ridge\,regression$ נראה כי קיבלנו הן פיזור צפוף יותר ($variance$ נמוך יותר) וגם $features$ עם מקדמים סביב ה־0 נשארו עם מקדמים שכאלה ־ כלומר כמו במקרה האמת הם אינם משמשים לצורך הפרדיקציה (ונוכל לשקול להורידם).



כאשר ניקח את ERM ועבור $\mathcal{R}$ נבחר את $||\cdot||_2^2$ (לעתים מסמנים את הנורמות גם בתור $||\cdot||$ $for\,\ell_2\,norm$) נקבל את $Tikhonov\,regulaization$: $\mathcal{A}(S)=\underset{w}{argmin}\left(L_S(w)+\lambda||w||_2^2\right)$. כשנשליך זאת על רגרסיה לינארית עם $squared\,loss$ נקבל את ה־$Ridge\,regression$:

$$\left|\left|y-X^Tw\right|\right|_2^2+\lambda\,||w||_2^2$$

כשננסה לפתור בעיית מינימיזציה זו נקבל את המשוואות הנורמליות הבאות:

$$Xy=\left(XX^T+\lambda I\right)w\iff\left(XX^T+\lambda I\right)Xy=w$$

אשר $\left(XX^T+\lambda I\right)$ סימטרית חיובית עם ע"ע אי שליליים. על כן מטריצה הפיכה (כזכור XX^T איננה בהכרח הפיכה) - בדומה לרגרסיה לינארית קיים פתרון סגור. כפי שהגדרנו $\Sigma^\dagger$ נגדיר:

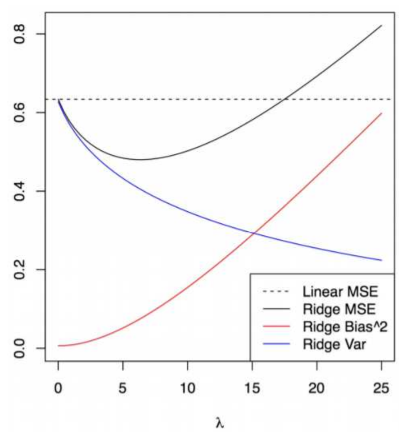
$$\Sigma_{i,i}^{\lambda}=\frac{\sigma_i}{\sigma_i^2+\lambda}$$

ועבור פירוק ה-SVD של $X=U\Sigma V^T$ הפתרון הינו:

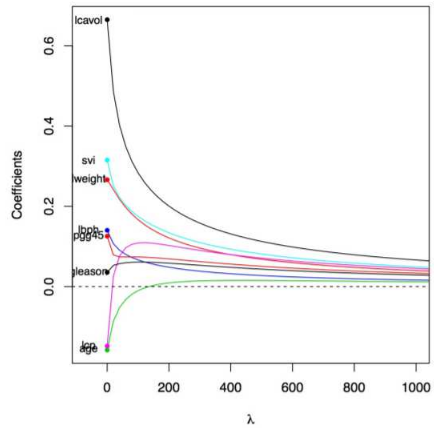
$$\hat{w}_\lambda=U\Sigma^\lambda V^T y$$

$Bias\,variance\,tradeoff$

נסתכל על ה-$bias\,variance\,tradeoff$ כתלות ב-λ. עבור $\lambda=0$ קיבלנו למעשה את ה-ERM ומודל מורכב יותר. עבור $\lambda=25$ קיבלנו בהכרח מודל פשוט שכן מודלים מורכבים נקנסו ועל כן לא יבחרו. באופן כללי נראה כי ה-$variance$ קטן בעוד שה-$bias$ גדל. היות וככל ש-λ גדל אנו מכריחים מודלים פשוטים יותר היכולת שלהם ללכת אחרי הרעש יורדת והתוצאות בהתאם. נראה כי אכן ישנם ערכי λ עבורם ה-$ridge\,regression$ מצליח להוריד $variance$ בקצב מהיר יותר מעליית ה-$bias$ - אלה הם ערכי ה-λ שנרצה לבחור.



$Regulaization\,Path$ כאשר נקבע בעיה ספציפית ונבדוק את שינוי המקדמים של ה-$features$ כתלות ב-λ נקבל את הגרף הבא. גרף זה מכונה ה-$regulization\,path$. נראה כי על אף ש-$Ridge$ משפר את MSE וכאן כאשר $\lambda\to\infty$ המקדמים שואפים ל-0 הוא איננו נותן לנו אינטואיציה טובה איזה $features$ נוכל להוריד ולא להשתמש בהם.

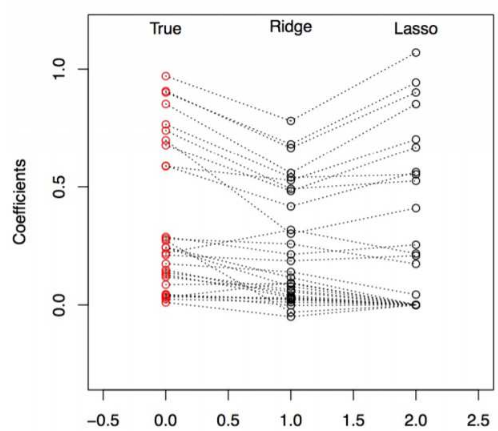


15.1.3 L_1 $Lasso$

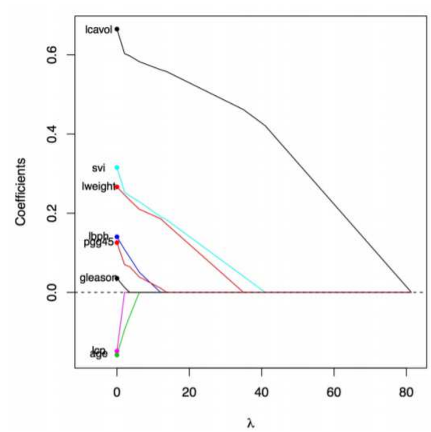
בדומה ל־$ridge$ ה־$lasso$ הינה פונקציית כיווץ מקדמים כאשר ה־$penalty$ הינו לפי $||\cdot||_1$. כלומר $\mathcal{R}(w)=\sum_i |w_i|$:

$$\underset{w}{argmin} \left|\left|y-X^Tw\right|\right|_2^2+\lambda\,||w||_1$$

השימוש ב־$lasso$ מוביל לכך שההיפותזה המוחזרת אינם לינאריים ב־y_i ואין נוסחה סגורה למציאת ההיפותזה המינימלית. RLM עם $Lasso$ (אשר הינו בעיית למידה קמורה ־ ראו בהמשך הסיכום) הינו $quadratic\,programming$ ולכן קיימים אלגוריתמים יעילים לפתרון בעיית אופטימיזציה. בדומה ל־$ridge$ כאשר $\lambda=0$ נקבל את כלל ה־ERM וככל ש־λ גדל נחפש היפותזות אשר $||w||_1$ קטן אשר מוביל למעשה לפתרון שבו יותר ויותר מהמקדמים הם 0. מכאן כי ה־$constraint$ ש־$lasso$ גורם הוא למעשה סוג של $subset\,selection$. כאשר נבדוק את פיזור המקדמים עבור $lasso$, בדומה למה שעשינו עבור $ridge$ נראה כי בשונה מ־$ridge$ עבור $features$ שבקושי תרמו לפרדיקציה (מקדמים קרובים ל־0) ה־$lasso$ נתן מקדם 0. כלומר האלגוריתם החליט כך ש־$features$ אלא אינם רלוונטיים לפרדיקציה.



$Regulaization\,Path$ כאשר נבדוק את ה־$regulaization\,path$ עבור $lasso$ נקבל את התמונה הבאה. נראה כי בשונה מהשינוי האקספוננציאלי שקיבלו המקדמים ב־$ridge$, במקרה הזה השינוי הינו כמעט לינארי ב־λ. כלומר נוכל אפוא להחליט מה יהיה גודל ה$subset$ של ה־$features$ שנרצה (ובכך להגביל את ה־$complexity$ של ההיפותזה הנבחרת) ואז בהתאם לבחור את ה־λ. לדוגמה עבור $\lambda = 20$ בדוגמה הזאת נקבל כי מחצית ה־$features$ קיבלו מקדם 0 ולא ישחקו תפקיד בפרדיקציה.



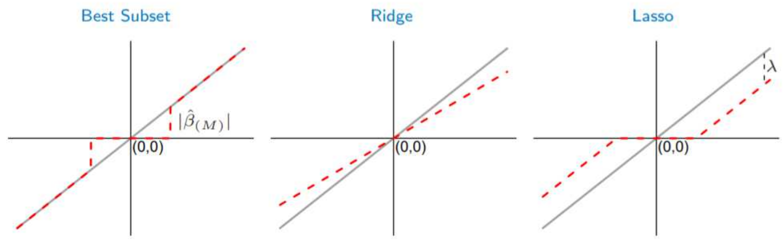
15.2 השוואה בין פונקציות הרגולריזציה

תחילה נתייחס ל־$intercept$. הן ב־$ridge$ והן ב־$lasso$ לא מבצעים רגולריזציה ל־$intercept$. כלומר בנורמה, בפעולת הסכימה עובדים רק על הקואורדינטות 1 עד d ולא מ־0. ספציפית למקרה של $lasso$ במקרים רבים נתאים את המטריצה X כך ששורות המטריצה היו בעלות תוחלת אפס . כעת ננסה להבין ממה נובעים ההבדלים בין $ridge$ ו־$lasso$.

15.2.1 $Hard/Soft\,Thresholding$

במקרה שבו X אורתונורמלית הפונקציות השונות משרות טרנספורמציה פשוטה על ה־$Least\,Squares$. נשים לב כי $ridge$ לא שולח אף ערכים ישירות לאפס בעוד ש־$best\,subset$ ו־$lasso$ כן. בנוסף $Lasso$ מכווץ מקדמים שאינם נשלחים ל־0 בעוד ש־$best\,subset$ לא משתנים המקדמים של ה־$features$ שלא נשלחו ל־0.

Estimator	Formula
Best subset (size M)	$\hat{\beta}_j \cdot I(\lvert\hat{\beta}_j\rvert \geq \lvert\hat{\beta}_{(M)}\rvert)$
Ridge	$\hat{\beta}_j/(1+\lambda)$
Lasso	$\text{sign}(\hat{\beta}_j)(\lvert\hat{\beta}_j\rvert - \lambda)_+$



ה־$best\, subset$ נותן ערך 0 לכל מקדם שקטן מ־M המקדמים הראשונים. זה נקרא $hard\, thresholding$.
$ridge\, regression$ מכווץ את המקדמים באופן פרופורציונאלי ו־$lasso$ מכווץ את המקדמים בפקטור λ ומאפס ערכים קטנים מ־0. שני אלה נקראים $soft\, thresholding$. כלומר עבור $\lambda \geq 0$ ו־$x \in \mathbb{R}$:

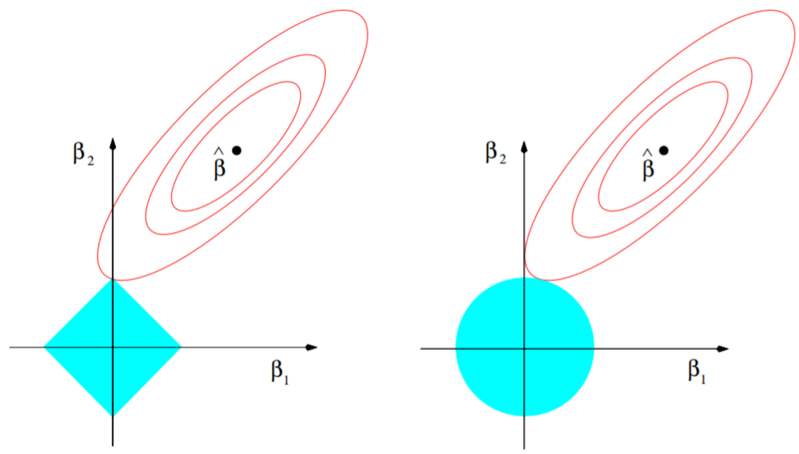
$$\eta_\lambda^{soft}(x) = \begin{cases} x-\lambda & x \geq \lambda \\ 0 & -\lambda > x > \lambda \\ x+\lambda & -\lambda \geq x \end{cases}, \quad \eta_\lambda^{hard}(x) = x \cdot \mathbb{1}_{|x| \geq \lambda}$$

ונקבל:

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

15.2.2 כדורי יחידה של הנורמות ו־$sparsity$

בדוגמה בה אנו עוסקים, פונקציית האיבוד $Least\, square$, עבור רגרסיה לינארית, הינה תבנית ריבועית ומכאן כי היא קמורה וקווי הגובה שלה אליפסואידים. הנקודות $\hat{\beta}$ מייצגות את נקודת המינימום של הפונקציות. כאשר נסתכל על הבעיה ב־$\mathbb{R}^2$ ונמקם זאת ליד כדורי היחידות של הנורמות השונות ($||\cdot||_1$ ־ משמאל, $||\cdot||_2$ ־ מימין) נקבל את נקודות החיתוך בין קווי הגובה השונים לבין כדורי היחידה. במקרה של $||\cdot||_1$ (שימוש ב־$lasso$) הנקודות השפיציות הן נקודות שבהן המקדם של $feature$ מתאפס. על כן עיקרון ה־RLM יחזיר היפותזות אשר נמצאות בנקודות מפגש אלה. הקנס על שימוש בעוד $features$ ידחוף לבחירת היפותזה על הקודקוד אשר שם יש פחות $features$ עם מקדמים שונים מאפס. כאשר נדמיין זאת ב־$\mathbb{R}^d$ נקבל "כדור" יחידה עם קודקודים רבים ־ כאשר בכל קודקוד כמות ושילוב שונה של $features$ שהמקדמים שלהם הם 0.



15.3 $Lasso$ **עבור** $Logistic\,Regression$

כשם ש־$\ell_1\,regulaization$ עובד טוב עבור רגרסיה לינארית הוא גם עובד טוב עבור רגרסיה לוגיסטית. כאשר ניקח את הרגרסיה לוגיסטית עם ה־$log-likelyhood$

$$\ell\left(\beta_0,\beta\right)=\sum_{i=1}^{n}y_i\left(\beta_0+x_i^T\beta\right)-\sum_{i=1}^{n}log\left(1+exp\left(\beta_0+x_i^T\beta\right)\right)$$

ננרמל ונוסיף ל־$lasso$ $negative\,log-likelyhood$ נקבל את הפונקציית $loss$ הבאה למצוא לה מינימום:

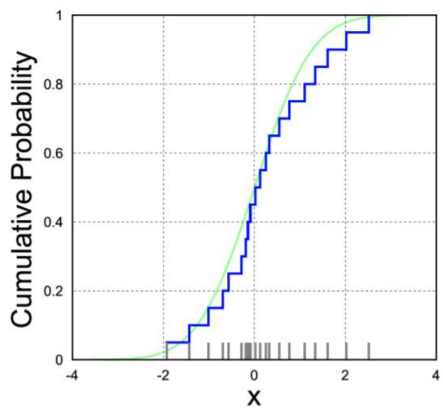
$$-\left[\frac{1}{m}\sum_{i=1}^{m}y_i\left(\beta_0+x_i^T\beta\right)-log\left(1+exp\left(\beta_0+x_i^T\beta\right)\right)\right]+\lambda\left|\left|\beta\right|\right|_1$$

נשתמש ב־$cross\,validation$ למצוא λ טוב (והיות ובעיה קמורה קיימים אלגוריתמים גנריים יעילים). למעשה באימפליקציה זו נקבל כי ℓ_1 יוצר לנו $sparse$־יות של וקטור המקדמים β ־ עבור λ גדול נקבל יותר מקדמים 0 ־ יותר $sparse$ ־ פחות $features$ שהינם חלק מהפרדיקציה. יוצא שמסווג זה הינו עם $misclassification\,rates$ טובים ובוחר רק את ה־$feature\,subset$ שרלוונטי לבעיה.

16 $Bootstrapping\ and\ Bagging$

16.1 $Bootstrapping$

$Bootstrapping$ הינו כלי כללי לשיערוך דיוק סטטיסטי. נניח כי בידינו $z_1,...,z_n \overset{iid}{\sim} \mathcal{D}$ דגימות מעל התפלגות $\mathcal{D}$ לא ידועה. נשים לב כי בהינתן n הדגימות הללו יש בידינו התפלגות אמפירית $\hat{\mathcal{D}}_n$ של דגימות אלו. כעת נניח כי בידינו איזושהי סטטיסטיקה (פונקציה על ה־$data$) $\hat{\theta}$ וברצוננו לשערך את $\hat{\theta}\left(X\right)$ כאשר $X\sim\mathcal{D}$ (כלומר מתוך ההתפלגות הכללית אשר איננה ידועה לנו). אם נניח כי $\hat{\mathcal{D}}_n\approx\mathcal{D}$ אזי עבור דגימה כרצוננו מהתפלגות של $\hat{\theta}\left(X^*\right)$ כאשר $X\sim\hat{\mathcal{D}}_n$ נקבל כי $\hat{\theta}\left(X^*\right)\approx\hat{\theta}\left(X\right)$. כעת נוכל לשלוף B דגימות באופן $monte-carlo$ מ־$\hat{\theta}\left(X^*\right)$ ונקבל כי הדגימות דומות להתפלגות המקורית (והלא ידועה)



באופן ספציפי עבור בעיות למידה, עבור $B\in\mathbb{N}$ שנחליט, נבצע את התהליך הבא: נשלוף באופן מקרי, עם חזרות, m דגימות מתוך ה־$dataset$ שנתון לנו. נסמן דגימות אלה בתור $\left(x_i^{(b)},y_i^{(b)}\right)_{i=1}^m$ ונכנה בשם $bootstrap\ samples$. עבור הסט ה־b נאמן מודל $h^{(b)}$. נוכל כעת למצע על סמך כל אוסף המסווגים הללו ולשערך את דיוקם טוב יותר.

16.2 $Bagging$

כאשר ראינו את $bootstrapping$ המטרה הייתה לשערך את הדיוק שבפרדיקציה. כעת נשתמש ב־$bootstrapping$ כדי לשפר את הפרדיקציה עצמה. יהי סט האימון $S=\left\{(x_i,y_i)\right\}_{i=1}^N$ ונניח לשם הדוגמה כי ביצענו רגרסיה לינארית ובידינו היפותזה $\hat{f}\left(x\right)$. ב־$Bagging$ (או $Bootstrap\ aggregation$) נמצע את הפרדיקציה על קבוצת $bootstrap\ samples$. כלומר:

- עבור $B\in\mathbb{N}$ שנבחר, לכל $b\in[B]$:
  - נשלוף m דגימות באופן מקרי עם חזרות $Z^{*b}=\left(x_i^{(b)},y_i^{(b)}\right)_{i=1}^m$.
  - נאמן מודל חדש על הסט Z^{*b} שיספק את הפרדיקציה $\hat{f}^{*b}\left(x\right)$.

- ה־$estimation$ עבור דגימה חדשה יהיה $ensamble$ של המסווגים שיצרנו: (נוכל להחליט על שיטות מיצוע אחרות כמו $majority\,vote$)

$$\hat{f}_{bag}\left(x\right)=\frac{1}{B}\sum_{b=1}^{B}\hat{f}^{*b}\left(x\right)$$

באופן כללי כאשר $B\rightarrow\infty$ נקבל כי $\hat{f}_{bag}\left(x\right)\rightarrow\hat{f}\left(x\right)$. כלומר ה־$bagged\,estimation$ שואפת לפרדיקציה אילו ידענו את $\mathcal{D}$ (בספר הקורס Elements of Statistical Learning עמוד 282 יש הסבר תחת אילו תנאים הדבר איננו מתקיים). בשיטה זו נקטין את ה־$variance$ מבלי להגדיל את ה־$bias$.

$Bagging\,vs,\,Boosting$

	Bagging	Boosting
sampling training data	uniform	non-uniform
Can we train in parallel?	yes	yes
weights over the classifiers	uniform	non-uniform
bias-variance trade-off	reduce variance	reduce bias

17 $Feature\,Selection$

בשלב זה ראינו כיצד מתנהג ה־$bias-complexity\,tradeoff$, כיצד לבחור מודל בהתאם לפרמטר כיול (בהתאם ולמודל קיים) ושיטות שונות לשיפור יכולות הלימוד בהינתן $data$ מוגבל. כאשר מימד הקלט (הן $feature\,space$ והן כמות דגימות) גבוה קשה לנו להתמודד עם זאת. בהמשך נראה שני אלגוריתמים, SVM ו־$Kernels$ שעוזרים להתמודד עם הבעיה. כרגע נדון כיצד נוכל לפשט בעיה זאת על ידי צמצום חלק מה־$features$. נשים לב כי ניתן להשיג מטרה זו באופן חלקי גם באמצעות שימוש ב־$Lasso$ תחת $Regulaization$ אולם זמן העבודה (אימון אך גם עיבוד ה־$data$ ואולי גם גורמים אחרים כמו עליות להשגתו) עדין קיימות. על כן נרצה למצוא איזה $features$ נוכל להורידם באופן כללי. בספר ישנן מספר גישות לכך (עמוד 358) אך בקורס כיסינו רק את $Filter$.

בשיטה זו, שאולי מהפשוטה שבשיטות, אנו בוחנים $features$ באופן בלתי תלוי מהשאר ומספקים "ניקוד" ל־$feature$. ערך זה יעיד על מידת הקשר בין התיוג של הדגימה לערך ה־$feature$. כמובן שגם לאופן חישוב הניקוד יתכנו פונקציות רבות. בחירה טבעית אחת תהיה ה־$Empirical\,Square\,Loss$ של $predictor$ לינארי:

$$score\left(j\right)=\min_{a,b\in\mathbb{R}}\frac{1}{m}\sum_{i=1}^{m}\left(a\left(x_{i}\right)_{j}+b-y_{i}\right)^{2}$$

כאשר $\{(x_i,y_i)\}_{i=1}^m \subset \mathbb{R}^d \times \mathbb{R}$ ו־$j \in [d]$ מייצג $feature$ ספציפי. פונקציה זו תכווין ל־$features$ בעלי ניקוד נמוך. נסמן $\bar{v} = \frac{1}{m}\sum v_i$ ו־$\bar{y} = \frac{1}{m}\sum y_i$ אזי עבור $v^j = \left((x_1)_j, ..., (x_m)_j\right)^T$, $1_m = (1,...,1)^T \in \mathbb{R}^m$ נקבל:

$$\begin{aligned} score(j) &= \min_{a,b\in\mathbb{R}} \frac{1}{m}\left|\left|av^j + b\cdot 1_m - y\right|\right|_2^2 \\ &= \min_{a,b\in\mathbb{R}} \frac{1}{m}\left|\left|a\left(v^j - \bar{v^j}\cdot 1_m\right) + b\cdot 1_m - (y - \bar{y}\cdot 1_m)\right|\right|_2^2 \end{aligned}$$

וכדי למצוא a,b מתאימים נגזור לפי b ונשווה לאפס. נקבל כי $b=0$. כעת נחשב עבור a ונקבל כי

$$a = \frac{\left\langle v^j - \bar{v^j}\cdot 1_m, y - \bar{y}\cdot 1_m\right\rangle}{\left|\left|v^j - \bar{v^j}\cdot 1_m\right|\right|_2^2}$$

כאשר נציב בפונקציה נקבל כי

$$score(j) = \frac{1}{m}\left(\left|\left|y - 1_m\cdot\bar{y}\right|\right|_2^2 - \frac{\left\langle v^j - \bar{v^j}\cdot 1_m, y - \bar{y}\cdot 1_m\right\rangle^2}{\left|\left|v^j - \bar{v^j}\cdot 1_m\right|\right|_2^2}\right)$$

כעת נוכל על בסיס פונקציה זו להגדיר פונקציות $score$ חדשות:

$$score_2(j) = -\frac{\left\langle v^j - \bar{v^j}\cdot 1_m, y - \bar{y}\cdot 1_m\right\rangle^2}{\left|\left|v^j - \bar{v^j}\cdot 1_m\right|\right|_2^2 \cdot \left|\left|y - 1_m\cdot\bar{y}\right|\right|_2^2}$$

$$\Downarrow$$

$$score_3(j) = \left|\frac{\left\langle v^j - \bar{v^j}\cdot 1_m, y - \bar{y}\cdot 1_m\right\rangle}{\left|\left|v^j - \bar{v^j}\cdot 1_m\right|\right|_2 \cdot \left|\left|y - 1_m\cdot\bar{y}\right|\right|_2}\right| = \left|\frac{\frac{1}{m}\left\langle v^j - \bar{v^j}\cdot 1_m, y - \bar{y}\cdot 1_m\right\rangle}{\sqrt{\frac{1}{m}\left|\left|v^j - \bar{v^j}\cdot 1_m\right|\right|_2^2}\cdot\sqrt{\frac{1}{m}\left|\left|y - 1_m\cdot\bar{y}\right|\right|_2^2}}\right|$$

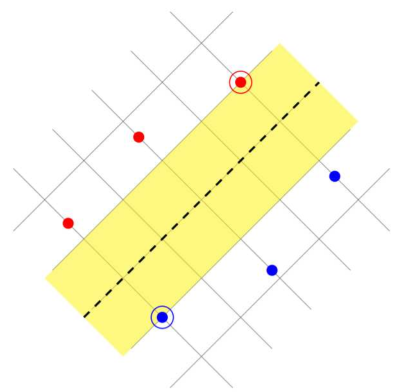
$$\Downarrow$$

$$score_3(j) = \left|\frac{\frac{1}{m}\sum_{i=1}^m\left((x_i)_j - \bar{v^j}\right)(y_i - \bar{u})}{\sqrt{\frac{1}{m}\sum_{i=1}^m\left((x_i)_j - \bar{v^j}\right)^2}\cdot\sqrt{\frac{1}{m}\sum_{i=1}^m(y_i - \bar{u})^2}}\right|$$

כאשר הפונקציה האחרונה למעשה מדרגת הפוך מהראשונה ־ $features$ עם ציון גבוה הם משמעותיים יותר. למעשה מה שקיבלנו הוא פונקציית $Pearson's\ correlation\ coefficient$ אשר מודדת קורלציה לינארית בין שני משתנים. ערכיה של פונקציה זו הם בין -1 ל־1 כאשר 1 מציין קורלציה חיובית מושלמת ו־-1 קורלציה שלילית מושלמת.

18 SVM

כאשר נרצה ללמד מפרידים לינאריים במימד $feature\,space$ גבוה נראה כי המימד מגדיל הן את ה־$sample\,complexity$ והן את ה־$computational\,complexity$. שיטת $SVM - support\,vector\,machine$ באה להתמודד עם אתגר ה־$sample\,complexity$ על ידי מציאת על־מישור מפריד בעל המרחק ($margin$) הגדול ביותר מהדגימות בסט האימון



הגדרה 18.1 יהי $S = \{(x_i, y_i)\}_{i=1}^m$ סט אימון של m דוגמאות מעל $\mathbb{R}^d$ ותיוגים $y_i \in \{\pm 1\}$. נאמר כי S ניתן להפרדה לינארית ($linearly\,separable$) אם קיים על־מישור (w, b) כך שמתקיים $\forall i \in [m]\ sign\left(\langle w, x_i\rangle + b\right) = y_i$. תנאי שקול: $\forall i \in [m]\ y_i\left(\langle w, x_i\rangle + b\right) > 0$

נבחין כי כל על־מישור הניתן להפרדה לינארית מקיים את עיקרון ה־ERM ביחס לפונקציית האיבוד $0-1$ ועל כן נרצה להיות מסוגלים לבחור חצי מרחב ספציפי מתוך הקבוצה.

18.1 $Margin\,and\,Hard - SVM$

טענה 18.2 בהינתן על־מישור (w, b) כך ש־$||w|| = 1$ ונקודה x במרחב, אזי המרחב בין הנקודה לעל־מישור הוא $|\langle w, x\rangle + b|$

הוכחה: נשים לב כי המרחק בין הנקודה לעל־מישור מוגדר בתור $min\left\{||x - v|| \ : \ \langle w, v\rangle + b = 0\right\}$. על כן אם נבחר $v = x - \left(\langle w, x\rangle + b\right) w$ נקבל:

$$\begin{aligned}
\langle w, v\rangle + b &= \langle w, (x - (\langle w, x\rangle + b)\, w)\rangle + b &&= \langle w, x\rangle - \langle w, (\langle w, x\rangle + b)\, w\rangle + b \\
&= \langle w, x\rangle - (\langle w, x\rangle + b)\,\langle w, w\rangle + b &&= \langle w, x\rangle - (\langle w, x\rangle + b)\,||w||^2 + b \\
&= \langle w, x\rangle - \langle w, x\rangle - b + b = 0
\end{aligned}$$

ובנוסף $|\langle w, x\rangle + b| = ||w|| \cdot |\langle w, x\rangle + b| = ||x - v||$. כל כן המרחק הינו לכל היותר $|\langle w, x\rangle + b|$. כעת בהינתן נקודה u

על העל־מישור אזי $\langle w,u\rangle + b = 0$ ונקבל:

$$\begin{aligned} ||x-u||^2 &= ||x-v+v-u||^2 \\ &= ||x-v||^2 + ||v-u||^2 + 2\langle x-v, v-u\rangle \\ &\geq ||x-v||^2 + 2\langle x-v, v-u\rangle \\ &= ||x-v||^2 + 2\langle x-x+(\langle w,x\rangle + b)w, v-u\rangle \\ &= ||x-v||^2 + 2(\langle w,x\rangle + b)(\langle w,v\rangle - \langle w,u\rangle) \\ &= ||x-v||^2 \end{aligned}$$

כל כן המרחק בין x ל־u הוא לפחות כמו המרחק בין x ל־v אשר ראינו כי הוא: $|\langle w,x\rangle + b|$ כנדרש. ■

הגדרה 18.3 המרחק בין הנקודה הקרובה ביותר ב־S לעל־מישור המפריד $\{u : \langle w,u\rangle + b = 0\}$ נקרא ה־$margin$ ושווה ל: $\underset{i\in[m]}{min}|\langle w,x_i\rangle + b|$.

הגדרה 18.4 כלל למידה $Hard - SVM$ הינו:

$$\underset{(w,b)\,:\,||w||=1}{argmax}\ \underset{i\in[m]}{min}|\langle w,x_i\rangle + b| \ \ s.t\ \ \forall i,\ y_i(\langle w,x_i\rangle + b) > 0$$

הגדרה 18.5 בעיית אופטימיזציה תקרא $Quadratic\,Programming$ אם היא ניתנת לכתיבה בצורה הבאה:

$$\underset{w\in\mathbb{R}^n}{min}\left(\frac{1}{2}w^TQw + c^Tw\right)\ \ s.t.\ Aw \leq d$$

כאשר $Q\in\mathbb{R}^{n\times n},\ A\in\mathbb{R}^{m\times n},\ c\in\mathbb{R}^n,\ d\in\mathbb{R}^m$.

כעת, בדומה לפתרון בעיות של תכנון לינארי אשר קיימים אלגוריתמים יעילים לפתרונן, נראה כיצד להציג את הבעיה כבעיית תכנון $quadratic$ אשר גם לה קיימים אלגוריתמים יעילים לפתרונה.

טענה 18.6 יהיו v^*, c^* פתרונות אופטימליים של:

$$(1)\ \underset{(w,b)}{argmin}\,||w||^2\ \ s.t.\ \forall i\ y_i(\langle w,x_i\rangle + b) \geq 1$$

נסמן $\gamma = \frac{1}{||v^*||}$ אזי $w^* = \gamma v^*,\ b^* = \gamma c^*$ פתרונות אופטימליים לכלל הלמידה של $Hard - SVM$ הלינארי.

הוכחה: תחילה נראה כי w^*, b^* הינם אכן פתרונות של שלל הלמידה $Hard-SVM$ (כבעיית תכנון לינארי) ולאחר מכן נוכיח אופטימליות. יהי w פתרון חוקי לכלל הלמידה $Hard-SVM$ אזי w אכן מגדיר על־מישור מפריד ומתקיים כי $|\langle w, x_i\rangle + b| = y_i\left(\langle w, x_i\rangle + b\right)$ בנוסף, בהיותו w פתרון חוקי, תנאי החיוביות מתקיים ולכן נוכל לייצג את הכלל בתור:

$$(2)\ \underset{||w||=1,b}{argmax}\underset{i}{min}\, y_i\left(\langle w, x_i\rangle + b\right)$$

כעת נתבונן בפתרונות v^*, c^*. בהיותם אופטימליים עבור (1) מתקיים כי:

$$\underset{i}{min}\, y_i\left(\langle v^*, x_i\rangle + c^*\right) = 1$$

שכן אחרת נוכל לחלק את v^* במספר חיובי ולקבל פתרון חוקי קטן מהמינימלי. נשים לב כי w^*, b^* פתרון חוקי עבור (2) שכן $1 = \left|\left|\frac{1}{||v^*||}v^*\right|\right| = ||\gamma v^*|| = ||w^*||$ ומתקיים:

$$\underset{i}{min}\, y_i\left(\langle \gamma v^*, x_i\rangle + \gamma c^*\right) = \gamma\left(\underset{i}{min}\, y_i\left(\langle v^*, x_i\rangle + c^*\right)\right) = \gamma$$

(נבחין כי γ איננו תלוי ב־i ולכן נוכל להוציאו מהמכפלה הפנימית, להוציא כגורם משותף ולבסוף להוציא מחוץ ל־min).
כעת נראה כי זהו פתרון אופטימלי עבור (2). יהיו $w_2 \neq v^*,\ b_2 \neq c^*$ המקיימים

$$\underset{i}{min}\, y_i\left(\langle w_2, x_i\rangle + b_2\right) = \delta > \gamma$$

כעת נוכל להגדיר פתרון $\frac{w_2}{\delta}, \frac{b_2}{\delta}$ אשר חוקי עבור (1) ועבורו מתקיים כי $\frac{1}{\gamma} < \frac{1}{\delta} = \left|\left|\frac{w_2}{\delta}\right|\right|$ וקיבלנו פתרון טוב יותר מ־v^*, c^* בסתירה לאופטימליות שלהם. ■

18.1.1 המקרה ההומוגני

נוכל לעשות רדוקציה מהמקרה של למידת חצאי־מרחב לא הומוגניים ללמידת חצאי־מרחב הומוגניים. נעשה זאת על ידי הוספת $feature$ נוסף לכל דגימה x_i. כאשר נעשה זאת נעלה את המימד ל־$\mathbb{R}^{d+1}$. נשים לב כי במקרה הקודם ה־$bias$ b לא נכלל בהגבלה של $||w||^2 = 1$ ואם נלמד את הבעיה ב־$\mathbb{R}^{d+1}$ למעשה אנו מכניסים רגולריזציה לגודל ה־$bias$.

18.1.2 $Sample\, Complexity$ של $Hard-SVM$

נזכר כי ה־$VCdim$ של חצאי מרחב ב־$\mathbb{R}^d$ הינו $d+1$. מכאן כי ה־$sample\, complexity$ של למידת החצאי־מרחב גדלה עם המימד. מהמשפט המרכזי של הלמידה הסטטיסטית אנחנו מקבלים כי אם מספר הדגימות קטן משמעותית מ־$\frac{d}{\epsilon}$ אז לא קיים אלגוריתם שיכול ללמוד את החצי־מרחב בדיוק ϵ.

כדי להתמודד עם הבעיה נצטרך להוסיף הנחה עבור התפלגות המידע. נגדיר הנחה כי המידע ניתן להפרדה לינארית עם $margin$ של γ ונראה שבתנאים אלה ה־$sample\, complexity$ חסום מלמעלה על ידי $\frac{1}{\gamma^2}$.

הגדרה 18.7 יהי $\mathcal{D}$ התפלגות מעל $\mathbb{R}^d \times \{\pm 1\}$. נאמר כי $\mathcal{D}$ ניתנת להפרדה לינארית עם $(\gamma, \rho) - margin$ אם קיים (w^*, b^*) כך שמתקיים: $||w^*|| = 1$, $\mathbb{P}_{(x,y)\sim\mathcal{D}}\left(y\left(\langle w^*, x\rangle + b^*\right) \geq \gamma\right) = 1$ ו־$||x|| \leq \rho$.

באופן דומה נגדיר כי $\mathcal{D}$ ניתנת להפרדה לינארית עם $(\gamma,\rho)-margin$ עם חצי־מרחב הומוגני אם קיים $(w^*,0)$ כך שהתנאים הקודמים מתקיימים.

משפט 18.8 תהי $\mathcal{D}$ התפלגות מעל $\mathbb{R}^d\times\{\pm1\}$ הניתנת להפרדה לינארית עם $(\gamma,\rho)-margin$. מתקיים כי בהסתברות של לפחות $1-\delta$ עבור סט אימון מגודל m שגיאת ה־$0-1$ של $Hard-SVM$ תהיה לכל היותר:

$$\sqrt{\frac{4\left(\frac{\rho}{\gamma}\right)^2}{m}}+\sqrt{\frac{2log\left(\frac{2}{\delta}\right)}{m}}$$

18.2 $Soft-SVM$

על מנת להשתמש ב־$Hard-SVM$ עלינו להניח כי המידע ניתן להפרדה לינארית. כדי להחליש הנחה זו נעשה שימוש ב־$Soft-SVM$. אם כך נגדיר $\xi_1,...,\xi_m\geq 0$ אשר:

$$\underbrace{y_i\left(\langle w,x_i\rangle+b\right)\geq 1}_{Hard-SVM}\Rightarrow\underbrace{y_i\left(\langle w,x_i\rangle+b\right)\geq 1-\xi_i}_{Soft-SVM}$$

אם כך ב־$Soft-SVM$ קיים $tradeoff$ בין ה־$margin$ שמחפשים לבין ה־$violation\,of\,constraints$ שמבוטא ב־ξ_i. $tradeoff$ זה מגולם בפרמטר λ.

הגדרה 18.9 כלל למידה $Soft-SVM$ הינו:

$$\underset{w,b,\xi}{min}\left(\lambda\left|\left|w\right|\right|^2+\frac{1}{m}\sum_{i=1}^{m}\xi_i\right)\ \ s.t\ \forall i,\ y_i\left(\langle w,x_i\rangle+b\right)\geq 1-\xi_i\ \xi_i\geq 0$$

הגדרה 18.10 בהינתן (w,b), סט אימון S ופונקציית איבוד $hinge$ ממוצעת $L_S^{hinge}\left((w,b)\right)$ (כאשר $\ell^{hinge}\left((w,b),(x,y)\right)=max\left\{0,1-y\left(\langle w,x\rangle+b\right)\right\}$) נוכל להגדיר

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

אשר הינה בעיה שקולה לכלל ה־$Soft-SVM$. נשים לב כי תחת הגדרה זו אנו רואים כי $Soft-SVM$ נופל תחת הפרדיגמה של $Regulaized\,loss\,minimization\,(RLM)$ וכי $\mathcal{R}(w)=\lambda\left|\left|w\right|\right|^2$.

באופן כללי ה־$VCdim$ של למידת $halfspace$ תלוי במימד d ועל כן ה־$sample\,complexity$ גדל יחד עם גדילת d. בניגוד לזה, שימוש ב־SVM מגדיר חסם ל־$sample\,complexity$ על ידי $\left(\frac{\rho}{\gamma}\right)^2$ אשר יעיל מאוד במצבים שבהם $d>>\left(\frac{\rho}{\gamma}\right)^2$.

18.3 $SVM\ vs.\ Boosting$

	SVM	Boosting
output classier	$\text{sign}(\mathbf{w}\cdot\psi(\mathbf{x}))$	$\text{sign}(\mathbf{w}\cdot\psi(\mathbf{x}))$
requirement from ψ	$\langle\psi(\mathbf{x}),\psi(\mathbf{x}')\rangle$ easy to compute	base classifier is WL
coordinates of ψ	$\mathbb{R}$	$\{-1,1\}$
norm on weights	$\|\|\mathbf{w}\|\|_2$	$\|\|\mathbf{w}\|\|_0$

19 $Kernels$

ראינו אפוא כי חיפוש מפרידים לינאריים בממדים גבוהים מספק יכולת ביטוי רבה לאלגוריתמים לחיפוש חצאי־מרחב. שיטה זו עולה לנו הן ב־$sample\ complexity$ והן ב־$computational\ complexity$. SVM נתן מענה לבעיית ה־$sample\ complexity$ באמצעות $margin$. על מנת להתמודד עם $computational\ complexity$ נשתמש בשיטת $kernels$.

19.1 $Embedding\ into\ Feature\ Space$

נראה כי מפרידים לינאריים הינם בעלי יכולת פתרון בעיות מוגבלת שכן יתכן ובמימד הנוכחי לא ניתן להפריד את המידע בצורה טובה. על כן נוכל תחילה למפות את הדגימות ל־$feature\ space$ אחר ובו לחפש מפריד לינארי טוב. הפרדיגמה הכללית למיפוי זה:

1. עבור $domain$ $\mathcal{X}$ ומשימת אימון נבחר פונקציית מיפוי $\psi:\mathcal{X}\to\mathcal{F}$.
2. בהינתן סט אימון $S=\{(x_i,y_i)\}_{i=1}^m$ ניצור $\hat{S}=\{(\psi(x_i),y_i)\}_{i=1}^m$
3. נאמן $predictor$ לינארי h מעל $\hat{S}$.
4. עבור דגימת בדיקה x ניתן חיזוי $h(\psi(x))$.

ההצלחה של שיטה זו תלויה בהצלחה בבחירת פונקציית מיפוי כך שבתמונתה הדוגמאות יהיו $linear\ seperable$. דוגמה אחת לפונקציית מיפוי גנרית הינה $polynomial\ mapping$. אם במפרידים לינאריים דיברנו על $x\mapsto\langle w,x\rangle$ אזי נוכל להרחיב ל־ $x\mapsto p(x)$ כאשר p הינו $multivariate\ polynomial$ מדרגה k.

הגדרה 19.1 $multivariate\ polynomial$ מדרגה k מ־$\mathbb{R}^n$ ל־$\mathbb{R}$ הינו ביטוי מהצורה:

$$p(x)=\sum_{J\in[n]^r:r\le k}\left(w_J\prod_{i=1}^{r}x_{J_i}\right)$$

$Kernel\,Trick$ 19.2

השימוש בפונקציית מיפוי זו אשר מעלה את המימד של ה־$data$ טומנת בתוכה גם בעיות: מציאת ψ איננה בהכרח משימה פשוטה, חישוב $\langle w,\psi\left(x\right)\rangle$ עלול להיות יקר וה־$sample\,complexity$ שהוא ה־$VCdim\left(half\,spaces\,in\,\mathbb{R}^{k}\right)=k+1$ מצריך מספר רב של דגימות אימון.

את בעיית ה־$sample\,complexity$ ניתן לפתור כפי שראינו ב־SVM. על מנת להתמודד עם העלות חישוב ($computational\,complexity$) נראה את ה־$Kernel\,Trick$.

שיטה זו הינה יעילה כאשר חישוב המיפוי $\psi\left(x\right)$ יקר אבל המכפלה הפנימית $\langle\psi\left(x_{i}\right),\psi\left(x_{j}\right)\rangle$ זול.

משפט 19.2 ($representers\,theorem$) תהי $\psi:\mathcal{X}\to\mathcal{F}$ וכלל הלמידה

$$w^{*}\in\underset{w}{argmin}f\left(\langle w,\psi\left(x_{1}\right)\rangle,...,\langle w,\psi\left(x_{m}\right)\rangle\right)+R\left(||w||\right)$$

כאשר $f:\mathbb{R}^{m}\to\mathbb{R}$ ו־$R:\mathbb{R}_{+}\to\mathbb{R}$ פונקציה מונוטונית עולה, אזי קיים פתרון w^{*} ווקטור $\alpha\in\mathbb{R}^{m}$ כך ש: $w^{*}=\sum_{i=1}^{m}\alpha_{i}\psi\left(x_{i}\right)$

הערה 19.3 כל כללי ה־SVM מהפרק הקודם (ואופטימיזציות נוספות שניתן לראות בספר הקורס) מקיימות את כלל הלמידה הנ"ל. בנוסף R לדוגמה עבור $Soft-SVM$ הינה למעשה λa^{2} או עבור $Hard-SVM$ הינה למעשה a^{2}.

הוכחה: יהי w^{*} פתרון אופטימלי לכלל הלמידה. נוכל לכתבו בתור

$$w^{*}=\sum_{i=1}^{m}\alpha_{i}\psi\left(x_{i}\right)+u$$

כאשר $\langle u,\psi\left(x_{i}\right)\rangle=0$ $(*)$ לכל i. נגדיר $w=w^{*}-u$. נראה כי מתקיים $||w^{*}||=||w||^{2}+||u||^{2}\ \Leftarrow\ ||w||\leq||w^{*}||$. ממונוטוניות R מתקיים כי $R\left(||w||\right)\leq R\left(||w^{*}||\right)$. בנוסף

$$\forall i\ \ \langle w,\psi\left(x_{i}\right)\rangle=\langle w^{*}-u,\psi\left(x_{i}\right)\rangle=\langle w^{*},\psi\left(x_{i}\right)\rangle-\langle u,\psi\left(x_{i}\right)\rangle\overset{(*)}{=}\langle w^{*},\psi\left(x_{i}\right)\rangle$$
$$\Downarrow$$
$$f\left(\langle w,\psi\left(x_{1}\right)\rangle,...,\langle w,\psi\left(x_{m}\right)\rangle\right)\ =\ f\left(\langle w^{*},\psi\left(x_{1}\right)\rangle,...,\langle w^{*},\psi\left(x_{m}\right)\rangle\right)$$

ולמעשה הראנו כי $w=\sum_{i=1}^{m}\alpha_{i}\psi\left(x_{i}\right)$ פתרון אופטימלי כנדרש. ■

על בסיס משפט זה נשים לב כי לכל i מתקיים:

$$\langle w,\psi\left(x_{i}\right)\rangle=\left\langle \sum_{j}\alpha_{j}\psi\left(x_{j}\right),\psi\left(x_{i}\right)\right\rangle =\sum_{j=1}^{m}\alpha_{j}\left\langle \psi\left(x_{j}\right),\psi\left(x_{i}\right)\right\rangle$$
$$||w||^{2}=\left\langle \sum_{j}\alpha_{j}\psi\left(x_{j}\right),\sum_{j}\alpha_{j}\psi\left(x_{j}\right)\right\rangle =\sum_{i,j=1}^{m}\alpha_{i}\alpha_{j}\left\langle \psi\left(x_{i}\right),\psi\left(x_{j}\right)\right\rangle$$

אזי תהי $K\left(x,x'\right)=\left\langle \psi\left(x\right),\psi\left(x'\right)\right\rangle$ פונקציית $kernel$ נוכל לכתוב את כלל הלמידה בתור

$$\underset{a\in\mathbb{R}^{m}}{min}f\left(\sum_{j=1}^{m}\alpha_{j}\left\langle \psi\left(x_{j}\right),\psi\left(x_{1}\right)\right\rangle ,...,\sum_{j=1}^{m}\alpha_{j}\left\langle \psi\left(x_{j}\right),\psi\left(x_{m}\right)\right\rangle \right)+R\left(||w||\right)$$

$$\Downarrow$$

$$\underset{a\in\mathbb{R}^{m}}{min}f\left(\sum_{j=1}^{m}\alpha_{j}K\left(x_{j},x_{1}\right),...,\sum_{j=1}^{m}\alpha_{j}K\left(x_{j},x_{m}\right)\right)+R\left(\sqrt{\sum_{i,j=1}^{m}\alpha_{i}\alpha_{j}K\left(x_{i},x_{j}\right)}\right)$$

וביטוי זה איננו תלוי כלל ב־w וכל שעלינו לדעת זה כיצד לחשב את המטריצה $G\in M^{m\times m}\left(\mathbb{R}\right)$ כך ש: $G_{i,j}=K\left(x_{i},x_{j}\right)$ ולמעשה מתקיים כי:

$$\underbrace{\begin{gathered}\left\langle w,\psi\left(x_{i}\right)\right\rangle =\left(G\alpha\right)_{i}\\ ||w||^{2}=\alpha^{T}G\alpha \end{gathered}}$$

$$\Downarrow$$

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

היתרון בעבודה עם $kernels$ על פני מציאת w אופטימלי ב־$feature\,space$ הוא שבמקרים שמימד ה־$feature\,space$ מאוד גבוהה מימוש וחישוב $kernel$ יכול להיות פשוט (וזול חישובית) בהרבה.

דוגמה: כאשר ניקח את כלל הלמידה של $Soft-SVM$ ונבטאו באופן זה נקבל:

$$\underset{w,b,\xi}{min}\left(\lambda||w||^{2}+\frac{1}{m}\sum_{i=1}^{m}\xi_{i}\right)\ \ s.t\ \forall i,\ y_{i}\left(\left\langle w,x_{i}\right\rangle +b\right)\geq1-\xi_{i}\ \xi_{i}\geq0$$

$$\Updownarrow$$

$$\underset{a\in\mathbb{R}^{m}}{min}\left(\lambda\alpha^{T}G\alpha+\frac{1}{m}\sum_{i=1}^{m}max\left\{ 0,1-y_{i}\left(G\alpha\right)_{i}\right\} \right)$$

משוואה זו נוכל לכתוב בעיית $quadratic\,programming$ עבורה קיים אלגוריתם יעיל. ברגע שנלמד את המקדמים α חישוב דגימה חדשה מתבצע על ידי:

$$\left\langle w,\psi\left(x\right)\right\rangle =\sum_{j=1}^{m}\alpha_{j}\left\langle \psi\left(x_{j}\right),\psi\left(x\right)\right\rangle =\sum_{j=1}^{m}\alpha_{j}K\left(x_{j},x\right)$$

באופן כללי אפיון של פונקציות $kernel$ מתבצע באמצעות המשפט הבא:

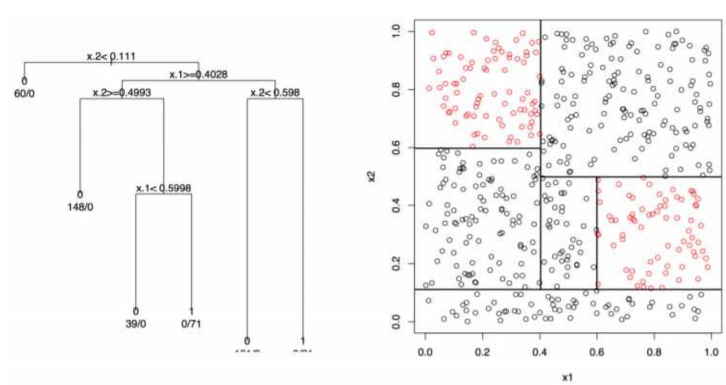
הגדרה 19.4 נאמר כי מטריצה סימטרית $M \in M^{n \times n}(\mathbb{R})$ הינה $positive\, semidefinite$ אם ורק אם לכל וקטור $z \in \mathbb{R}^n$ מתקיים כי $z^T M z \geq 0$.

משפט 19.5 פונקציה סימטרית $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ מממשת מכפלה פנימית במרחב $Hilbert$ אם ורק אם היא $semidefinite$ $positive$. כלומר אם לכל $x_1, ..., x_m$ מטריצת גראם $G_{i,j} = K(x_i, x_j)$ הינה מטריצה $positive\, semidefinite$.

20 $Decision\, Trees$

עץ החלטה הוא predictor $h : \mathcal{X} \to \mathcal{Y}$ אשר נותן חיזוי לתיוגים על ידי מסלול בעץ החלטה מהשורש לעלים. בכל קודקוד ישנו תנאי התלוי ב־$feature$ אחד או יותר ובהתאם לקיומו נבחר תת העץ אליו ממשיכים. בדרך הכלל התנאי הינו תנאי של $threshold$ $(\leq, \geq)$ עבור $feature$ יחיד.

נוכל לחשוב על עץ החלטה בתור חיתוכים של המרחב שלנו כל פעם לחצאי־מרחב בהתאם לתנאי ולבסוף לתאים כאשר כל עלה ממפה לתא ב־$feature\, space$. נובע כי עץ בעל k עלים יכול לנתץ סט $C \subset \mathcal{X}$ בגודל k. מכאן כי עצים בגדלים שונים מספקים מחלקות היפותזות בעלות $VCdim$ שונות.



נבחין כי גם במקרה הפשוט של $\mathcal{X} = \{0,1\}^d$, $\mathcal{Y} = \{0,1\}$ ותנאי החלטה עם $feature$ בודד "ימינה אם $\mathbb{1}_{x.i=1}$" אם ניקח עץ עם כל האופציות, 2^d עלים, ועומק $d+1$ נקבל כי ה־$VCdim(\mathcal{H}) = 2^d$. כדי להתגבר על בעיה זאת, ובכל גם למנוע מצב של $overfitting$, נוכל להשתמש בעיקרון $minimum\, description\, length\, (MDL)$ שנמצא בפרק 7 בספר הלימוד.

באופן כללי עבור d $features$ ועצים בינאריים נקבל כי בהסתברות לפחות $1-\delta$ עבור m דגימות, לכל n טבעי המייצג את מספר הקדקודים בעץ, ולכל עץ $h \in \mathcal{H}$ נקבל כי:

$$L_{\mathcal{D}}(h) \leq L_S(h) + \sqrt{\frac{(n+1)\, log_2(d+3) + log\left(\frac{2}{\delta}\right)}{2m}}$$

20.1 $Bias - Complexity\, tradeoff$

אם כך נשים לב כי במקרה של עצי החלטה המנופים ב־$tradeoff$ זה הם למעשה מספר העלים (שקול לעומק העץ) שנבחר. עבור מודלים של עצי החלטה עם תנאים מסובכים יותר או עצים שאינם בינאריים הן העומק המקסימלי והן המפתח של כל קודקוד הם המנופים שב־$tradeoff$.

20.2 אלגוריתמים

נרצה אלגוריתם שיחזיר עצים הממזערים את צד ימין בחסם האיבוד שתואר לעיל. אולם היות ומדובר בחיפוש בכל תתי הקבוצות של d כתלות ב־n ו־m לא קיים אלגוריתם יעיל לכך ולכן האלגוריתמים השונים למציאת עצי ההחלטה מבוססים על יוריסטיקות שונות.

המודל הכללי לבניית העץ מתחיל בעץ בעל עלה אחד (השורש) אשר תיוגו בהתאם ל־$majority\, vote$ ב־$training\, set$. כעת נרוץ מספר איטרציות כאשר בכל אחת נבחן את ההשפעה של פיצול עלה ספציפי. נגדיר מדד שיכמת את השיפור שבפיצול. כעת על בסיס כל הפיצולים האפשריים ניקח את הפיצול הממקסם את השיפור או שנבחר שלא לפצל את העלה.

כדי לסווג באמצעות העץ יהיו $\dot{\bigcup}_{j=1}^{N} R_j = \mathbb{R}^d$ N ה־$partitions$ שהתקבלו על ידי על ההחלטה שבנינו אזי בהינתן דגימה x לתיוג תיוגה יהיה:

$$\underset{y=\pm1}{argmax}\hat{p}_y\left(R_j\right) \quad s.t \quad \begin{array}{c} \hat{p}_y\left(R_j\right)=\frac{1}{n_j}\sum\limits_{x_i\in R_j}\mathbb{1}_{y_i=y} \\ and\, n_j = number\, of\, training\, points\, in\, R_j \end{array}$$

20.2.1 $CART - \; Classification\, and\, Regression\, Trees$

באלגוריתם זה היוריסטיקה המנחה את בניית העץ הינה חמדניות. נתחיל מהשורש ונרד לעלים, כאשר בכל שלב נבחר קואורדינטה $1 \leq j \leq d$ ו־$c \in \mathbb{R}$ כך שהפיצול לפי $x_j > c$ יצור את העץ עם ה־$classification\, error$ הנמוך ביותר. כעת נמשיך לעשות זאת לכל מלבן שהתקבל ונחפש את האופציה שבה השיפור המשמעותי ביותר ב־$misclassification\, error$. כלומר:

יהי $S = \{(x_i, y_i)\}_{i=1}^{m}$ סט האימון ונניח כי חילקנו את ה־$feature\, space$ ל־M אזורים $R_1, ..., R_M$. נניח כי התשובה עבור כל אזור היא הקבוע $c_m \in \mathcal{Y}$ אזי

$$f(x) = \sum_{m=1}^{M} c_m I\left(x \in R^d\right)$$

עבור קריטריון המינימום של $minimum\, sum\, of\, squares$ ($\sum\left(y_i-f\left(x_i\right)\right)^2$) נראה כי ה־$\hat{c}_m$ האופטימלי הוא פשוט ממוצע התיוגים באזור:

$$\hat{c}_m=avg\left(y_i\,|\,x_i\in R^d\right)$$

כעת בהתאם לעיקרון החמדני לכל $1\leq j\leq d$ יהי $s\in\mathbb{R}$ ערך ה־$threshold$ אזי נגדיר

$$R_1\left(j,s\right)=\left\{x\in\mathbb{R}^d|x_j\leq s\right\},\ R_2\left(j,s\right)=\left\{x\in\mathbb{R}^d|x_j>s\right\}$$

ונחפש j אשר הפיצול לפיו מביא למינימום את השגיאה בשני תתי העצים:

$$\min_{j,s}\left[\min_{c_1}\sum_{x_i\in R_1(j,s)}\left(y_i-c_1\right)^2+\min_{c_2}\sum_{x_i\in R_2(j,s)}\left(y_i-c_2\right)^2\right]$$

כאשר ספיציפית למקרה של קלסיפיקציה אנו בעצם מחפשים:

$$\min_{j,s}\left(n_0\left[1-\hat{p}_{y_0}\left(R_0\right)\right]+n_1\left[1-\hat{p}_{y_1}\left(R_1\right)\right]\right)$$

ועל כן המינימיזציה ניתנת לפתרון על ידי

$$\hat{c}_1=avg\left(y_i\,|\,x_i\in R_1\left(j,s\right)\right),\ \hat{c}_2=avg\left(y_i\,|\,x_i\in R_2\left(j,s\right)\right)$$

נמשיך באופן רקורסיבי.

הערה 20.1 לשם אינטואיציה ניתן להסתכל על בעיית מציאת על ההחלטה כאנלוג לבעיית חיתוך הבד מקורס אלגוריתמים, כאשר בכל איטרציה של בעיית חיתוך הבד אנו מסתכלים על תמונת אזורים $R_1,...,R_m$ שונה עד אשר נחליט להפסיק. בבעיה זו במקום חיתוך אופקי או אנכי לפי ערך כלשהו ישנן d אפשרויות חיתוך לפי ערכים כלשהם.

20.3 $Pruning$ (גיזום)

העצים שחוזרים מהאלגוריתמים לבניית העצים, למרות שבעלי $empirical\, risk$ נמוך, נוטים להיות בעלי $true\, risk$ גבוה. כשם שאופציה אחת להגביל זאת היא על ידי מספר האיטרציות לבניית העץ, נוכל גם לבצע תהליך של $pruning$ שבו נקטין את העץ בתקווה ל־$empirical\, risk$ דומה. בדרך כלל תהליך ה־$pruning$ הולך $bottom-up$ מהעלים לשורש.

נרצה להגדיר קריטריון ל"עלות" הסיבוך של העץ ($cost\, complexity$). יהי T_0 העץ שחזר מתהליך בניית העץ אזי נחפש עץ $T\subset T_0$ אשר נחסיר ממנו חלק מה־$nodes$ הפנימיים. נסמן קודקודים טרמינליים (עלים) m בהתייחס לאזור ה־R_m ונגדיר $|T|$ כמספר הקודקודים הטרמינליים ב־T. כעת:

$$N_m=\#\left\{x_i\in R_m\right\},\quad \hat{c}_m=avg\left(y_i\,|\,x_i\in R_m\right),\quad Q_m\left(T\right)=\frac{1}{N_m}\sum_{x_i\in R_m}\left(y_i-\hat{c}_m\right)^2$$

אזי קריטריון העלות סיבוך של העץ:

$$C_{\alpha}(T)=\sum_{m=1}^{|T|}N_{m}Q_{m}(T)+\alpha|T|$$
$$\Downarrow$$
$$C_{\alpha}(T)=\sum_{m=1}^{|T|}\left[1-\hat{p}_{c_{m}}(R_{m})\right]+\alpha|T|$$

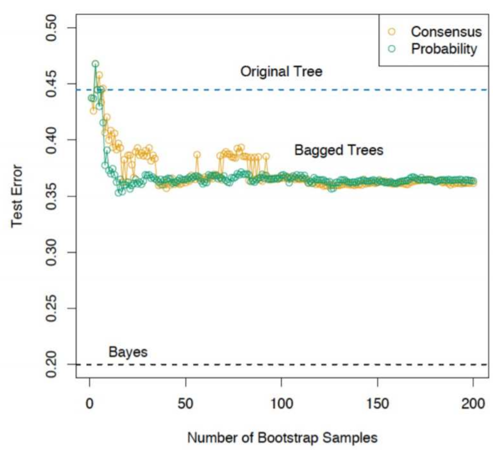
כלומר לכל α נחפש את תת העץ $T_{\alpha}\subseteq T_{0}$ אשר מביא למינימום את $C_{\alpha}(T)$ ומכאן כי α הוא המנוף בין גודל העץ לבין כמה העץ מתאים על ה־$data$. עבור α גדול נקבל עצים קטנים. נראה כי גם אלגוריתם זה נופל תחת פרדיגמת RLM כאשר $\mathcal{R}(T)=\alpha|T|$.

20.3.1 $Bias\,variance$

עבור α קטנים מקבל $bias$ נמוך ו־$variance$ גבוה. באופן כללי עצי החלטה הינם בעלי $variance$ גבוה שכן ברגע שבחרנו $feature$ וסף לפצל לפי כל יתר מבנה העץ כבר תלוי בחלטה הזאת.

20.4 שימוש ב־$Bagging$ ו־$Bootstrapping$

כפי שראינו בפרק על $Bagging$ נוכל עבור $B\in\mathbb{N}$ שנקבע לאמן B עצים $\left\{T_{\alpha}^{(b)}\right\}_{b=1}^{B}$ ובהינתן דגימה חדשה לסווגה על פי ה־$majority\,vote$ של עצים אלה.



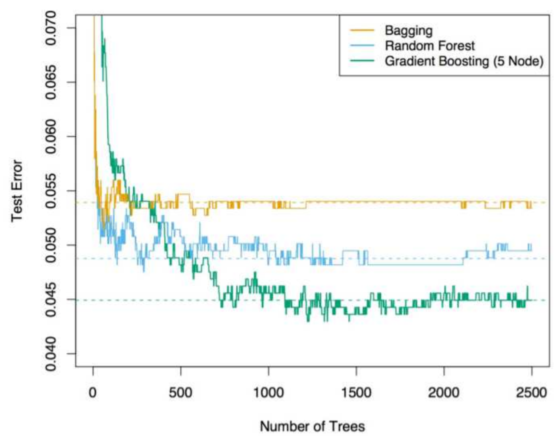
היות ועצי החלטה הינם מודל בעל $variance$ גבוה (החלטת הסף וה־$feature$ בבסיס העץ כבר משנים את מבנה כל העץ) $Bagging$ היא שיטה נוחה להורדת ה־$variance$ מבלי להעלות ב־$bias$ שכן הסיווג מתבצע על סמך מיצוע של עצים שונים.

כדי להבין מדוע זה עובד נסמן p אז סיכוי ה־$misclassification$ של העץ אזי עבור דגימה x ו־B עצים ב־$bagging$ נקבל כי ההסתברות לטעות היא $\mathbb{P}\left(X\geq\frac{B}{2}\right)$ כאשר $X\sim Bin\left(B,p\right)$. ומכאן שאשר $B\rightarrow\infty$ נקבל כי $\mathbb{P}\left(X\geq\frac{B}{2}\right)\rightarrow0$. נשים לב כי ההנחה הסמויה היא שהעצים בלתי תלויים (המשתנה הבינומי). הנחה זו איננה נכונה שכן העצים אומנו על אותו מדגם ועל כן קיימת קורלציה ביניהם אך עדין נקבל ששימוש ב־$Bagging$ משפר ביצועים ככל ש־B גדל (והיות וקיימת קורלציה לא נוכל באמת להגיע להסתברות 0 לטעות).

20.5 $Random\,Forest$

דרך נוספת להוריד את הסיכון לקבלת מסווג עם $true\,risk$ גבוה הוא על ידי $ensamble$ של עצים קטנים. $random\,forest$ הוא מסווג המכיל אוסף של $decision\,trees$. ישנן דרכים רבות ליצור עצים אלה. דרך אחת היא להגדיר $|I_i|=k$ $I_1,...,I_t\subset[d]$. כעת ניקח סט אימון S ולכל I_i נדגום את S' אשר הוא m' דגימות בהתפלגות אחידה מעל S עם החזרה. נאמן עץ על סמך ה־$features$ ב־I_i. לבסוף כאשר נרצה לתייג ניקח את התיוג על פי ה־$majority\,vote$ של כלל העצים.

כלל אצבע לבחירת k: $k\sim\sqrt{d}$.



21 $k - Nearest\, Neighbors$

בשפחת אלגוריתמים זו, בהינתן דגימה חדשה לסיווג נשתמש ב־$trainset$ כדי למצוא את k הדגימות שהכי קרובות לדגימה שקיבלנו ונשתמש בהן. ההנחה העומדת בבסיס היא שדגימות קרובות מרחבית בעלי תיוג דומה.

21.1 האלגוריתם

נסמן $\rho : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ הפונקציה המחזירה את המרחק בין שני אובייקטים ב־$\mathcal{X}$. יהי $S = \{(x_1, y_1)\}_{i=1}^{m} \subseteq \mathcal{X}$ ויהי $\pi_1(x), ..., \pi_m(x)$ סידור הדגימות ב־S על פי מרחקן $\rho(x, x_i)$. כלומר $\forall i \in [m]\ \rho\left(x, x_{\pi_i(x)}\right) \leq \rho\left(x, x_{\pi_{i+1}(x)}\right)$.

הגדרה 21.1 כלל ה־$k - NN$ עבור תיוגים בינאריים מוגדר בתור ה־$majority\, vote$ של $\left\{y_{\pi_i(x)}\ :\ i \leq k\right\}$.

הערה 21.2 במקרה של רגרסיה ($\mathcal{Y} = \mathbb{R}$) נוכל להגדיר את הכלל כממוצע התיוגים של k הדגימות הקרובות: $h_S(x) = \frac{1}{k}\sum_{i=1}^{k} y_{\pi_i(x)}$.

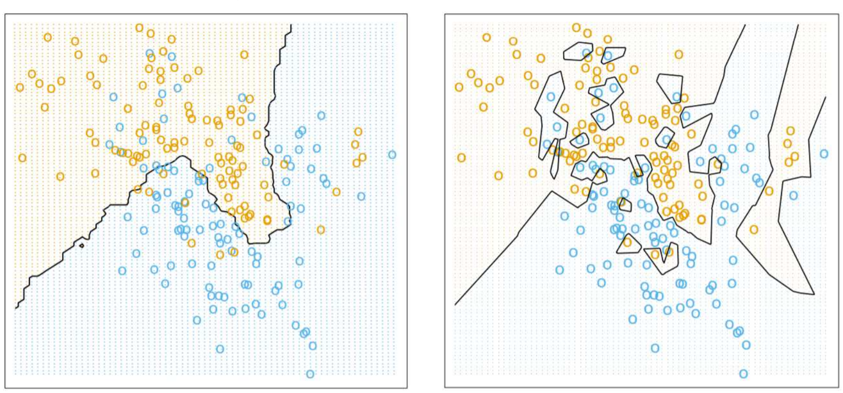
הגדרה 21.3 במקרה הכללי עבור סט תיוגים $\mathcal{Y}$ ופונקציה $\phi : (\mathcal{X} \times \mathcal{Y})^k \to \mathcal{Y}$ אזי כלל ה־$k - NN$ ביחס ל־ϕ הוא:

$$h_S(x) = \phi\left(\left(x_{\pi_1(x)}, y_{\pi_1(x)}\right), ..., \left(x_{\pi_k(x)}, y_{\pi_k(x)}\right)\right)$$

21.2 $Bias - complexity\, tradeoff$

במודל זה המנוף של $tradeoff$ זה מבוטא בערך ה־k אשר הוא מספר השכנים שממצעים. ככל ש־k קטן יותר אזי המודל "גמיש יותר" ורגיש יותר לרעש שקיים בכל אחת מהדגימות. עבור k גדול יותר המיצוע נעשה על יותר דגימות וכך הרעש של כל דגימה אינדיבידואלית פחות משפיע.

$k = 15$ $\qquad\qquad$ $k = 1$



באופן כללי, כל אזור ($neighborhood$) ב־$k-NN$ תלוי משכנים שמסביב ועל כן הגבולות הללו שבין התיוגים מאוד $wiggly$ ו־$unstable$. מכאן כי סך הכל במודל יש $variance$ גבוהה ו־$bias$ נמוך.

21.3 $Bellman's\, "Curse\, of\, dimensionality"$

קיימת טענה כי ניתן להגיע ב־$1-NN$ ל־$true\, error$ של $2L_{\mathcal{D}}\left(h^*\right)+\epsilon$ כאשר h^* הוא ה־ $Bayes\, optimal$ (עמוד 260 בספר הקורס) ובאופן כללי ב־$k-NN$ לחסם של $\left(1+\sqrt{8/k}\right)$ פעמים השגיאה של $Bayes\, optimal$. מהוכחת הטענה נובע כי גודל סט האימון הנדרש גדל <u>אקספוננציאלית</u> עם גדילת מימד ה־$features$ וכי עבור התפלגויות שונות זהו תנאי הכרחי ללמידה:

משפט 21.4 לכל $c>1$ ולכל כלל למידה L קיימת התפלגות $[0,1]^d \times \{0,1\}$ כך שה־ $Bayes\, error$ הינה 0 אבל אבל לכל $m \leq (c+1)^d/2$ דגימות אימון ה־$true\, error$ גדול מ־$\frac{1}{4}$.

על כן נרצה להוריד את מימד ה־$feature\, space$ כלל שניתן. ישנן דרכים שונות לעשות זאת כמו שימוש ב־$Kernels$ או $Feature\, Selection$ למציאת ה־$features$ להם המשקל הרב ביותר בניבוי נכון של התיוגים.

22 $Convex\, Learning\, Problems$

22.1 בעיות למידה

להגדרות המתמטיות אודות קבוצות ופונקציות קמורות, לפשיציות וחלקות ראו בפרק ה"ידע מוקדם". רוב בעיות הלמידה שנוכל להבטיח עליהן למידה יעילה (PAC וחישובי) הן בעיות קמורות, ליפשיציות וחסומות. ישנן בעיות למידות שאינן קמורות (כמו בשימוש שרשתות נוירונים) ולא כל בעיה קמורה ניתנת ללימוד. בבעיות אלה נתייחס למחלקת ההיפותזות כתת קבוצה $\mathcal{H}\subseteq\mathbb{R}^d$ ונסמן היפותזה במחלקה ב־w.

הגדרה 22.1 בעיית למידה $(\mathcal{H},\mathcal{Z},\ell)$ תקרא קמורה אם ורק אם מחלקת ההיפותזות $\mathcal{H}$ הינה קבוצה קמורה ופונקציית האיבוד ℓ קבוצה קמורה בארגומנט הראשון שלה (לכל $z\in\mathcal{Z}$ מתקיים $\ell\left(w,z\right)$ היא פונקציה קמורה).

הערה 22.2 על אף שקיימים אלגוריתמים יעילים ללמידה של בעיות קמורות לא מתקיימת הכלה בין הקבוצות: לא כל הבעיות הקמורות למידות ולא כל הבעיות הלמידות קמורות. (קיימת דוגמה של רגרסיה לינארית הומוגנית ספציפית אשר איננה למידה למרות שהיא קמורה ־ עמוד 164). אולם אם נוסיף שני תנאים נוספים נוכל להפוך בעיית למידה זו לקמורה:

1. <u>$\mathcal{H}$ חסומה (עם פרמטר B)</u> - אם לכל $w\in\mathcal{H}$ מתקיים כי $||w||\le B$.

2. <u>ℓליפשיצית (עם פרמטר ρ)</u> - אם לכל $z\in\mathcal{Z}$ $f\left(w\right)=\ell\left(w,z\right)$ היא ρ-ליפשיצית. הבעיה תקרא ρ-ליפשיצית.

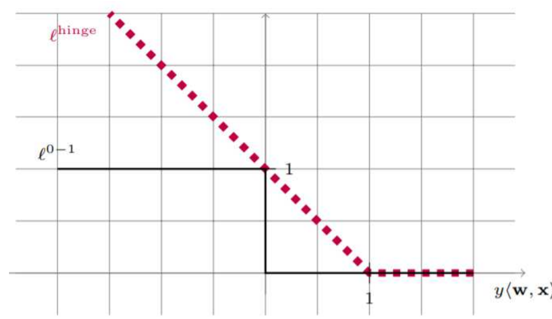
מכאן נובעת ההגדרה הבאה:

הגדרה 22.3 בעיית למידה $\left(\mathcal{H},\mathcal{Z},\ell\right)$ תקרא קמורה-לפשיצית-חסומה ($Convex-Lipschitz-Bounded$) עם פרמטרים ρ,B אם:

1. מחלקת ההיפותזות $\mathcal{H}$ קבוצה קמורה כאשר $||w||\le B$ $\forall w\in\mathcal{H}$.

2. לכל $z\in\mathcal{Z}$ פונקציית האיבוד $\ell\left(\cdot,z\right)$ קמורה ו-ρ-ליפשיצית.

דוגמה: לפונקציית איבוד קמורה הוא ה-ℓ^{hinge}:



הגדרה 22.4 בעיית אופטימיזציה תקרא קמורה אם היא מהצורה $\underset{u\in U}{min}f\left(u\right)$ עבור U,f קמורות.

למה 22.5 עבור $\mathcal{H}$ מחלקת היפותזות קמורה ו-ℓ פונקציית איבוד קמורה אזי $ERM_{\mathcal{H}}$, הבאה למינימום של השגיאה האמפירית, הינה בעיית אופטימיזציה קמורה

הוכחה: תהי $\mathcal{H}$ מחלקת היפותזות קמורה ו-ℓ פונקציית איבוד קמורה. נזכר כי $ERM_{\mathcal{H}}\left(S\right)=\underset{w\in\mathcal{H}}{argmin}L_{S}\left(w\right)$. עבור סט אימון $S=\left\{ z_{i}\right\}_{i=1}^{m}$ מתקיים כי $L_{S}\left(w\right)=\frac{1}{m}\sum_{i=1}^{m}\ell\left(w,z_{i}\right)$ $\forall w\in\mathcal{H}$. היות ו-ℓ קמורה ופונקציות קמורות סגורות תחת סכום וכפל בסקלר אזי גם L_{s} קמורה. מכאן כי $ERM_{\mathcal{H}}$ בעיית אופטימיזציה קמורה. ■

22.2 פונקציית איבוד $Surrogate$

כפי שכבר ראינו ניתן ללמוד בעיות למידה קמורות בצורה יעילה ובמקרים אלה נרצה גם פונקציית איבוד קמורות. למשל פונקציית האיבוד ℓ^{0-1} איננה קמורה ומימוש עיקרון ה-ERM עבורה הוא בעיה NP-קשה. נרצה אפוא להגדיר פונקציית איבוד חדשה אשר תספק לנו את התכונת הקמירות ושתחסום מלמעלה את פונקציית האיבוד המקורית. עבור הדוגמה של ℓ^{0-1} פונקציית האיבוד המתאימה לכך היא ℓ^{hinge}.

ברגע שהגדרנו את ה־$surrogate\,loss\,function$ נרצה לבדוק את חסם השגיאה. עבור הדוגמה של ℓ^{0-1},ℓ^{hinge} נראה כי:

$$L_{\mathcal{D}}^{hinge}\left(\mathcal{A}\left(S\right)\right)\leq\underset{w\in\mathcal{H}}{min}L_{\mathcal{D}}^{hinge}\left(w\right)+\epsilon\quad for\;\;L_{\mathcal{D}}^{hinge}\left(w\right)=\underset{\left(x,y\right)\sim\mathcal{D}}{\mathbb{E}}\left[\ell^{hinge}\left(w,\left(x,y\right)\right)\right]$$

מתכונות ה־$surrogate$ נוכל להקטין את אגף שמאל ולכתוב את אגף ימין באופן הבא:

$$L_{\mathcal{D}}^{0-1}\left(\mathcal{A}\left(S\right)\right)\leq\underset{w\in\mathcal{H}}{min}L_{\mathcal{D}}^{0-1}\left(w\right)+\left(\underset{w\in\mathcal{H}}{min}L_{\mathcal{D}}^{hinge}\left(w\right)-\underset{w\in\mathcal{H}}{min}L_{\mathcal{D}}^{0-1}\left(w\right)\right)+\epsilon$$

וקיבלנו חסם עליון ל־L^{0-1} על ידי שלושה ביטויים:

1. הראשון הוא ה־$approximation\,error$ המודד כמה טוב מצליחה מחלקת ההיפותזות על ההתפלגות.
2. השני הוא ה־$estimation\,error$ (ϵ) הנובע מההבדל בין סט האימון לכלל ההתפלגות.
3. השלישי הוא הביטוי בסוגריים והוא ה־$optimization\,error$. הוא מודד את ההבדל בין ה־$approximation\,error$ ביחס ל־$surrogate\,loss$ ופונקציית האיבוד המקורית. שגיאה זו נובעת מחוסר היכולת שלנו להביא למינימום את האיבוד ב־$training\,set$ עם פונקציית האיבוד המקורית.

23 $Stochastic\,Gradient\,Descent$

עד כה כאשר עסקנו בבעיות למידה ניסינו להביא למינימום את ה־$empirical\,risk$. בפרק זה נעסוק בבעיות למידה קמורות ובאמצעות (SGD) $stochastic\,gradient\,descent$ ננסה להביא למינימום לא את השגיאה האמפירית אלא את ה־$risk\,function$ $L_{\mathcal{D}}\left(h\right)$ באופן ישיר. שיטה זו עובדת באופן איטרטיבי כאשר בכל שלב נלך בכיוון הנגדי לגרדיאנט של הפונקציה בנקודה. נשים לב כי היות ואיננו יודעים את $\mathcal{D}$ איננו יודעים את הגרדיאנט. כדי להתגבר על כך SGD מתקדם בכיוון שהערך הצפוי בו הוא נגדי לגרדיאנט. היות ובפרק זה נעבוד עם בעיות למידה קמורות נסמן היפותזה במחלקת ההיפותזות בתור $w\in\mathcal{H}\subseteq\mathbb{R}^{d}$.

23.1 $Gradient\,Descent$

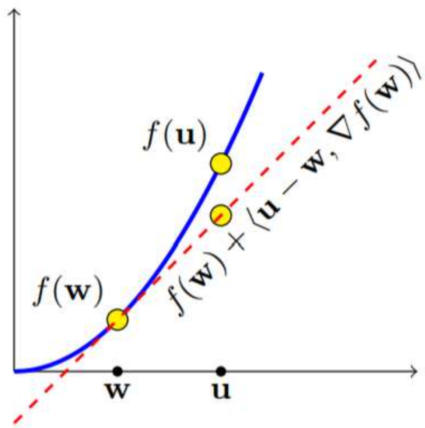
כפי שכבר ראינו הגרדיאנט של פונקציה דיפרנציאבילית $f:\mathbb{R}^{d}\to\mathbb{R}$ בנקודה $w\in\mathbb{R}^{d}$ הינו וקטור הנגזרות החלקיות $\nabla f\left(x\right)=\left(\frac{\partial f(x)}{\partial w[1]},...,\frac{\partial f(x)}{\partial w[d]}\right)$. בשלב זה נניח כי הגרדיאנט ידוע לנו

האלגוריתם:

- נתחיל עם $w^{(1)}=0$
- בכל איטרציה נצעד נגד כיוון הגרדיאנט: $w^{(t+1)}=w^{(t)}-\eta\nabla f\left(w^{(t)}\right)$ כאשר $\eta>0$

• פלט האלגוריתם הוא $\bar{w} = \frac{1}{T}\sum_{t=1}^{T} w^{(t)}$

למה 23.1 עבור $f : \mathbb{R}^d \to \mathbb{R}$ קמורה ואינטגרבילית בנקודה $w \in \mathbb{R}^d$ מתקיים כי $\forall u \in \mathbb{R}^d \ f(u) \geq f(w) + \langle \nabla f(w), u - w \rangle$. כלומר המשיק של f ב־w נמצא מתחת לגרף הפונקציה.



נראה כי אם w קרוב ל־$w^{(t)}$ אזי הגרדיאנט של f בנקודה w מספק את קירוב טיילור מסדר ראשון של f סביב w על ידי $f(u) \approx f(w) + \langle u - w, \nabla f(w) \rangle$. מכאן כי עבור w הקרוב ל־$w^{(t)}$ נקבל כי

$$f(w) \approx f\left(w^{(t)}\right) + \left\langle w - w^{(t)}, \nabla f\left(w^{(t)}\right)\right\rangle$$

על כן נרצה להביא למינימום את הקירוב וגם לוודא כי אנו נשארים קרובים ל־$w^{(t)}$. פרמטר $\eta > 0$ הוא ששולט ב־$tradeoff$ בין שני אלה ולכן נעדכן את הכלל להיות (אם ניקח כלל זה, נגזור ונשווה לאפס נקבל את הכלל שכתוב באלגוריתם לעיל):

$$w^{(t+1)} = \underset{w}{argmin} \frac{1}{2}\left\|w - w^{(t)}\right\|^2 + \eta\left(f\left(w^{(t)}\right) + \left\langle w - w^{(t)}, \nabla f\left(w^{(t)}\right)\right\rangle\right)$$

23.2 $Sub\,Gradient\ Descent$

אלגוריתם ה־GD מסתמך על כך ש־f דיפרנציאבילית. כעת נקל הנחה זו ונראה כיצד בעזרת $sub - gradient$ של $f(w)$ ב־$w^{(t)}$ נוכל להשתמש באלגוריתם גם על פונקציות שאינן דיפרנציאביליות.

הגדרה 23.2 תהי S קבוצה קמורה פתוחה. פונקציה $f : S \to \mathbb{R}$ הינה קמורה $\iff$ לכל $w \in S$ קיים v כך ש:

$$\forall u \in S \ \ f(u) \geq f(w) + \langle u - w, v \rangle$$

הגדרה 23.3 וקטור v המקיים את התנאי מההגדרה הקודמת יקרא ה־$sub-gradient$ של f בנקודה w. קבוצת כל ה־$sub-gradient$ של f בנקודה w נקראת ה־$differential\ set$ ומסומנת $\partial f(w)$.

למה 23.4 f הינה קמורה $\iff$ לכל w מתקיים כי $\partial f(w) \neq \emptyset$.

הערה 23.5 בספר יש הסבר על חישוב ה־$sub-gradient$ לפונקציות אולם חלק זה לא הועבר בשיעור. בנוסף האלגוריתם שהוצג עבור שימוש בגרדיאנט מתאים גם עבור שימוש ב־$sub-gradient$.

האלגוריתם

Algorithm 1 Subgradient Descent

Parameter: $T, \eta > 0$
Initialize: $w^{(1)} = \mathbf{0}$
for $t = 1, \dots, T$ **do**
  $w^{(t+1)} = w^{(t)} - \eta v_t$, where $v_t \in \partial f(w^{(t)})$
end for
Return: $\bar{w} = \frac{1}{T}\sum_{t=1}^{T} w^{(t)}$

משפט 23.6 תהי f פונקציה קמורה, ρ־ליפשיצית ויהי $w^* \in \underset{w}{argmin} f(w)$. אם נריץ את אלגוריתם $sub-gradient\ descent$ על f ל־T איטרציות עם $\eta = \frac{||w^*||}{\rho\sqrt{T}}$ אזי וקטור התוצאה $\bar{w}$ מקיים

$$f(\bar{w}) \leq f(w^*) + \frac{||w^*||\,\rho}{\sqrt{T}}$$

מסקנה 23.7 תהי f פונקציה קמורה, ρ־ליפשיצית ויהי $w^* \in \underset{w}{argmin} f(w)$. לכל $\epsilon > 0$ אם נריץ את אלגוריתם $sub-gradient\ descent$ על f עבור לפחות $\frac{||w^*||^2\rho^2}{\epsilon^2}$ צעדים עם $\eta = \frac{||w^*||}{\rho\sqrt{T}}$ אזי וקטור התוצאה $\bar{w}$ מקיים:

$$f(\bar{w}) \leq f(w^*) + \epsilon$$

דוגמה: מציאת על־מישור מפריד. יהי $S = \{(x_i, y_i)\}_{i=1}^{m}$ אזי נרצה $w \in \mathbb{R}^d$ כך ש: $\forall i\ \ y_i \langle w, x_i\rangle > 0$. נסמן w^* העל־מישור המפריד של נורמת היחידה ונניח בה"כ כי $||x_i|| = 1$. נסמן את השולים ($margin$) של S על ידי $\gamma = \underset{i}{min} y_i \langle w^*, x_i\rangle$.
נחפש את שוליים כך ש: $\underset{w}{max}\,\underset{i}{min}\, y_i \langle w, x_i\rangle$ השקול למציאת $\underset{w}{min}\,\underset{i}{max} - y_i \langle w, x_i\rangle$ שכן אם w איננו על העל־מישור המפריד ערך המינימום יהיה שלילי ולכן לא יבחר. על כן אנו מתבוננים בבעיה:

$$\underset{w}{min} f(w) \quad where \quad f(x) = \underset{i}{max} - y_i \langle w, x_i\rangle$$

נבחין כעת כי f קמורה בהיותה הרכבה של max על אוסף פונקציות אפיניות (אשר קמורות) וכי ה־$sub-gradient$ של f בנקודה w הוא $-y_i x_i$ עבור i כלשהו השייך ל־$-y_i \langle w, x_i\rangle$.

על כן נרצה את האלגוריתם הבא:

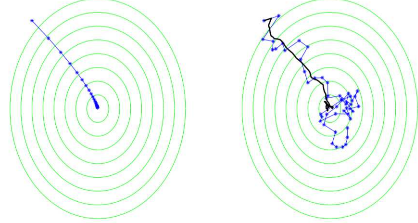
- נאתחל $w^{(1)}=0$
- לכל $t=1,...,\left\lceil \frac{1}{\gamma^{2}}\right\rceil +1$ נבצע:
  - נמצא i המקיים: $i\in \underset{i}{argmax}\,-y_{i}\left\langle w,x_{i}\right\rangle$
  - נעדכן: $w^{(t+1)}=w^{(t)}+\eta y_{i}x_{i}$
- נחזיר $\bar{w}=\frac{1}{T}\sum\limits_{t=1}^{T}w^{(t)}$

נשים לב כי f הינה 1־ליפשיצית וכי $f\left(w^{*}\right)=-\gamma$. מכאן כי לאחר $t>\frac{1}{\gamma^{2}}$ איטרציות נקבל כי $f\left(\bar{w}\right)<f\left(w^{*}\right)+\gamma=0$ ועל כן $\bar{w}$ הינו אכן על־מישור מפריד כנדרש.

עלות האלגוריתם - על האלגוריתם לרוץ לפחות $\frac{1}{\gamma^{2}}$ איטרציות ובכל איטרציה עלות החישוב dm. לכן זמן הריצה של האלגוריתם $O\left(\frac{dm}{\gamma^{2}}\right)$.

23.3 אלגוריתם $Stochastic\,Gradient\,Descent\;(SGD)$

נזכר כי אנו מנסים למצוא את ההיפותזה שתביא למינימום את השגיאה (מעל S או מעל $\mathcal{D}$). SGD הינו אלגוריתם בעל וריאציות רבות אשר המטרה היא שימוש בדגימה אקראית (יחידה או קבוצה, $batch$) לחישוב הגרדיאנט והתקדמות בכיוון הנגדי. בתוחלת נקבל כי אכן נתקרב למינימום. באלגוריתם GD השתמשנו ב־$\nabla f\left(w^{(t)}\right)$ כאשר f במקרה שלנו היא פונקציית האיבוד. נשים לב כי למקרה של $f\left(w\right)=L_{\mathcal{D}}\left(w\right)=\underset{z\sim\mathcal{D}}{\mathbb{E}}\left[\ell\left(w,z\right)\right]$ איננו יודעים את $L_{\mathcal{D}}$ ולכן איננו יודעים כיצד לחשב את $\nabla L_{\mathcal{D}}$. בהמשך נראה כיצד להתמודד עם זאת. בנוסף גם אילו ידענו את ההתפלגות והיה ביכולתנו לחשב את הגרדיאנט העלות החישובית של חישובו בכל נקודה הייתה גדולה. על כן ב־SGD, נבחר בכל פעם נקודה $z\sim\mathcal{D}$ (או מספר מצומצם של נקודות) באופן מקרי ונחשב את ה־$sub-gradient$ של $sample$ זה.



23.3.1 SGD עבור מינימום $L_{\mathcal{D}}$

למרות שאיננו יודעים את $\mathcal{D}$ נוכל לחשב שיערוך ל־$sub-gradient$ של $L_{\mathcal{D}}\left(w\right)$ היות ומתקיים:

$$\begin{aligned}\nabla L_{\mathcal{D}}\left(w^{(t)}\right) & = \nabla\underset{z\sim\mathcal{D}}{\mathbb{E}}\ell\left(w^{(t)},z\right)\\ & = \underset{z\sim\mathcal{D}}{\mathbb{E}}\nabla\ell\left(w^{(t)},z\right)\end{aligned}$$

וכעת:

Stochastic Gradient Descent (SGD) for minimizing $L_{\mathcal{D}}(\mathbf{w})$

parameters: Scalar $\eta > 0$, integer $T > 0$
initialize: $\mathbf{w}^{(1)} = \mathbf{0}$
for $t = 1, 2, \ldots, T$
  sample $z \sim \mathcal{D}$
  pick $\mathbf{v}_t \in \partial\ell(\mathbf{w}^{(t)}, z)$
  update $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta\mathbf{v}_t$
output $\bar{\mathbf{w}} = \frac{1}{T}\sum_{t=1}^{T}\mathbf{w}^{(t)}$

מסקנה 23.8 תהי בעיית למידה קמורה־ליפשיצית־חסומה עם פרמטרים ρ, B. אזי לכל $\epsilon > 0$ אם נריץ את SGD עם מספר איטרציות $T \geq \frac{B^2\rho^2}{\epsilon^2}$ ו־ $\eta = \sqrt{\frac{B^2}{\rho^2 T}}$ אזי אלגוריתם SGD מקיים:

$$\mathbb{E}\left[L_S\left(\bar{w}\right)\right] \leq \underset{w\in\mathcal{H}}{min} L_{\mathcal{D}}\left(w\right) + \epsilon$$

23.3.2 SGD עבור מינימום L_S (ERM)

נשים לב כי היות ופונקציית הגרדיאנט הינה טרנספורמציה לינארית מתקיים:

$$\begin{aligned}\nabla L_S\left(w\right) &= \nabla\left(\frac{1}{m}\sum_{i=1}^{m}\ell\left(w, z_i\right)\right)\\ &= \frac{1}{m}\sum_{i=1}^{m}\nabla\ell\left(w, z_i\right)\\ &= \underset{i\sim[m]}{\mathbb{E}}\left[\nabla\ell\left(w, z_i\right)\right]\end{aligned}$$

טענה 23.9 אם לכל $i \in [m]$ מתקיים כי $v_i \in \partial\ell\left(w, z_i\right)$ אזי $\frac{1}{m}\sum_i v_i = \underset{i\in[m]}{\mathbb{E}}\left[v_i\right] \in \partial L_S\left(w\right)$. **הוכחה:** נסמן $v = \frac{1}{m}\sum_i v_i$. לכל u מתקיים:

$$\begin{aligned}L_S\left(u\right) &= \frac{1}{m}\sum_{i=1}^{m}\ell\left(u, z_i\right)\\ &\geq \frac{1}{m}\sum_{i=1}^{m}\left(\ell\left(u, z_i\right) + \langle v_i, u - v\rangle\right)\\ &= L_s\left(w\right) + \langle v, u - w\rangle\end{aligned}$$

כלומר אם נגריל דגימה מסט האימון ונלך נגד כיוון ה־$subgradient$ המתאים לדגימה z_i בנקודה w אזי בתוחלת נתקרב לערך הנכון. ■

אזי אלגוריתם ה־SGD להבאת ה־$L_S\left(w\right)$ למינימום נראה כך:

```
Stochastic Gradient Descent (SGD) for minimizing L_S(w)
```

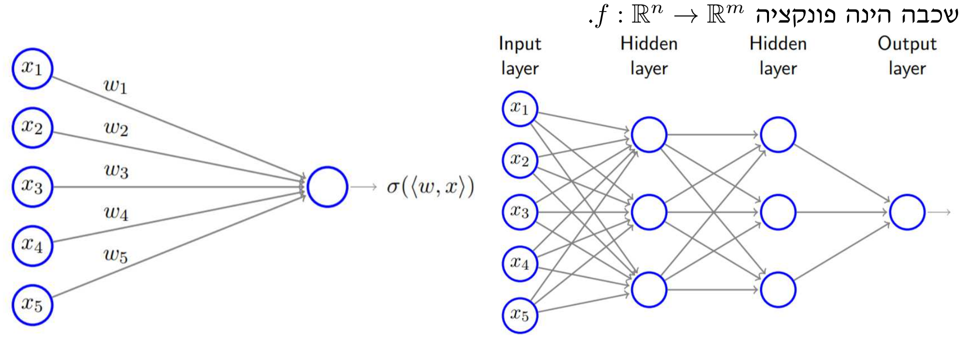
parameters: Scalar $\eta > 0$, integer $T > 0$
initialize: $w^{(1)} = \mathbf{0}$
for $t = 1, 2, \ldots, T$
  choose $i \in [m]$ at random
  update $w^{(t+1)} = w^{(t)} - \eta v_t$, where $v_t \in \partial\ell(w^{(t)}, z_i)$
output $\bar{w} = \frac{1}{T}\sum_{t=1}^{T} w^{(t)}$

מסקנה 23.10 תהי בעיית למידה קמורה־ליפשיצית־חסומה עם פרמטרים ρ, B. אזי לכל $\epsilon > 0$ אם נריץ את SGD עם מספר איטרציות $T \geq \frac{B^2\rho^2}{\epsilon^2}$ ו־$\eta = \sqrt{\frac{B^2}{\rho^2 T}}$ אזי אלגוריתם SGD מקיים:

$$\mathbb{E}\left[L_S\left(\bar{w}\right)\right] - L_S\left(w^*\right) \leq \epsilon$$

24 $Deep\,Learning$

הרעיון המרכזי ב־$neural\,netwrks$ הוא הרכבה של צירופים לינאריים של קלט כ־$features$ ומידול פונקציית המטרה כפונקציה לא לינארית של $features$ אלה. כל אחד מהקודקודים ברשת, הנוירונים, מבצע חישוב פשוט, כאשר ההרכבה שלהם מאפשרת ביצוע חישובים מורכבים. כל רשת מכילה אוסף שכבות כאשר כל שכבה למעשה מבצעת פעולה חישובית כלשהי. סוגי שכבות שונות, ושילובים של פונקציות שונות הם שבסופו של דבר נותנים לרשת הנוירונים את הכוח שלה. כל שכבה הינה פונקציה $f:\mathbb{R}^n\to\mathbb{R}^m$.



בקורס אנו עוסקים רק ברשתות המכונות $feedforward\,networks$. $feedforward\,networks$ הינן גרף מכוון חסר מעגלים $G=\langle V,E\rangle$ ופונקציית משקולות $w:E\to\mathbb{R}$. כל $neuron/nodes$ מיוצג על ידי פונקציה פשוטה $\sigma:\mathbb{R}\to\mathbb{R}$ המכונה פונקציית האקטיבציה ($acctivation\,function$) של ה־$node$. כל קשת בגרף מחברת בין ה־$output$ של $node$ קודם ל־$input$ של $node$ הבא כאשר עבור $node$ נתון ה־$input$ שלו הוא שיכלול ה־$outputs$ של סט $nodes$ ומשקולות לכל $node$ בסט (כפי שניתן לראות בתמונה השמאלית).

כל ה־$nodes$ מאורגנים בשכבות, $layers$, ומכאן כי נוכל לתאר את כל ה־$nodes$ בתור: $V=\dot{\bigcup}_{t=0}^{T}V_t$ וכל קשת ב־E מחברת בין $nodes$ ב־V_{t-1} ל־$node$ ב־V_t (ניתן להגדיר רשתות אחרות בהן קשתות יכולות להיות תוך שכבתיות). השכבה V_0 מכונה ה־$input\,later$ והשכבה V_T ה־$output\,layer$. יתר השכבות $V_1,...,V_{T-1}$ מכונות $hidden\,layers$.

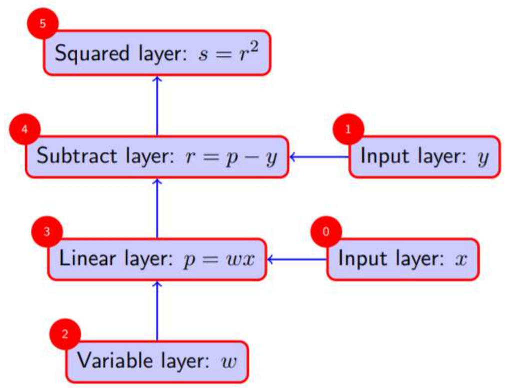
בתמונות הנ"ל כל $node$ מייצג נוירון אשר מקבל קלט מ־n $input\,nodes$ ומוציא $output$ כלשהו על בסיס פונקציית האקטיבציה שמחשב הנוירון. כמובן שנוכל לתת לכל קלט בסט הקלטים של נוירון משקולת w ובכך לציין איזה $input$ משמעותי יותר לנוירון (כאשר מדברים על אימון/בניית הרשת מדברים על מהן הפונקציות שמבצעים הנוירונים וכן מהן המשקולות השונות). שתי השכבות החיצוניות של רשת הנוירונים מכונות ה־$input\,layer$ וה־$output\,layer$ אליהן יש לנו גישה. אוסף השכבות הפנימיות מכונות $hidden\,layers$.

אם כך ניתן לייצג רשת שלמה בתור (V,E,σ,w) אשר מגדירה לנו פונקציה $h_{V,E,\sigma,w}:\mathbb{R}^{|V_0|}\to\mathbb{R}^{|V_T|}$ כאשר כל סט של פונקציות שכאלה מגדיר מחלקת היפותזות. השלשה (V,E,σ) מכונה הארכיטקטורה של הרשת כאשר בעיות שונות מאופיינות בארכיטקטורה שונה שנמצאה מתאימה להן.

24.1 $Computation\, Graph\, and\, Backpropagating$

שיטה שעובדת טוב היא שילוב בין SGD ורשתות נוירונים כאשר פונקציית הפלט מהרשת תהיה פונקציית $loss$ שעליה נפעיל SGD להבאת האיבוד למינימום. על כן תהי מחלקת היפותזות המאופיינת בוקטור $\theta \in \mathbb{R}^d$ ופונקציית האיבוד $\ell\left(\theta,(x,y)\right)$. נניח כי ℓ פונקציה דיפרנציאבילית ב־θ אזי נוכל לחשב את $\nabla\ell\left(\theta,(x,y)\right)$. כאשר נחפש מינימום של $L_{\mathcal{D}}$ או L_S בעזרת SGD נתחיל מ־$\theta^{(0)}$ ונעדכן בכל איטרציה $\theta^{(t+1)}=\theta^{(t)}-\eta t\nabla\ell\left(\theta^{(t)},(x,y)\right)$. בנקודה זו חשוב לשים לב כי בפועל לא מובטח לנו כי הפונקציה או בעיית הלמידה קמורות ובנוסף לא מובטח לנו שקצב הצעדים (η) קטן דיו. על כן יתכן שנתכנס למינימום שאיננו כלל המינימום הגלובאלי (בפונקציה קמורה מינימום מקומי הוא מינימום גלובאלי אך לא כך בפונקציות לא קמורות). בנוסף כלל לא מובטח לנו שהפונקציה הינה דיפרנציאבילית בכלל. בכל זאת, נניח כי הפונקציה ניתנת לתיאור על ידי הרכבה של פונקציות דיפרנציאביליות פשוטות וחיפוש המינימום יעשה באמצעות הגרדיאנט. במבחן התוצאה השיטה עובדת.

אם כך נרצה מודל שיאפשר לנו לחשב את פונקציית ה־$loss$ שנבחר בצורה יעילה וכן את הגרדיאנט בצורה יעילה. היות ואנו מניחים כי הפונקציה הכללית ניתנת לתיאור על ידי הרכבה של פונקציות דיפרנציאביליות פשוטות נוכל על בסיסן, מכלל השרשרת, לחשב את הגרדיאנט של הפונקציה המקורית. כדוגמה נניח כי פונקציית האיבוד ℓ היא $Least\, Squares$ מעל $\mathbb{R}$, אזי הגרף מתחת מראה את גרף חישוב של $Least\, Squares$ במימד אחד.



נראה כי קיימים שלושה סוגים של $layers$:

- $input\, layer$ המגדירה קבלת קלט כלשהו.
- $variable\, layer$ המגדירה את הפרמטרים של הבעיה.
- $functional\, layers$ המגדירות את הפעולות החישוביות הפשוטות שנעשה בכל שלב.

מספורי השכבות מייצגים את הסידור הטופולוגי של השכבות ואת סדר החישוב. ונבחין כי:

1. <u>$Forward$</u> - נוכל לייצג גרף זה גם בכתיב מתמטי כהרכבה של הפונקציות השונות: $\ell(w)=s\left(r_y\left(p_x(w)\right)\right)=\left(s\circ r_y\circ p_x\right)(w)$ ($subscript$ מייצג את הקבועים של הפונקציה). אם נלך עם הסידור הטופולוגי אכן בסופו נקבל חישוב מלא של ℓ כהרכבה של פונקציות פשוטות. בקוד נוכל לעשות זאת בפשטות על ידי:

$$\begin{aligned}&For\, t=0,...,T-1:\\&\quad Layer[t].output\;=\;Layer[t].function\left(Layer[t].inputs\right)\end{aligned}$$

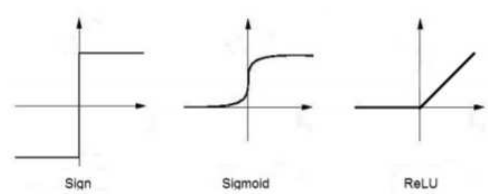
2. <u>$Backward$</u> - כעת נרצה לחשב את הנגזרת של פונקציית ℓ. נזכר שעל פי כלל השרשרת $\ell'(w) = s'(r_y(p_x(w))) \cdot r'_y(p_x(w)) \cdot p'_x(w)$ ועל כן אם לאחר שעלינו בעץ עם כיוון הסידור הטופולוגי וחישבנו את ערכי הביניים, נרד בעץ בחזרה ובכל שלב נחשב את הנגזרת של הפונקציה הפשוטה באותה שכבה נצליח לחשב את הנגזרת של פונקציות מסובכות באופן יעיל. זהו ה-<u>$backpropagation$</u>. בקוד נוכל לעשות זאת בפשטות על ידי:

$$
\begin{aligned}
& Layer\,[T-1]\,.delta = 1 \\
& For\, t = T-1, ..., 0: \\
& \quad For\, i\ in\ Layer\,[t]\,.inputs: \\
& \qquad i.delta\ =\ Layer\,[t]\,.delta\ *\ Layer\,[t]\,.derivative\,(i,\ Layer\,[t]\,.inputs)
\end{aligned}
$$

$delta$ מייצג את תוצאת חישוב הנגזרת באותו השלב. למעשה אם נבטיח שכשנבנה את גרף החישוב נשתמש רק בפונקציות בעלות נגזרות פשוטות שנוכל לשמור כמשתנים של השכבה וכך בדרך חזרה למטה נצליח לחשב את הנגזרת ביעילות.
כשנסיים את ההרצה בגרף למעשה קיבלנו את הערך של $\nabla \ell,$ עבור הנקודה w שהזנו כמשתנה. נעשה זאת בכל איטרציה של ה-SGD לחישוב ה-$w^{(t)}$.

דוגמאות ל-$layers$ נפוצים:

1. שכבה אפינית - $O = WX + b \cdot 1^T$ עבור $W \in \mathbb{R}^{m\times n},\ x \in \mathbb{R}^{n\times c}, \in \mathbb{R}^m$
2. שכבה אונרית - $f:\mathbb{R}\to\mathbb{R}\ ,\ \forall i,\ o_i = f(x_i)$ כאשר f לדוגמה היא:
   (א) פונקציית סיגמואיד - $f(x) = (1+exp(-x))^{-1}$
   (ב) פונקציית $ReLU$ $(Rectified\,Linear\,Unit)$ - $f(x) = max\{0, x\}$
   (ג) $sign$

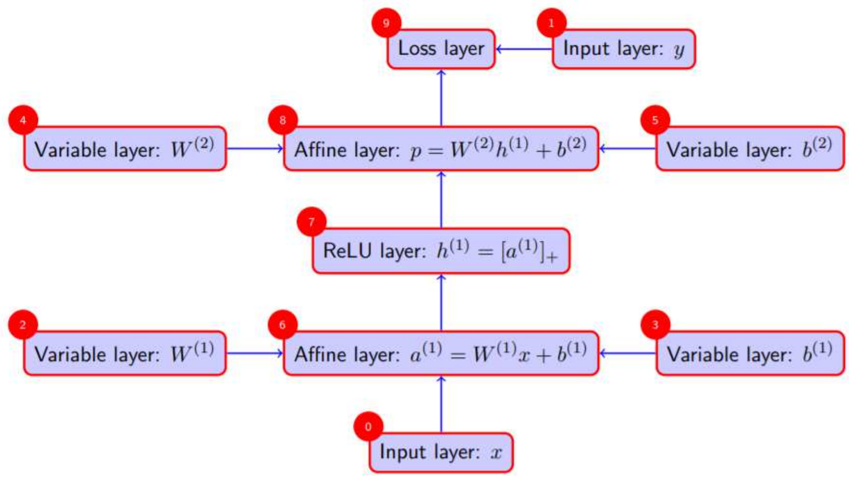


3. שכבה בינארית - $f:\mathbb{R}^2\to\mathbb{R}\ ,\ \forall i\, o_i = f(x_i, y_i)$ כאשר f לדוגמה היא:
   (א) חיבור - $f(x,y) = x+y$
   (ב) $hinge\,loss$ - $f(x,y) = [1-y_i x_i]_+$
   (ג) $logistic\,loss$ - $f(x,y) = log(1+exp(-y_i x_i))$

עד כה עסקנו ב-$Backpropagation$ של $f:\mathbb{R}^n\to\mathbb{R}$. נוכל להכליל זאת לפונקציות $f:\mathbb{R}^n\to\mathbb{R}^m$ על ידי כך ש-$delta$ מה-$pseudocode$ יהיה וקטור כך שכמות הקואורדינטות בו שווה לכמות ב-V_T (שכבת ה-$output$). בנוסף $derivative$ איננו

נגזרת אחת אלא מטריצת היעקוביאן $J_x\left(f\right)$ כאשר במיקום ה־i,j נמצאת הנגזרת החלקית של $f_i:\mathbb{R}^n\rightarrow\mathbb{R}$ לפי המשתנה ה־j והכפל הינו כפל מטריצות. בכך האלגוריתם למעשה מקיים את $multivariate\, chain\, rule$:
$J_w\left(f\circ g\right)=J_{g(w)}\left(f\right)J_w\left(g\right)$ (במיקום זה ה־$subscript$ איננו מייצגת את הקבועים כמו בכתיב של הגרף אלא את המשתנה שלפיו מתבצעת הגזירה).

דוגמה נוספת עבור $computation\, graph$ מורכב יותר הינו הגרף הבא. בגרף זה מבצעים רגרסיה לינארית (השכבה האפינית התחתונה), מעבירים בשכבת $ReLU$ המאפסת ערכים שליליים ואז מבצעים רגרסיה לינארית נוספת.



דוגמה־מנחה ־ נתבונן בבעיית הלמידה של זיהוי ספרות אזי לתמונות שחור לבן בגודל 28×28 פיקסלים:

$$\mathcal{X}=\{0,...,255\}^{28\times28},\ \ \mathcal{Y}=\{0,...,9\}$$

כלומר בעיה זו היא של $multiclassification$. ניקח היפותזה $h:\mathcal{X}\rightarrow\mathbb{R}^{|\mathcal{Y}|}$ כאשר $h\left(x\right)$ הינו וקטור ציונים לכל התיוגים. למשל אם נשתמש ברגרסיה לוגיסטית לשם כך נוכל לייצג זאת כהסתברות לכל אחד מהתיוגים. התיוג שנבחר הינו $\underset{i}{argmax}h_i\left(x\right)$.

- ארכיטקטורת רשת שטובה לבעיה ־ $x\rightarrow Affine\left(500\right)\rightarrow ReLU\rightarrow Affine\left(10\right)$ ־ כלומר עבור תמונה $x\in\mathcal{X}$ נעשה רגרסיה לינארית (עם למשל 500 $nodes$ ב־$hidden\, layer$ ראשונה), לאחר מכן נעביר בשכבת $ReLU$ להשארת רק ערכים $\mathbb{R}_{\geq0}$ ואז נבצע רגרסיה שנייה עם 10 $nodes$ ־ אחד לכל ספרה/תיוג.
- שימוש ב־$logistic\, loss$:
  - $SoftMax$ ־ $\forall i\ p_i=\frac{exp(h_i(x))}{\sum\limits_j exp(h_j(x))}$
  - $LogLoss$ ־ עבור y התיוג הנכון ־ $log\left(p_y\right)=log\left(\sum\limits_j exp\left(h_j\left(x\right)-h_i\left(x\right)\right)\right)$

24.2 $Expressiveness\, and\, Sample\, Complexity$

נראה כי כוח הביטוי של רשתות נוירונים הוא רב. נוכל לבטא בעזרת רשתות נוירונים קבוצות של "כל הפונקציות מ־A ל־B". נשים לב אבל שביטוי שכזה משמעותו גודל אקספוננציאלי של הרשת כתלות בגודל הקלט.

טענה 24.1 כל פונקציה בוליאנית $f:\{\pm1\}^n\rightarrow\{\pm1\}$ ניתנת לביטוי על ידי רשת נוירונים בעלת שכבה $hidden$ יחידה.

משפט 24.2 לכל $n\in\mathbb{N}$ יהי $s(n)$ המספר הטבעי המינימלי כך שקיימת רשת נוירונים בעלת שכבה $hidden$ יחידה עם $s(n)$ נוירונים אשר מממשת את כל הפונקציות מ־$\{0,1\}^n$ ל־$\{0,1\}$. אזי $s(n)$ אקספוננציאלי ב־n. **הוכחה:** יהי $G=\langle V,E\rangle$ הגרף המתאים לרשת הנוירונים $\mathcal{H}_{V,E,sign}$ המממשת את כל הפונקציות מ־$\{0,1\}^n$ ל־$\{0,1\}$. נראה כי בהינתן $m=2^n$ וקטורים מעל $\{0,1\}^n$ נוכל לנתץ סט זה. מכאן כי $VCdim\left(\mathcal{H}_{V,E,sign}\right)\geq2^n$. מן הצד השני $VCdim\left(\mathcal{H}_{V,E,sign}\right)$ חסום על ידי $O\left(|E|\, log\,|E|\right)\leq O\left(|V|^3\right)$ (משפט בספר בעמוד 274). על כן $|V|\geq\Omega\left(2^{n/3}\right)$ ועל כן מספר הנוירונים בשכבה ה־$hidden$ אכן אקספוננציאלי ב־n כנדרש. ■

משפט 24.3 יהי $\epsilon\in(0,1)$ ותהי σ פונקציה סיגמואידית כלשהי. לכל $n\in\mathbb{N}$ יהי $s(n)$ המספר הטבעי המינימלי כך שקיימת רשת נוירונים בעלת $|V|=s(n)$ אשר מחלקת ההיפותזות $\mathcal{H}_{V,E,\sigma}$ יכולה לקרב בדיוק ϵ כל פונקציה 1־ליפשיצית $f:[-1,1]^n\rightarrow[-1,1]$. אזי $s(n)$ אקספוננציאלי ב־n.

נראה כי פונקציות הניתנות לחישוב בזמן פולינומיאלי ניתנות לביטוי על ידי רשת בגודל פולינומיאלי:

משפט 24.4 תהי $T:\mathbb{N}\rightarrow\mathbb{N}$ ולכל $n\in\mathbb{N}$ תהי $\mathcal{F}_n$ סט הפונקציות הניתנות לחישוב על ידי מכונת טיורינג בזמן ריצה $T(n)$ אזי קיימים קבועים $b,c\in\mathbb{R}_+$ כך שלכל n קיים גרף מגודל לכל היותר $cT(n)^2+b$ כך שהרשת $\mathcal{H}_{V_n,E_n,sign}$ מממשת את $\mathcal{F}_n$.

24.3 $Tricks$

1. <u>נרמול ה־$data$</u> ־ למשל עבור הדוגמה המנחה בתחילת הפרק לחלק את כל הערכים ב־255.
2. <u>נקודת התחלה</u> ־ היות והבעיה איננה קמורה נקודת ההתחלה חשובה (כאשר קמור אזי מינימום מקומי הינו גלובלי, בבעיה לא קמורה דבר לא מובטח) פתרון שעובד טוב בפועל הוא לאתחל את שורות W (מטריצת המשקלים של השכבה האפינית) באופן רנדומלי בטווח $\left[-\frac{1}{\sqrt{n}},\frac{1}{\sqrt{n}}\right]$.
3. <u>$Mini-batches$</u> ־ בכל שלב של SGD נמצע את האיבוד על סמך $k>1$ דוגמאות רנדומליות. בכך נוריד את ה־$variance$ בעדכון כיוון ההתקדמות ונוכל לממש חלק זה ב־$parallel$ ועל כן לא יעלה הרבה זמן ריצה.
4. <u>$Learning\, rate$</u> ־ נתחיל בערך η כלשהו ונרד כל פעם ב־$\frac{1}{2}$ כאשר האימון מפסיק להתקדם.
5. <u>אלגוריתם SGD</u> ־ קיימים וריאנטים רבים של האלגוריתם למטרות שונות. נבחר בהתאם למה שנחוץ/טוב לבעיה.

דוגמה ל-$pseudocode$ של SGD עבור רשתות נוירונים (על $backwards\, propagation$):

SGD for Neural Networks

parameters:
number of iterations τ
step size sequence $\eta_1, \eta_2, \ldots, \eta_\tau$
regularization parameter $\lambda > 0$
input:
layered graph (V, E)
differentiable activation function $\sigma : \mathbb{R} \to \mathbb{R}$
initialize:
choose $\mathbf{w}^{(1)} \in \mathbb{R}^{|E|}$ at random
(from a distribution s.t. $\mathbf{w}^{(1)}$ is close enough to $\mathbf{0}$)
for $i = 1, 2, \ldots, \tau$
sample $(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}$
calculate gradient $\mathbf{v}_i = \texttt{backpropagation}(\mathbf{x}, \mathbf{y}, \mathbf{w}, (V, E), \sigma)$
update $\mathbf{w}^{(i+1)} = \mathbf{w}^{(i)} - \eta_i(\mathbf{v}_i + \lambda \mathbf{w}^{(i)})$
output:
$\bar{\mathbf{w}}$ is the best performing $\mathbf{w}^{(i)}$ on a validation set

Backpropagation

input:
example $(\mathbf{x}, \mathbf{y})$, weight vector $\mathbf{w}$, layered graph (V, E),
activation function $\sigma : \mathbb{R} \to \mathbb{R}$
initialize:
denote layers of the graph $V_0, \ldots, V_T$ where $V_t = \{v_{t,1}, \ldots, v_{t,k_t}\}$
define $W_{t,i,j}$ as the weight of $(v_{t,j}, v_{t+1,i})$
(where we set $W_{t,i,j} = 0$ if $(v_{t,j}, v_{t+1,i}) \notin E$)
forward:
set $\mathbf{o}_0 = \mathbf{x}$
for $t = 1, \ldots, T$
for $i = 1, \ldots, k_t$
set $a_{t,i} = \sum_{j=1}^{k_{t-1}} W_{t-1,i,j}\, o_{t-1,j}$
set $o_{t,i} = \sigma(a_{t,i})$
backward:
set $\boldsymbol{\delta}_T = \mathbf{o}_T - \mathbf{y}$
for $t = T-1, T-2, \ldots, 1$
for $i = 1, \ldots, k_t$
$\delta_{t,i} = \sum_{j=1}^{k_{t+1}} W_{t,j,i}\, \delta_{t+1,j}\, \sigma'(a_{t+1,j})$
output:
foreach edge $(v_{t-1,j}, v_{t,i}) \in E$
set the partial derivative to $\delta_{t,i}\, \sigma'(a_{t,i})\, o_{t-1,j}$

24.4 $Convolutional\, Networks$

לא נלמד בשלב זה

$Reinforcement\ Learning$ 25

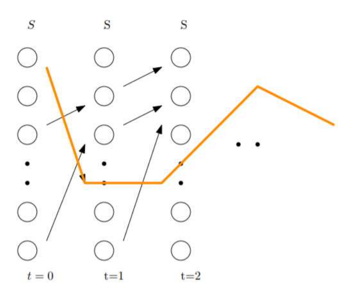
25.1 הקדמה

$reinforcement\,learning$ הינה שיטה חישובית להבנה ואוטומטיזציה של למידה מבוססת $goal$ ו־$decision\,making$.
בשיטה זו קיימים מצבים ($situations$) הממפים לפעולות שונות ($actions$), כאשר לבחירת פעולה מסוימת יש $reward/cost$.
ה־$agent$ (מקביל ל־$learner$ בטרמינולוגיה של $supervised\,learning$) לא מקבל כידע מוקדם איזה פעולות כדאי לו לבחור ועליו ללמוד איזה פעולות יניבו את ה־$reward$ המקסימלי. למידה זו שונה מ־$supervised\,learning$.
ב־$supervised\,learning$ לומדים על בסיס סט דגימות מתוייג על ידי $supervisor$ כלשהו ועל הלומד ללמוד על פי איזה $features$ וערכי $features$ מתקבל התיוג הנכון. ב־$reinforcement\,learning$ על ה־$agent$ להצליח ללמוד איזה פעולות לבצע (אשר יניבו $reward$ מקסימלי) על מצבים שאינם בסט האימון. כלומר על ה־$agent$ ללמוד מ"ניסיונו האישי" על הבעיה תוך כדי אינטראקציה עם הסביבה. מכל צורות הלמידה, $reinforcement\,learning$ הינה הדומה ביותר לתהליכי הלמידה שקורים ביצורים חיים ורבים מהאלגוריתמים המרכזיים בשיטה זו הינם בהשראת מערכות למידה ביולוגיות.

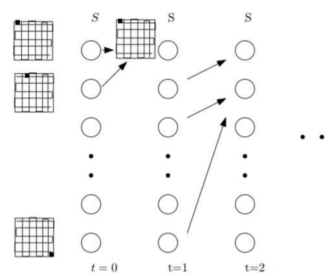
בנוסף לסביבת הבעיה ($environment$) וה־$agent$ קיימים מספר אלמנטים מרכזיים ב־$reinforcement\,learning$:

- פוליסה ($policy$) ־ ציינו כי בלמידה זו אנו לומדים מיפוי אופטימלי בין מצבים שונים לפעולות שונות. מכאן כי $policy$ מגדירה ל־$agent$ כיצד להתנהג בכל רגע נתון. למעשה פונקציה $\pi : S \to A$ כאשר S מתאר את המצבים ו־A את הפעולות. נבחין כי פונקציה זו איננה חייבת להיות דטרמינימסטית ונוכל להגדירה גם בתור $\pi : S \to 2^A$ כאשר עבור מצב נתון $s \in S$ קיימת הסתברות שונה להחזרת $a \in A$ שונים.
- חיזוקים ($reward\,signal$) ־ מגדירים את המטרה ($goal$) שבלמידה. בכל שלב ה־$agent$ מקבל מהסביבה $reward$ בהתאם למצבים/פעולות שה־$agent$ נמצא בהם. מטרתו אם כך למקסם את ה־$reward$ בטווח הארוך. זוהי הדרך שלנו לומר ל־$agent$ מה אנחנו רוצים להשיג (ולא איך, את זה הוא מגלה לבד). על כן ה־$reward$ מאפשר ל־$agent$ ללמוד איזה פעולות היו טובות ואיזה לא. כמו בפוליסה, גם כאן ה־$reward$ יכול להיות סטוכסטי.
- פונקציית ה־$value$ ($value\,function$) ־ מגדירה מה למעשה טוב לבצע בטווח הארוך. באופן גס, מייצגת את כמות הרווח הכוללת שה־$agent$ יכול לקבל בעתיד. באופן כללי נחפש לבצע $actions$ אשר יניבו את ה־$value$ המקסימלי ולא בהכרח ה־$reward$ המקסימלי.

כלומר, באופן כללי, ב־$reinforcement\,learning$ בכל איטרציה ה־$agent$ עושה פעולה, הסביבה משתנה בהתאם וה־$agent$ מקבל איזשהו $reward$ בהתאם:



דוגמה־מנחה: נדמיין הליכה במבוך כאשר אנו מנסים למצוא מסלול שיאפשר הגעה לצד השני. בכל שלב נתקדם במבוך ונפעל בהתאם. בדוגמה זו מתקיים כי ה־$State\, space$ הם משבצות המבוך וה־$Action\, space$ ־ הכיוונים שיכול ה־$agent$ ללכת. כאשר נתקדם במבוך נראה זאת בתור:



כדי להבין את האופי האיטרטיבי שבו למעשה לא "סיימנו להתאמן" נניח כי מצאנו מסלול אשר מוציא אותנו מהמבוך אבל האם ישנם מסלולים אחרים שמובילים לכך והאם מסלולים אלה מעניקים $reward$ גבוה יותר?

ב־$reinforcement\, learning$ המטרה ($goal$) מגולמת בקבלת ה־$reward$ מספרי כלשהו בצעדים השונים וה־$goal$ הוא מיקסום הסכום של ה־$expected\, rewards$. נוכל לחשוב על שיטות שונות לעשות זאת :

- $\underline{Total\, award}$ ־ סכימת $reward$ כללית $\mathbb{E}\left[\sum_{t=0}^{\infty} r_t\right]$. הבעיה בשיטה זו היא כי יתכן והסכום הינו אינסופי.
- $\underline{Finite\, horizon}$ ־ סכימת $reward$ במסגרת $H \in \mathbb{N}$ הצעדים הראשונים: $\mathbb{E}\left[\sum_{t=0}^{H} r_t\right]$.
- $\underline{Discounted\, reward}$ ־ סכימת $reward$ עם משקולת תלויה במספר הצעד $\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$ עבור $\gamma \in [0,1]$ המכונה גם ה־$discount\, rate$ (נשים לב שעבור $\gamma = 1$ נקבל את $Total\, award$) . שיטה זו נוחה שכן בשונה מ־$total\, award$ פוליסה זו מוגדרת היטב וכן היא מובילה להעדפה של $immdediate\, reward$

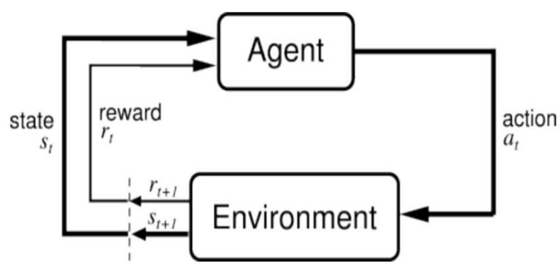
כאשר בונים $agents$ ישנן מספר דילמות שונות: האם אנחנו מעוניינים בתוצאות קצרות טווח ($local\, reward$) או ארוכות טווח ($future\, reward$); האם אנחנו רוצים שהתהליך יגלה מסלולים חדשים ($exploration$) או יחזיר את התוצאה הכי טובה על מה שכבר מכיר ($exploitation$); ובנוסף איך נדע לקשר בין רווח לבין פעולה שהתרחשה בעבר אשר כעת זיכתה אותנו באותו הרווח ($delay$).

$Supervised\, vs.\, Reinforcement$

	$Supervised\, learning$	$Reinforcement\, learning$
$Feedback$ מהסביבה	תשובה נכונה ($ground\, truth$)	משוב חלקי ($train-and-error$)
קלט ל־$learner$	סט $samples$ בלתי תלויות	מסלולים (קבוצת המצבים, הפעולות וה־$policy$)
פלט	החזרת היפותזה. מיפוי בין דוגמאות ותיוגים	החזרת $policy$. מיפוי בין מצבים ופעולות
מטרה	$min\ loss$	$min\ total\ cost$
השפעות ארוכות טווח (כלומר האם הדוגמאות הבאות יושפעו מהבחירות שבמצעים)	אין	בחירת פעולות תביא למצבים שונים

25.2 $Finite\, Markov\, Decision\, Processes\, (MDP)$

$Markov\, decision\, processes$ הינם פורמליזציה של תהליך סידרתי של קבלת החלטות, כאשר החלטה שהתקבלה בשלב מסויים עשויה להשפיע על החלטות בנקודת זמן מרוחקת יותר. במקרה של $reinforcement\, learning$ ה־$agent$ מתקשר עם הסביבה באיטרציות $t=0,1,2,3...$ כאשר בכל איטרציה t ה־$agent$ מחזיר מצב של הסביבה $S_t\in\mathcal{S}$ אשר על בסיס מצב זה בוחר לבצע פעולה $A_t\in\mathcal{A}$. בהתאם לכך מקבל ה־$agent$ את ה$reward$ $R_{t+1}\in\mathcal{R}\subset\mathbb{R}$ ומוצא עצמו במצב חדש $S_{t+1}\in\mathcal{S}$. כלומר מסלול ($trajectory$) MDP של ה־$agent$ הוא סדרה סופית: $S_0,A_0,R_1,S_1,A_1,...$.



באופן פורמלי ההגדרה היא:

הגדרה 25.1 MDP הינו רביעיה $\langle\mathcal{S},\mathcal{A},\tau,r\rangle$ כאשר:

1. $\mathcal{S}$ קבוצת מצבים סופית
2. $\mathcal{A}$ קבוצת פעולות סופית
3. $\tau:\mathcal{S}\times\mathcal{A}\to[0,1]^{|\mathcal{S}|}$ פונקציית הסתברות המעברים. למעשה $\tau(s,a)$ הינה פונקציית המעברים שנותנת הסתברות לכל מצב $s'\in\mathcal{S}$ אם נמצאים במצב $s\in\mathcal{S}$ ומבצעים פעולה $a\in\mathcal{A}$.
4. $r:\mathcal{S}\times\mathcal{A}\to\mathbb{R}$ פונקציית ה־$reward$.

25.2.1 $The\, Markovian\, Assumption$

עבור MDP סופי נתייחס ל־R_t, S_t כמשתנים מקריים בדידים בעלי הסתברות לקבלם בהתאם ל־S_{t-1} ול־A_{t-1} ומכאן:

הגדרה 25.2 ($The\, Markovian\, Assumption$) ההסתברות בהינתן MDP סופי לקבלת המצב וה־$rewward$ הבאים תלוי רק במצב הנוכחי ולא המצבים המוקדמים שהובילו למצב הנוכחי. כלומר:

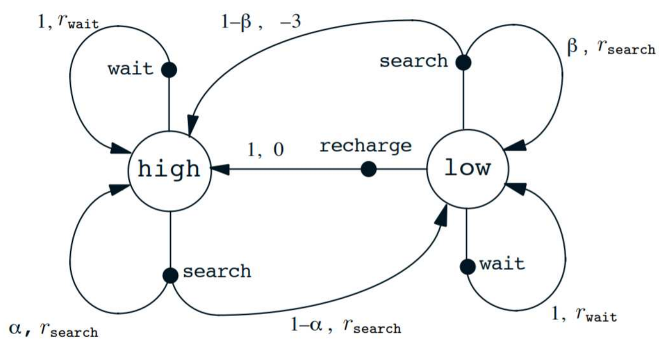
$$State\, Transition$$
$$\mathbb{P}\left(S_{t+1}=s'\,|\,S_t=s\,\wedge\,A_t=a\,\wedge\,R_t=r_t\,\wedge\,S_{t-1}=s_{t-1}\,\wedge\,...\right)=\mathbb{P}\left(S_{t+1}=s'\,|\,S_t=s,\,A_t=a\right)$$

$$Expected\, Reward$$
$$\mathbb{E}\left[R_{t+1}\,|\,S_t=s\,\wedge\,A_t=a\,\wedge\,R_t=r_t\,\wedge\,S_{t-1}=s_{t-1}\,\wedge\,...\right]=\mathbb{E}\left[R_{t+1}\,|\,S_t=s\,\wedge\,A_t=a\right]$$

נסמן זאת ב:

$$p\left(s'|s,a\right)=\mathbb{P}\left(S_{t+1}=s'\,|\,S_t=s,\,A_t=a\right)$$
$$r\left(s,a\right)=\mathbb{E}\left[R_{t+1}\,|\,S_t=s\,\wedge\,A_t=a\right]$$

נשים לב כי במקרים רבים הנחה זו הינה ראלית. באופן תאורתי נוכל להגדיר מצב כתמונת כל ההיסטוריה כולה (אשר עלולה להיות גדולה מאוד) או כסט של מספר מצומצם של הפעולות האחרונות שהתבצעו. דרך נוחה להצגת הדינמיות של ה־MDP הינה $transition\, graph$ (אוטומט):



25.2.2 $Policies\, and\, Value\, functions$

מרבית אלגוריתמי ה־$reinforcement\, learning$ כוללים שיערוך של $value\, functions$ המעריכים כמה טוב (=$reward$ עתידי צפוי) ל־$agent$ להיות במצב נתון. $reward$ עתידי זה תלוי בפעולות שיבחר ה־$agent$ לבצע, כלומר ב־$policy$. באופן פורמלי:

הגדרה 25.3 $policy$ הינה פונקציית מיפוי בין מצבים לבין ההסתברות לביצוע פעולות מסוימות. אם ה־$agent$ פועל לפי פוליסה π בצעד t אזי $\pi\left(s|t\right)$ הינה ההסתברות ש־$A_t=a$ אם $S_t=s$

הגדרה 25.4 עבור MDP ופוליסה π נעריך את הפוליסה על ה־MDP באופן הבא:

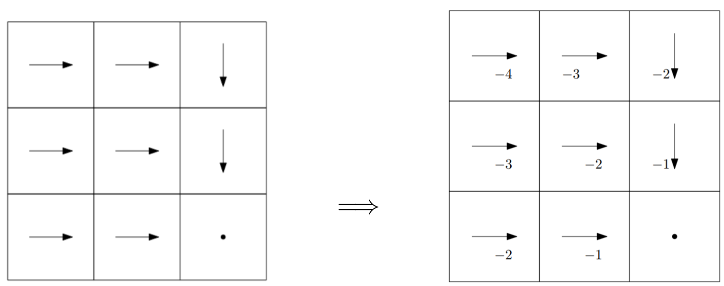
- $Total\, reward$ ־ $\underset{s_1,s_2,\ldots}{\mathbb{E}}\left[\sum_{t=0}^{\infty} r\left(s_t,\pi\left(s_t\right)\right)\right]$
- $Discounted\, reward$ ־ $\underset{s_1,s_2,\ldots}{\mathbb{E}}\left[\sum_{t=0}^{\infty} \gamma^t r\left(s_t,\pi\left(s_t\right)\right)\right]$

הגדרה 25.5 ה־$value$ של מצב $s\in\mathcal{S}$ תחת פוליסה π, אשר נסמן $v_\pi\left(s\right)$, הינו ה־$expected\, return$ כאשר נתחיל ב־s ונפעל על פי π. עבור MDP נתון נגדיר זאת:

$$v_\pi\left(s\right)=\underset{s_1,s_2,\ldots}{\mathbb{E}}\left[\sum_{t=0}^{\infty}\gamma^t r\left(s_t,\pi\left(s_t\right)\right)\mid s_0=s\right]$$

נקרא ל־v_π ה־$state-value\, function$ עבור פוליסה π.

דוגמה־מנחה ־ חזרה למבוך. נניח שברשותנו פונקציית מעברים דטרמיניסטית ופוליסה. נניח שה־$reward$ הוא -1 לכל המשבצות למעט המשבצת הימנית תחתונה ונניח כי $\gamma=1$ אזי:



טענה 25.6 לכל פוליסה $\pi:\mathcal{S}\to\mathcal{A}$ ומצב $s\in\mathcal{S}$ מתקיים כי $v_\pi\left(s\right)=r\left(s,\pi\left(s\right)\right)+\gamma\underset{s'\sim\tau\left(s,\pi\left(s\right)\right)}{\mathbb{E}}\left[v_\pi\left(s'\right)\right]$

הוכחה: נפתח לפי הגדרה ונקבל:

$$\begin{aligned}
v_\pi\left(s\right) &= \underset{s_1,s_2,\ldots}{\mathbb{E}}\left[\sum_{t=0}^{\infty}\gamma^t r\left(s_t,\pi\left(s_t\right)\right)\mid s_0=s\right]\\
&= r\left(s,\pi\left(s\right)\right)+\underset{s_1,s_2,\ldots}{\mathbb{E}}\left[\sum_{t=1}^{\infty}\gamma^t r\left(s_t,\pi\left(s_t\right)\right)\mid s_0=s\right]\\
&= r\left(s,\pi\left(s\right)\right)+\sum_{s'\in\mathcal{S}}p\left(s'\mid s,\pi\left(s\right)\right)\underset{s_2,\ldots}{\mathbb{E}}\left[\sum_{t=1}^{\infty}\gamma^t r\left(s_t,\pi\left(s_t\right)\right)\mid s_0=s\right]\\
&= r\left(s,\pi\left(s\right)\right)+\gamma\sum_{s'\in\mathcal{S}}p\left(s'\mid s,\pi\left(s\right)\right)v_\pi\left(s'\right)
\end{aligned}$$

$\blacksquare$

באופן דומה נוכל להגדיר זאת עבור בחירת פעולה מסוימת:

הגדרה 25.7 ה־$value$ של בחירת פעולה $a \in \mathcal{A}$ במצב $s \in \mathcal{S}$ תחת הפוליסה π, אשר נסמן $q_\pi(s,a)$, הינו ה־$expected\, return$ כאשר נתחיל ב־s, נבצע a ונפעל לפי π. עבור MDP נתון נגדיר זאת:

$$q_\pi(s,a) = \underset{s_1,s_2,\ldots}{\mathbb{E}}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t) \mid s_0 = s \ \wedge\ a_0 = a \ \wedge\ \forall t>0\, a_t = \pi(s_t)\right]$$

נקרא ל־q_π ה־$action - value\, function$ עבור פוליסה π.

היחס בין v_π ו־q_π

- v_π כפונקציה של q_π: $\forall s \in \mathcal{S}\ \ v_\pi(s) = q_\pi(s,\pi(s))$
- q_π כפונקציה של v_π: $\forall s \in \mathcal{S}, a \in \mathcal{A}\ \ q_\pi(s,a) = r(s,a) + \gamma \underset{s'\sim\tau(s,\pi(s))}{\mathbb{E}}[v_\pi(s')]$

25.2.3 פוליסות ופונקציות ערך אופטימליות

באופן כללי פתרון בעיית $reinforcement\, learning$ הינה מציאת $policy$ המניבה $reward$ גבוה בטווח הארוך. עבור MDP נוכל להגדיר זאת באופן הדוק.

הגדרה 25.8 עבור פוליסות $\pi,\pi' : \mathcal{S} \to \mathcal{A}$ נאמר כי π טובה יותר מ־π', $\pi \geq \pi'$ אם ורק אם $\forall s \in \mathcal{S}\ v_\pi(s) \geq v_{\pi'}(s)$.

הגדרה 25.9 נגדיר את הפונקציות ערך, פעולה־ערך ופוליסה האופטימליות באופן הבא:

$$\begin{aligned}
&Opt.\, value\, function: && v_*(s) = \underset{\pi}{max} v_\pi(s)\\
&Opt.\, action - value\, function: && q_*(s,a) = \underset{\pi}{max} q_\pi(s,a)\\
&Optimal\, policy: && \pi_*(s) = \underset{\pi}{argmax} v_\pi(s)
\end{aligned}$$

משפט 25.10 לכל MDP מתקיים:

- קיימת π_* אשר טובה או שווה לכל הפוליסות האחרות: $\forall \pi\ \pi_* \geq \pi$.
- כל הפוליסות האופטימליות משיגות את ה־$optimal\, value\, function$: $\forall \pi_*\ \ v_{\pi_*}(s) = v_*(s)$.
- על הפוליסות האופטימליות משיגות את ה־$optimal\, action - value\, function$: $\forall \pi_*\ \ q_{\pi_*}(s,a) = q_*(s,a)$.

היחס בין v_* ו־q_*

- v_* כפונקציה של q_*: $\forall s\in\mathcal{S}\ \ v_*(s)=\underset{a\in\mathcal{A}}{max}q_*(s,a)$.
- q_π כפונקציה של v_π:

$$\begin{aligned} q_*(s,a) &= q_{\pi_*}(s,a)\\ &= r(s,a)+\gamma\underset{s'\sim p(\cdot|s,\pi(s))}{\mathbb{E}}\left[v_{\pi_*}(s')\right]\\ &= r(s,a)+\gamma\underset{s'\sim p(\cdot|s,\pi(s))}{\mathbb{E}}\left[v_*(s')\right]\end{aligned}$$

- π_* כפונקציה של q_*: $\pi_*(s)=\underset{a\in\mathcal{A}}{argmax}q_*(s,a)$.

מסקנה 25.11 הפוליסה האופטימלית π_* הינה דטרמיניסטית.

משפט 25.12 ($Bellman's\ Equation$) הפוליסה π הינה אופטימלית אם ורק אם מקיימת את משוואת האופטימליות של $Bellman$: $v_*(s)=\underset{a\in\mathcal{A}}{max}\left[r(s,a)+\gamma\underset{s'\sim p(\cdot|s,\pi(s))}{\mathbb{E}}\left[v_*(s')\right]\right]$.

הוכחה:

$$\begin{aligned} v_*(s) &= \underset{a\in\mathcal{A}}{max}q_*(s,a)\\ &= \underset{a\in\mathcal{A}}{max}q_{\pi_*}(s,a)\\ &= \underset{a\in\mathcal{A}}{max}\left[r(s,a)+\gamma\underset{s'\sim p(\cdot|s,\pi(s))}{\mathbb{E}}\left[v_*(s')\right]\right]\end{aligned}$$

■

25.3 אלגוריתמים ל־MDP

כאשר אנו באים לפתור בעיות $reinforcement\,learning$ אנו מניחים כי ה־MDP סופי. כלומר $\mathcal{S},\mathcal{A},\mathcal{R}$ קבוצות סופיות וכי הדינמיקה שלהם ניתנת לנו על ידי ההסתברויות של $p(s',r\mid s,a)$ כפי שכבר ראינו. במקרים רבים פתרון בעיית הלמידה נעשה בעזרת תכנון דינמי ($Dynamic\,Programming$) על מנת למדל את המבנה והחיפוש של פוליסות טובות. ואכן, באופן כללי כאשר נרצה לפתור MDP נשתמש ב־$value\,functions$ כדי למדל את מבנה החיפוש ולמציאת הפוליסה האופטימלית.

25.3.1 $Policy\,Evaluation$

בחלק זה נראה כיצד לחשב את פונקציית ה־$state-value$: v_π. בהינתן קלט פונקציית הסתברות מעברים $p:\mathcal{S}\times\mathcal{A}\times\mathcal{S}\to[0,1]$ (עבור מצב, הפעולה שנבצע במצב והמצב אליו נגיע בעקבות כך), פונקציה לצפי $reward$ $r:\mathcal{S}\times\mathcal{A}\to\mathbb{R}$ ופונקציית פוליסה $\pi:\mathcal{S}\to\mathcal{A}$ נרצה להחזיר $v_\pi:\mathcal{S}\to\mathbb{R}$.

$$\begin{aligned}
&Initialize\ \forall s\in\mathcal{S}\ v_0(s)=0\\
&For\,t=1,2,....\\
&\forall s\in\mathcal{S}\ \ v_{t+1}(s)=\sum_{s'\in\mathcal{S}}p(s'\mid s,\pi(s))\cdot[r(s,\pi(s))+\gamma v_t(s')]
\end{aligned}$$

נבין כעת את החישוב שמבצעים. עבור $s\in\mathcal{S}$ נבדוק לכל $s'\in\mathcal{S}$ מהו הערך שמצב זה מוסיף לפתרון ונסכום באופן ממושקל, לפי ההסתברות של s' להתקיים (על פי המצב s והפעולה שנבצע ־ $(p(s'\mid s,\pi(s))$, את הערך הצפוי אילו נבחר בכיוון זה וערכו של מצב זה. אלגוריתם שכזה, אשר בכל איטרציה מעדכן את הערך הצפוי של $v_t(s)$ בסופו של דבר מתכנס לערך האמיתי של מצב זה, אולם היות ולא נוכל לתת לאלגוריתם לרוץ עבור $t\to\infty$ נרצה להחליט באיזה שלב נעצור. לשם כך הטענה הבאה (בכל איטרציה של האלגוריתם מצליחים לשפר את הערך אליו שואפים באופן אקספוננציאלי) וכן הפסאודו־קוד המלא אחרי הטענה.

טענה 25.13 עבור הרצת האלגוריתם $Policy\,Evaluation$ במשך t איטרציות מתקיים $||v_t-v_\pi||_\infty\le\gamma^t\,||v_0-v_\pi||_\infty$.

הוכחה: לכל $s\in\mathcal{S}$ מתקיים:

$$\begin{aligned}
|v_{t+1}(s)-v_\pi(s)| &= \left|\sum_{s'\in\mathcal{S}}p(s'\mid s,\pi(s))\cdot[r(s,\pi(s))+\gamma v_t(s')]\right|-\left|\sum_{s'\in\mathcal{S}}p(s'\mid s,\pi(s))\cdot[r(s,\pi(s))+\gamma v_\pi(s')]\right|\\
&= \gamma\left|\sum_{s;\in\mathcal{S}}p(s'\mid s,a)\cdot[v_t(s')-v_\pi(s')]\right|\\
&\le \gamma\sum_{s;\in\mathcal{S}}p(s'\mid s,a)\cdot|v_t(s')-v_\pi(s')|
\end{aligned}$$

■

Iterative policy evaluation

Input π, the policy to be evaluated
Initialize an array $V(s) = 0$, for all $s \in \mathcal{S}^+$
Repeat
  $\Delta \leftarrow 0$
  For each $s \in \mathcal{S}$:
    $v \leftarrow V(s)$
    $V(s) \leftarrow \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)\big[r + \gamma V(s')\big]$
    $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
until $\Delta < \theta$ (a small positive number)
Output $V \approx v_\pi$

(מכאן מעט הרחבה על סמך הספר) למעשה המטרה שבחישוב ה־$value\,function$ של הפוליסה היא כדי להצליח למצוא פוליסות טובות יותר. כעת אחרי שחישבנו את הערך הצפוי (כולל התוספת האיטרטיבית הצפויה) של בחירת $a \in \mathcal{A}$ עבור $s \in \mathcal{S}$ כלשהו נוכל לראות האם להחליף לפעולה אחרת ולקבל בסופו של דבר ערך גבוה יותר. לתהליך זה קוראים $policy\,improvement$ (אשר לא נכנסנו אליו בקורס) ונוכל להגדיר פוליסה אשר תפעל באופן חמדן ותבחר להתקדם לפי הערך האופטימלי של $q_\pi(s,a)$:

$$\begin{aligned}\pi'(s) &= \underset{a}{argmax}\, q_\pi(s,a)\\ &= \underset{a}{argmax}\,\mathbb{E}\left[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s,\ A_t = a\right]\\ &= \underset{a}{argmax} \sum_{s',r} p(s',r \mid s,a)\cdot\left[r + \gamma v_\pi(s')\right]\end{aligned}$$

לאחר השלב של $policy\,evaluation$ ניתן לעשות את ה־$policy\,improvement$ ולחזור על שתי הפעולות הללו עד אשר מגיעים ל־$policy$ אופטימלי.

25.3.2 $Value\,Iteration$

במקרה זה נרצה לקבל את המצבים והפעולות ולהיות מסוגלים למצוא מה ה־$policy$ האופטימלי. אלגוריתם ה־$value\,iteration$ פועל דומה ל־$policy\,iteration$ אך בכל איטרציה בוחר את ה־$action$ הממקסם את הרווח העתידי. נפעיל את האלגוריתם שוב ושוב ונתכנס לערך האופטימלי. בדומה ל־$policy\,iteration$ גם כאן קיימת טענה שמראה התקרבות אקספוננציאלית לערך האמיתי כתלות ב־t.

$$Initialize\ \forall s \in \mathcal{S}\ v_0(s) = 0$$

$$For\, t = 1,2,....$$

$$\forall s \in \mathcal{S}\ \ v_{t+1}(s) = \underset{a \in \mathcal{A}}{max} \sum_{s' \in \mathcal{S}} p(s' \mid s, \pi(s)) \cdot \left[r(s,\pi(s)) + \gamma v_t(s')\right]$$

Value iteration

Initialize array V arbitrarily (e.g., $V(s) = 0$ for all $s \in \mathcal{S}^+$)

Repeat
  $\Delta \leftarrow 0$
  For each $s \in \mathcal{S}$:
    $v \leftarrow V(s)$
    $V(s) \leftarrow \max_a \sum_{s',r} p(s', r | s, a)\big[r + \gamma V(s')\big]$
    $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
until $\Delta < \theta$ (a small positive number)

Output a deterministic policy, $\pi \approx \pi_*$, such that
  $\pi(s) = \operatorname{argmax}_a \sum_{s',r} p(s', r | s, a)\big[r + \gamma V(s')\big]$

הערה 25.14 נשים לב שב־$\Delta < \theta$ אנו בעצם מגדירים שהאלגוריתם ממשיך איטרטיבית כל עוד Δ לא התכנס (כלומר עד שהשיפור יהיה מספיק קטן, תחת ההנחה שמכאן התקרבנו מספיק לערך האמיתי).

הערה 25.15 להסבר יותר מפורט על ההבדל בין *value iteration* ו־*policy iteration* ניתן לקרוא את השאלת *stackoverflow* הבאה: What is the difference between value iteration and policy iteration? (או בקישור https://stackoverflow.com/q/37370015/6400526)

25.3.3 *Learning*

עד כה עסקנו בבעיית התכנון אשר בה ידענו מהם ה־*rewards* ופונקציית ה־*transition probability* ויכולנו להסיק מה הדרך המועדפת. כאשר איננו יודעים פונקציות אלה הקלט לאלגוריתם הלמידה הוא התנסויות. במקרה זה הם זוגות של *state, action* נוכחיים ושל תוצאה מביצוע הנוכחיים (הגדרה נוספת היא סדרה של זוגות עוקבים). ישנם שני סוגי למידה שכאלה:

- *On policy* ־ הפעולות מתבצעות על ידי ה־*agent* עצמו והוא שמחליט כיצד לפעול.
- *Off policy* ־ הגישה בה "מתבוננים מהצד". נותנים לפעולות להתבצע אך לא משפיעים עליהן, מודדים תוצאות ומסיקים מסקנות.

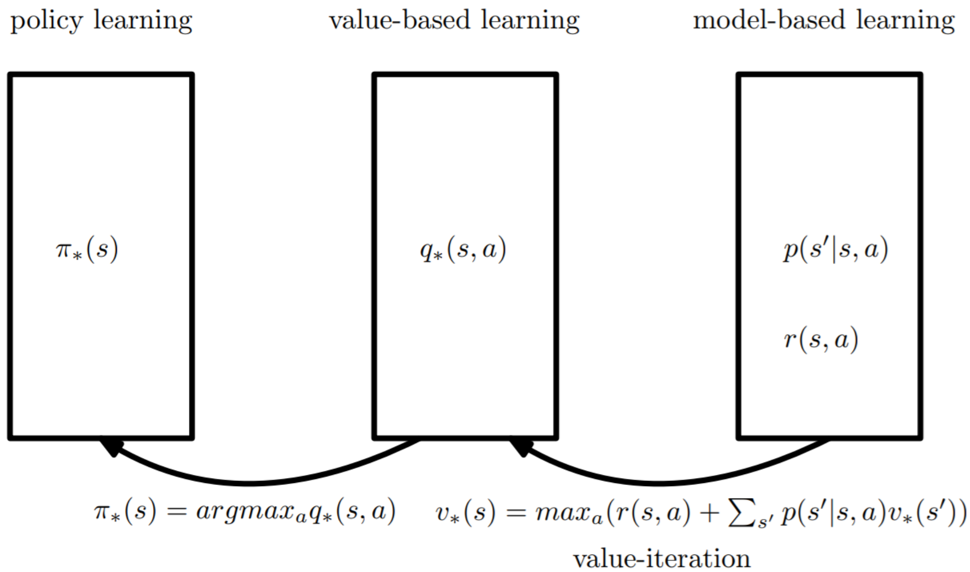
להבדלים בין השניים ניתן לקרוא כאן What is the difference between off-policy and on-policy learning? (או בקישור https://stats.stackexchange.com/q/184657)

כאשר נשאלת השאלה מה אנו רוצים ללמוד יש כמה גישות:

1. <u>לימוד מבוסס מודל</u> ־ נלמד תחילה את ה־*MDP*. נלמד את הרווחים ואת ה־*transition probabilities*. בהתאם לזאת נוכל ללמוד איך להתנהג אחר כך. מכאן כדי להסיק את ה־*value function* (בהינתן שכעת אנו יודעים את המודל) נוכל להשתמש ב־*value iteration* שראינו.
2. <u>לימוד מבוסס *value*</u> ־ ישנם אלגוריתמים ללמידת ה־$q - function$. בהינתן ה־$q - function$ נמצא את הפוליסה האופטימלית על ידי כך שניקח את ה־*actions* האופטימליים.

3. <u>לימוד מבוסס פוליסה</u> ־ אלגוריתמים ללמידה ישירה של ה־$policy$ האופטימלי.

אלגוריתמים של שני הסעיפים האחרונים האחרונים נקראים $model\, free\, algorithms$.



אלגוריתם $Q - Learning$

באלגוריתם הבא אנו מקבלים כקלט התנסויות ונעדכן את ה־$q\, function$ בהתאם. צורת העדכון, אשר דומה לצורה ב־GD, לוקחת את הערך הנוכחי עם תיקון קטן לפי ההפרש בין הערך החדש לקודם. אז זה נכפיל בפקטור η אשר משמש כ־$learning\, rate$. נשים לב כי את הערך החדש איננו יודעים (שכן איננו יודעים את פונקציית ההסתברות) ולכן נשערכו בהתאם לכך שניקח את הפעולה שמביאה למקסימום.

$$Initialize\ \forall s \in \mathcal{S},\ a \in \mathcal{A}\ q_0\left(s,a\right) = 0$$
$$For\ each\ experiene\ \left(s_t, a_t, r_t, s_{t+1}\right)$$
$$q_{t+1}\left(s_t,a_t\right) = q_t\left(s_t,a_t\right) - \eta_t\left(q_t\left(s_t,a_t\right) - \left(r_t + \gamma \underset{a}{max} q_t\left(s_{t+1},a\right)\right)\right)$$

משפט 25.16 אם כל (s,a) מתקיים $infinitly\, often$ ו־$\eta_t = \frac{1}{t}$ אזי $\lim\limits_{t\to\infty} q_t\left(s,a\right) = q_*\left(s,a\right)$.

נשים לב כי אלגוריתם זה הינו $off\ policy$ שכן קיבלנו התנסויות כלשהן ועדכנו מסקנות בהתאם.

אם נרצה לבחור אלגוריתם שהינו $on\ policy$ אשר בו נשפיע על איזה $actions$ יתבצעו, נתקל בדילמה שהזכרנו בתחילת הפרק של $exploration\ vs.\ exploitation$ ־ האם נרצה שהאלגוריתם יקח את ה־$policy$ הכי טוב (ה־$action$ שיביא

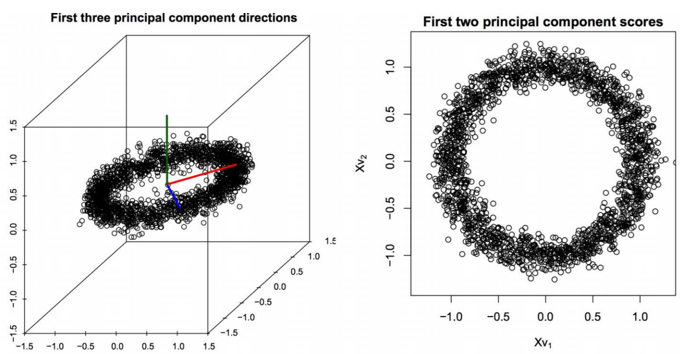
למקסימום) או לא ואולי נמצא כך מסלולים חדשים טובים יותר שלא ידענו עליהם. שיטה גנרית לעשות זאת ($\epsilon_t - greedy$) היא שעבור כל מצב בהסתברות $1 - \epsilon_t$ נבצע $exploitation$, כלומר נבחר את ה־$action$ הממקסם, ובהסתברות ϵ_t נבצע $exploration$, כלומר נבחר $action$ אקראי.

נשים לב שכאשר מספר המצבים גודל ההסתברות לעשות $exploration$ היא מאוד נמוכה. על כן נשלב בין $reinforcement\,learning$ ורשתות נוירונים.

26 $Dimensionality\,Reduction$

בפרק זה נעסוק בהתמודדות עם $feature\,space$ ממימד גבוה מדי. מעבר לקושי החישובי שבמספר $features$ גדול קשה לנו לחקור את ה־$data$ ולנתח את התוצאות. בשלב זה ראינו מספר דרכים כמו $lasso$ ו־$subset\,selection$ להוריד $features$.

במקרים רבים, על אף הריבוי ב־$features$, יש יתירות ($redundancy$) גבוהה ־ כלומר אין לנו צורך בכל ה־$features$ כדי להצליח לתייג את המידע. על כן יתכן ונוכל לקפל למימד נמוך יותר. תהליך ה־$dimensionality\,reduction$ לוקח $data$ ממימד גבוה (המרחב האמביינטי) וממפה אותו למרחב חדש, שמימדו קטן יותר (המרחב האינטרינזי), אשר עדין מגלם את אותו עולם ראשוני. במקרים רבים פעולת הורדת המימד הינה שלב מקדים ללמידה.



בתמונה זו אנו רואים $data$ שנמצא במרחב התלת מימדי וטרנספורמציה למרחב הדו מימדי שעדין מייצגת נכונה את ה־$data$. כלומר יכולנו להוריד את המימד הדגימות וזאת מבלי לאבד יותר מדי מהוריאביליות של ה־$data$.

26.1 $Linear\,Dimension\,Reduction\,and\,PCA$

ברוב המקרים, עבור $\mathcal{X}$ הנמצא ב־$\mathbb{R}^d$ מתקיים שבפועל $\mathcal{X} \subseteq \mathbb{R}^r \subset \mathbb{R}^d$. כלומר $\mathcal{X}$ הינו תת מרחב ממימד קטן יותר r. נרצה אפוא לחפש העתקה לינארית $w : \mathbb{R}^d \to \mathbb{R}^r$ כך שנוכל לייצג את המידע שלנו במימד נמוך יותר. העתקה זו היא למעשה $x \mapsto Wx$ כאשר $x \in \mathbb{R}^d$ ו־$Wx \in \mathbb{R}^r$. יהיו $x_1, ..., x_m \in \mathbb{R}^d$ דגימות אזי נחפש מטריצות $W \in \mathbb{R}^{r \times d}$ ו־$U \in \mathbb{R}^{d \times r}$ כך ש:

- $x \mapsto Wx$ הינה הטרנספורמציה כך ש־Wx הינו הייצוג של x במימד הנמוך יותר r.

- נסמן $y = Wx$ אזי נוכל לחשב את $\tilde{x} = Uy$ אשר הוא שיחזור מקורב של x במימד הגבוה d.

הגדרה 26.1 ב־$Principal\,Component\,Analysis\,(PCA)$ נחפש מטריצת כיווץ W ומטריצת שחזור U כל שסכום ריבועי המרחקים בין הוקטור המקורי והמשוחזר מינימלי:

$$\underset{W\in\mathbb{R}^{r\times d},\,U\in\mathbb{R}^{d\times r}}{argmin}\sum_{i=1}^{m}||x_i - UWx_i||_2^2$$

הערה 26.2 מטרת המטריצה U למעשה היא לספק כלי השוואה בין המרחב אליו הגענו לאחר הורדת המימד לבין המרחב המקורי.

הערה 26.3 נשים לב כי איננו יודעים את r האמיתי של ה־$data$ ועל כן בפועל אנו מחפשים למעשה k כלשהו שבתקווה הינו r.

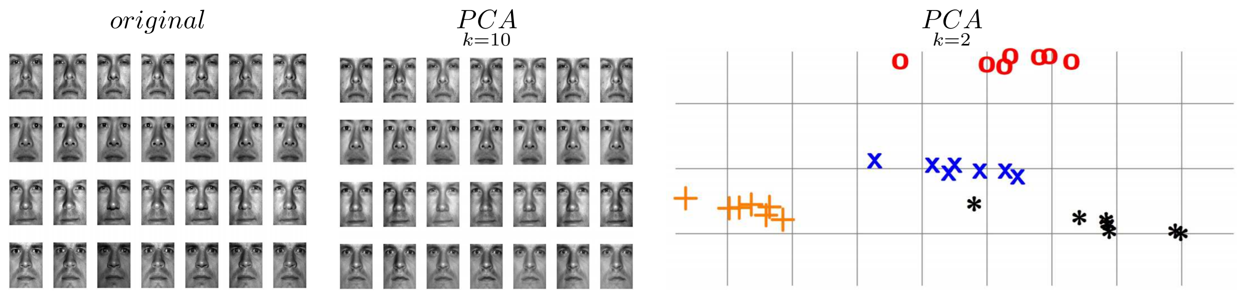
משפט 26.4 עבור הורדת מימד עם PCA ממימד d למימד k כלשהו ומטריצות $U \in \mathbb{R}^{d\times k}, W \in \mathbb{R}^{k\times d}$ כאשר עמודות U הן k הוקטורים העצמיים הגדולים של מטריצת ה־$covariance$ S אזי $W = U^T$

$$S = \frac{1}{m}\sum_{i=1}^{m}(x_i - \bar{x})(x_i - \bar{x})^T$$

נבחין כי במטריצה S (בנוסף למרכוז ־ ההחסרה של הערך הממוצע בכל נקודה) מתקיים שהיא מטריצה סימטרית עם ערכים עצמיים אי שליליים שעמודותיה בסיס אורתונורמלי לתת המרחב. למעשה עבור $\lambda_1, ..., \lambda_d \geq 0$ אם נניח בה"כ $\lambda_1 \geq ... \geq \lambda_d \geq 0$, מתקיים כי הוקטורים העצמיים השייכים לערכים העצמיים בעלי ערכים קטנים מייצגים מעט מאוד וריאביליות במידע. לעומת זאת הוקטורים העצמים השייכים לעצמיים עצמיים גדולים מייצגים וריאביליות גבוהה של המידע בכיוון של הוקטור העצמי. על כן אם ניקח את k הוקטורים העצמיים עם הערכים העצמיים הכי גדולים נקבל את תת המרחב ממימד k שמייצג הכי טוב את הוריאביליות של המידע המקורי ־ ובכל זאת במימד נמוך יותר.

למעשה עבור $u_1, ..., u_d$ הוקטורים העצמיים השייכים לערכים העצמיים $\lambda_1, ..., \lambda_d$ ועבור המטריצה X עם m דגימות ו־d $features$, כדי לקבל הורדת מימד למימד k ניקח את $Xu_1u_1^T, ...Xu_ku_k^T$ עבור k כרצוננו.

דוגמה־מנחה: עבור הבעיה של זיהוי פנים, בהינתן ה־$dataset$ של התמונות השונות, היות ואין באמת שוני רב בין התמונות, למעט האדם השונה, נוכל להפעיל PCA עם ערך k כלשהו (בדוגמה $k = 10$) ולמעשה לקבל $data$ בבמימד נמוך יותר שעדין מתאר טוב מספיק את ה־$data$ המקורי. נמשיך ונראה כי עבור $k = 2$, ההטלה של התמונות ממימד 50×50 למימד 2 מספקת לנו אזורים שונים שבכל אזור למעשה קיבלנו את אוסף הנקודות, התמונות, שמייצגות אדם נפרד.



27 $Clusters - k - means$

במשך הקורס עסקנו ב־$supervised\, learning$ שבו לדגימות שברשותנו יש תיוגים. במקרה של $unsupervised\, learning$ המידע איננו מכיל תיוגים. שתי דוגמאות לכך הן $clustering$ ו־$anomaly\, detection$. באופן אינטואיטיבי קליסטור ($clustering$) הינה הפעולה של קיבוץ המידע לסטים כך שפריטים בעלי אופי דומה יהיו באותה קבוצה, ופריטים בעלי אופי שונה יהיו בקבוצות שונות. לדוגמה תחת ביולוגיה חישובית ננסה לקלסטר קבוצות של גנים בהתבסס על דמיון ברמות הביטוי שלהם תחת ניסויים שונים. באופן יותר פורמלי עבור:

קלט: יהי $\mathcal{X}$ סט המידע ו־$d:\mathcal{X}\times\mathcal{X}\to\mathbb{R}_+$ פונקציית מרחק מעל $\mathcal{X}$ המקיימת סימטריות ו־$d(x,x)=0$ $\forall x\in\mathcal{X}$. לעיתים נדרוש שתקיים גם את אי שיוויון המשולש. לעיתים נגדיר את פונקציית המרחק להחזיר ערכים $[0,1]$ ו־$d(x,x)=1$. ישנם אלגוריתמים שגם יבקשו $k\in\mathbb{N}$ הקובע את מספר ה־$clusters$.

פלט: חלוקה של הדומיין $\mathcal{X}$. כלומר, $C=(C_1,...,C_k)\ \ s.t\ \bigcup_{i=1}^{k}C_i=\mathcal{X}$. נזכור כי בחלוקה הכוונה היא $\forall i\neq j\ C_i\cap C_j=\emptyset$. לעיתים ב־$"soft"\, clustering$ נחזיר וקטור הסתברויות לשייכות ל־$cluster$. כלומר לכל $x\in\mathcal{X}$ נחזיר $(p_1(x),...,p_k(x))\ \ s.t.\ \forall i\in[k]\ \ p_i(x)=\mathbb{P}(x\in C_i)$

האלגוריתמים השונים מגדירים פונקציה לחישוב מרחק בין $clusters$ (לדוגמה זוג הנקודות הכי קרובות בין השני $clusters$, הזוג הכי רחוק או ממוצע המרחקים) וכן פונקציית $cost$ עבור $cluster$. על כן נחפש חלוקה, $clustters$, שיביאו את העלות למינימום. כלומר $G:(\mathcal{X},d)\times C\to\mathbb{R}_+$ עבור $C=(C_1,...,C_k)$ חלוקה של הדומיין (הסימון G נובע מ־$goal$). בעיות מינימיזציה אלה הן $NP-hard$ וחלקן אפילו קשות לקירוב ועל כן במקרים רבים נבקש גם את הפרמטר k עבור כמות ה־$clusters$ הרצויה (במקום שהאלגוריתם ימצא לבד).

כאשר דנים באלגוריתם $\underline{k-means}$, מחפשים חלוקה $C_1,...,C_k$ כאשר C_i מיוצג על ידי הצנטרואיד ($centroid$) μ_i. מניחים כי $\mathcal{X}$ מוכל במטריקה רחבה יותר $\mathcal{X}'$ אשר נקודות בצנטרואיד נמצאות ב־$\mathcal{X}'$. כלומר נקודות הצנטרואידים אינם בהכרח נמצאים ב־$\mathcal{X}$. על כן ב־$k-means$ מודדים את המרחק הריבועי של כל דגימה מהצנטרואיד של ה־$cluster$:

$$\mu_i(C_i)\overset{def.}{=}\underset{\mu\in\mathcal{X}'}{argmin}\sum_{x\in C_i}d(x,\mu)^2$$

ומכאן כי עבור דומיין $\mathcal{X}$ ופונקציית מרחק בין דגימות d:

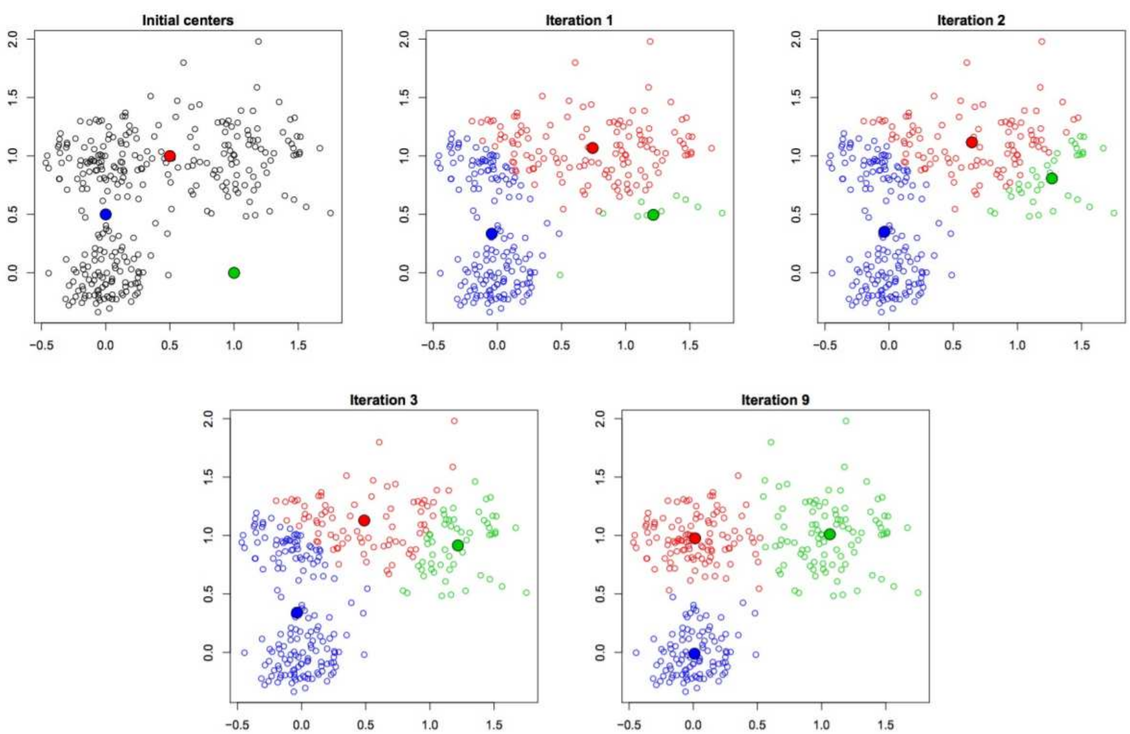
$$G_{k-means}\left(\left(\mathcal{X},d\right),\left(C_{1},...,C_{k}\right)\right)=\sum_{i=1}^{k}\sum_{x\in C_{i}}d\left(x,\mu_{i}\left(C_{i}\right)\right)^{2}$$

$$\Updownarrow$$

$$G_{k-means}\left(\left(\mathcal{X},d\right),\left(C_{1},...,C_{k}\right)\right)=\underset{\mu_{1},...,\mu_{k}\in\mathcal{X}'}{min}\sum_{i=1}^{k}\sum_{x\in C_{i}}d\left(x,\mu_{i}\right)^{2}$$

כפי שכבר תיארנו בעיות אלה הן $NP-hard$ ולכן אחת היוריסטיקות שבשימוש היא $Lloyd's\,algorithm$ אשר משתמשת ב־$alternating\,minimization$:

1. עבור סט דגימות $x_{1},...,x_{m}$ ומספר $clusters$ רצוי $k\in\mathbb{N}$.
2. נבחר באופן מקרי $\mu_{1},...,\mu_{k}$ צנטרואידים.
3. נפעל באופן איטרטיבי עד להתכנסות (עד אשר ההבדל בצנטרואידים באיטרציה t לאיטרציה $t+1$ קטנה מסף שנחליט)
   (א) נגדיר C_{i} קבוצת הנקודות שהכי קרובות לצנטרואיד μ_{i} מכל צנטרואיד אחר.
   (ב) נעדכן את הצנטרואיד μ_{i} להיות מרכז הכובד של הקבוצה: $\mu_{i}=\frac{1}{|C_{i}|}\sum\limits_{x\in C_{i}}x$



הערה 27.1 קיימים משפטים המוכיחים כי בכל איטרציה השונות בתוך $cluster$ יורדת וכי לבסוף יש התכנסות.

הערה 27.2 חשוב לשים לב שהערכים הראשונים שהצנטרואידים מקבלים משפיעים דרמטית על ה־$clusters$ שנקבל. כלומר ה–$variance$ של אלגוריתם זה גדול. נרצה למשל להריץ מספר פעמים ולראות מה האופציה שיצאה טובה מבחינתנו. בנוסף, נזכור גם שאנחנו קובעים את ה־k עבור מספר ה־$clusters$. לא מובטח שה־$data$ אכן ניתן להפרדה טובה (והגיונית על פי מה שה־$data$ מייצג) ל־k קבוצות.