אביב יעיש

1. למידת אונליין ו linear predictors

- בלמידת אונליין המטרה היא לעשות כמה שפחות טעויות.
- משימה בלתי אפשרית בלי ידע מוקדם. מניחים כי f האמיתית באה מ $\mathcal{H}\subset Y^X$.
- גם ידע מוקדם על מחלקת ההיפותזות לא תמיד עוזר (למשל במקרה של ספים על $\mathbb{R}$, תמיד אפשר להגדיל את הרזולוציה).
- שני אלג' ללמידת online של מחלקות סופיות: Halving ו Consistent learner
- Consistent learner - נאתחל את $V_1=\mathcal{H}$ ואז כל פעם נחזיר תשובה ע"פ איזשהי $h\in V_t$ ונוריד מ V_{t+1} את כל ההיפותזות שטועות. מבצע מקסימום $|\mathcal{H}|-1$ טעויות.
- Halving - עובד בצורה דומה, רק מחזיר את הצבעת הרוב על הדוגמה הנוכחית. מבצע מקסימום $\log_2(|\mathcal{H}|)$ טעויות. זמן הריצה גדל לינארית עם $|\mathcal{H}|$.
- ישנם מימושים יעילים יותר עבור מחלקות היפותזות עם מבנה מסוים (למשל ספים על תחום בדיד).
- חצאי מרחב (Halfspaces) - $\mathcal{H}=\{x\mapsto \text{sign}(\langle \mathbf{w},\mathbf{x}\rangle+b)\}$. הכללה של ספים לרב מימד. אפשר להוריד את b בשביל הנוחות (מוסיפים קואורדינטה ל x,w).
- כדור היחידה $B=\left\{\mathbf{w}\in\mathbb{R}^d: ||\mathbf{w}||^2\le 1\right\}$ (כאשר $||\mathbf{w}||^2=\langle\mathbf{w},\mathbf{w}\rangle=\mathbf{w}^T\mathbf{w}$).
- אלגוריתם האליפסואיד (מימוש יעיל שדומה ל Halving לספים בדידים רב מימדיים) שומר כדור יחידה המכיל את כל ה w האפשריים (כאשר התיוג הוא לפי $\text{sign}\left(\mathbf{w}_t^T\mathbf{x}_t\right)$) וכל פעם מצמצם אותו לאליפסואיד עם נפח מינימלי המכיל את כל מה שעדיין צודק על כל מה שראינו). מבצע לכל היותר $2d(2d+2)\log n$ טעויות.

2. PAC learning

- בלמידת Batch מקבלים קבוצת דוגמאות $S\in(X\times Y)^m$. הלומד מחזיר כלל תיוג h כרגיל. המטרה היא לצדוק על דוגמאות עתידיות, כלומר להחזיר h שדומה כמה שיותר ל f האמיתית.
- מגדירים טעות הכללה $L_{D,f}(h)=\mathbb{P}[h(x)\neq f(x)]=D(\{x\in X: h(x)\neq f(x)\})$. המטרה היא למזער את הפונקציה הזאת. הדוגמאות הן i.i.d וכן מניחים Realizability (קיימת פונקציית תיוג אמיתית).
- מודל PAC - מנסים למזער את $L_{D,f}$ עד כדי ϵ ובסיכוי $1-\delta$ ע"פ בחירת המדגם (פרמטרים של המשתמש), כאשר D,f לא ידועים ללומד, וכן הוא יכול לבקש לפחות $m(\epsilon,\delta)$ דוגמאות.
- משפט ה NFL אומר שאם $\mathcal{H}=Y^X$ הבעיה בלתי פתירה, כלומר לכל אלגוריתם לומד יש D,f שבסיכוי גדול מ δ הטעות עליהם תהיה יותר מ ϵ (וכמובן קטנה מחצי - ניחוש בין שתי אופציות).
- למידת מחלקות סופיות באמצעות כלל למידה Consistent, נקרא גם ERM - מזעור שגיאה אמפירית (שגיאת אימון ולא הכללה).
- עבור מחלקה סופית אם $m\ge\frac{\log(|\mathcal{H}|/\delta)}{\epsilon}$ אזי לכל D,f מתקיים בסיכוי של לפחות $1-\delta$ על בחירת S מגודל m: $L_{D,f}(\text{ERM}_{\mathcal{H}}(S))\le\epsilon$ (הוכחה עם חסם האיחוד).
- מחלקה $\mathcal{H}$ היא למידת PAC אם קיימת פונקציה $m_{\mathcal{H}}:(0,1)^2\to\mathbb{N}$ ואלג' כך שלכל ϵ,δ והסתברות D ופונקציית תיוג $f:X\to\{0,1\}$, כאשר מריצים את האלג' על $m\ge m_{\mathcal{H}}(\epsilon,\delta)$ דוגמאות i.i.d שנוצרו ע"י D ותוגו ע"י f, אזי האלג' יחזיר היפותזה h כך שבסיכוי של לפחות $1-\delta$ מתקיים $L_{D,f}(h)\le\epsilon$. $m_{\mathcal{H}}$ (המינימלית) היא סיבוכיות המדגם של $\mathcal{H}$.
- עבור קבוצת דוגמאות $C\subset X$, $\mathcal{H}$ מנתצת את C אם היא מכילה היפותזה לכל תיוג אפשרי של איברי C. מימד ה VC של מחלקה $\mathcal{H}$ הוא גודל הקבוצה הכי גדולה שמנותצת ע"י $\mathcal{H}$. (אם $\mathcal{H}$ מנתצת קבוצה C כלשהי, היא לא מספקת עליה שום ידע מוקדם). למחלקות סופיות מתקיים $\text{VCdim}(\mathcal{H})\le\log_2(|\mathcal{H}|)$. המימד הוא בדר"כ מספר הפרמטרים שמאפיינים את המחלקה.
- אם $\mathcal{H}$ היא מחלקת היפותזות של תיוג בינארי אזי קיימים C_1,C_2 (מוחלטים) כך שסיבוכיות המדגם של למידת PAC של $\mathcal{H}$ היא $C_1\frac{d+\log(1/\delta)}{\epsilon}\le m_{\mathcal{H}}(\epsilon,\delta)\le C_2\frac{d\log(1/\epsilon)+\log(1/\delta)}{\epsilon}$, ו ERM משיג סיבוכיות זאת. לכן, אם $\mathcal{H}$ היא למידת PAC אם ורק אם $\text{VCdim}(\mathcal{H})<\infty$.
- ERM עבור חצאי מרחב - אפשר עם אליפסואיד, או עם אלג' פרספטרון (מתחיל ממשקולות 0 ומעדכן את המשקולות עם y_ix_i כל עוד יש דוגמה שטועים עליה) אשר עושה לכל היותר $||w^*||^2\max_i||x_i||^2$ עדכונים.
- א"ש מרקוב: $\mathbb{P}[X\ge t]\le\frac{\mathbb{E}[X]}{t}$ עבור $X\ge 0, t>0$, ועבור $X_1,\ldots,X_m$ שמתפלגים i.i.d הא"ש מתקיים עבור $\overline{X}$.
- א"ש צ'ביצ'ב: עבור $t>0$ (X לא בהכרח אי שלילי) נקבל $\mathbb{P}[X-\mathbb{E}[X]\ge t]\le\frac{\text{Var}[X]}{t^2}$. עבור $\overline{X}$ מקבלים א"ש דומה עם m במכנה.
- א"ש הופדינג: עבור $a_1\le X_i\le b_i$ בלתי תלוים וחסומים נקבל $\mathbb{P}\left[\overline{X}-\mathbb{E}\left[\overline{X}\right]\ge\epsilon\right]\le 2\exp\left(-2m^2\epsilon^2/\sum(b_i-a_i)^2\right)$. עבור משתנים i.i.d חסומים בין 0 ל1 נקבל הסתברות של $2\exp\left(-2m\epsilon^2\right)$. עבור הטלות מטבע עם הטיה $\hat{p}$ נקבל ש $m(\epsilon,\delta)\le\left\lceil\frac{1}{2\epsilon^2}\cdot\log\left(\frac{2}{\delta}\right)\right\rceil$.

3. General learning model and Bias-Complexity tradeoff

- השוני במודל האגנוסטי של PAC הוא שההתפלגות D היא על $Z=X\times Y$, לא חייבת להיות פונקציה $f\in\mathcal{H}$, והמטרה היא למזער עד כדי השגיאה המינימלית: $L_D(A(S))\le\min_{h\in\mathcal{H}}L_D(h)+\epsilon$. נשים לב כי מתקיים $D_x(y)=D(y|x)=D((x,y)|x)$.
- בנוסף, במודל זה Y יכולה להיות קבוצה עם יותר איברים (קבוצה סופית בבעיית Multiclass או אינסופית בבעיית Regression). כתוצאה מכך, גם $\ell(h,(x,y))$ יכולה להיות אחרת (ריבועי, ערך מוחלט, טבלת מחירים על $Y\times Y$ וכו'). כך $L_D(h)=\mathbb{E}[\ell(h,z)]$. הלומד יודע את $\mathcal{H},Z,\ell$.
- מדגם נקרא ϵ-מייצג אם לכל $h\in\mathcal{H}$ מתקיים $|L_S(h)-L_D(h)|\le\epsilon$. אם מדגם הוא $\frac{\epsilon}{2}$ מייצג, אזי כל פלט של $\text{ERM}_{\mathcal{H}}(S)=h_S\in\arg\min_{h\in\mathcal{H}}L_S(h)$ מקיים $L_D(h_S)\le\min_{h\in\mathcal{H}}L_D(h)+\epsilon$.
- למחלקה $\mathcal{H}$ יש תכונת UC (Uniform Convergence) אם קיימת $m_{\mathcal{H}}^{UC}$ כך שלכל ϵ,δ והסתברות D מתקיים בהסתברות $1-\delta$ שמדגם בגודל m הוא ϵ מייצג.
- במידה ולמחלקה $\mathcal{H}$ יש את התכונה UC אזי היא למידת PAC אגנוסטי עם $m_{\mathcal{H}}(\epsilon,\delta)\le m_{\mathcal{H}}^{UC}(\epsilon/2,\delta)$ ואלג' ERM הוא אלג' למידה ל $\mathcal{H}$ (כלומר התכונה UC מספיקה ללמידות).
- משפט: אם $\mathcal{H}$ היא מחלקה סופית עם Loss חסום בקטע $[0,1]$ אזי $\mathcal{H}$ היא למידת PAC אגנוסטי עם $m_{\mathcal{H}}(\epsilon,\delta)\le\left\lceil\frac{2\log(2|\mathcal{H}|/\delta)}{\epsilon^2}\right\rceil$ (כלומר יש לה את תכונת UC).
- שימוש במשפט הנ"ל וטריק הדיסקרטיזציה מאפשר ללמוד כל מחלקה שמאופיינת עם d מספרים שכל אחד מהם בייצוג של b ביטים בסיבוכיות מדגם $m_{\mathcal{H}}(\epsilon,\delta)\le\left\lceil\frac{2db+2\log(2/\delta)}{\epsilon^2}\right\rceil$.
- בבעיית הרגרסיה הלינארית $X\subset\mathbb{R}^d, Y\subset\mathbb{R}, \mathcal{H}=\left\{\mathbf{x}\mapsto\langle\mathbf{w},\mathbf{x}\rangle:\mathbf{w}\in\mathbb{R}^d\right\}$ משתמשים ב Squared Loss $\ell(h,(x,y))=(h(x)-y)^2$ ומנסים למזער (את הממוצע שלו) על פני המדגם.
- גרדיאנט $\nabla f(x)$ של פונקציה $f:\mathbb{R}^n\to\mathbb{R}$ הוא וקטור נגזרות חלקיות של f. עבור f שמורכבת מ m פונקציות שונות כאלה נקבל את היעקוביאן: מטריצת $m\times n$ נגזרות חלקיות, כאשר השורה ה i היא הגרדיאנט של f_i. למשל עבור $f(w)=Ax$ מתקיים $J_w(f)=A$. כלל השרשרת קובע כי עבור f,g מתקיים $J_x(f\circ g)=J_{g(x)}(f)J_x(g)$.
- Least squares - רגרסיה לינארית עם Square loss. הפתרון הוא $w=\left(XX^T\right)^{-1}Xy$ (כאשר X היא מטריצת עמודות של הדוגמאות). במקרה ש XX^T לא הפיכה הפתרון הוא $w=X^{\dagger T}y$ כאשר $X^{\dagger T}=V\Sigma^{\dagger}U^T$ עבור $X=U\Sigma V^T$. אם נרצה לעשות Polynomial Fitting מדרגה n פשוט נפתור Least squares על $(\psi(x_i),y_i)$ כאשר $\psi(x)=\left(x^i\right)_{i=0}^n$.
- את שגיאת ההכללה של היפותזה $h_S=\text{ERM}_{\mathcal{H}}(S)$ ניתן לפרק לשגיאת אפרוקסימציה: $\epsilon_{app}=\min_{h\in\mathcal{H}}L_D(h)$ שתלויה בהגבלות שהצבנו על $\mathcal{H}$ שלנו, לא תלויה במדגם ויורדת עם הסיבוכיות של $\mathcal{H}$ (גודל או ממד VC) הגורם השני הוא שגיאת האסטימציה: $\epsilon_{est}=L_D(h_S)-\epsilon_{app}$ שתלויה בכמה המדגם מייצג טוב את העולם. עולה עם הסיבוכיות של $\mathcal{H}$.
- Validation - אחרי שלמדנו h נרצה להעריך את שגיאת ההכללה על מדגם חדש i.i.d (שנסמנו V בגודל m_v) ולפי חסם הופדינג (רק אם ℓ חסום בין $[0,1]$) נקבל: $|L_V(h)-L_D(h)|\le\sqrt{\frac{\log(2|\mathcal{H}|/\delta)}{2m_v}}$ בהסתברות לפחות $1-\delta$.
- ניתן להשתמש בוולידציה בשביל בחירת מודל מתאים - ניצור r היפותזות ממחלקות שונות ואז נבצע $h^*\in\text{ERM}_{\mathcal{H}}(V)$ כאשר $\mathcal{H}=\{h_i\}_{i=1}^r$. בגלל ש $\mathcal{H}$ סופית נקבל $L_D(h^*)\le\min_{h\in\mathcal{H}}L_D(h)+\sqrt{\frac{2\log(2|\mathcal{H}|/\delta)}{|V|}}$ בהסתברות לפחות $1-\delta$.
- בשיטת $k-$fold cross validation מחלקים את המדגם ל k קבוצות, ואז לכל מודל מגדירים את השגיאה להיות ממוצע השגיאות של ההיפותזות שנלמדו על כל הקבוצות פרט לקבוצה i (כאשר השגיאה היא על הקבוצה ה i, כלומר זאת שהאלג' לא למד עליה) ואז מלמדים את האלג' על כל המדגם ועם המודל שמזער את השגיאה שהגדרנו לעיל.

4. Boosting

- מחלקה $\mathcal{H}$ היא (ϵ,δ) למידה חלש אם קיים אלג' וגודל מדגם m כך שלכל התפלגות ולכל $f\in\mathcal{H}$ ההסתברות ששגיאת ההכללה תהיה קטנה מ ϵ גדולה מ $1-\delta$. כל מחלקה היא למידה $(1/2,0)$.
- בוסטינג ל Confidence - נניח כי A הוא (ϵ_0,δ_0) לומד חלש הדורש m_0 דוג' נרצה לכל ϵ,δ ללמוד את $\mathcal{H}$ עם דיוק $\epsilon+\epsilon_0$ וביטחון δ. שלב 1 - נפעיל את A על $k=\left\lceil\frac{\log(2/\delta)}{\log(1/\delta_0)}\right\rceil$ מדגמים i.i.d בגודל m_0 ואז ניקח את $\hat{h}\in\arg\min_{h_i}L_V(h_i)$ (כאשר V מדגם חדש בגודל $|V|\ge\frac{2\log(4k/\delta)}{\epsilon^2}$ ו h_i הן ההיפותזות משלב 1) ונקבל כי $L_D\left(\hat{h}\right)\le\epsilon_0+\epsilon$.
- אם יש לנו לומד A שמבטיח $\mathbb{E}[L_D(A(S))]\le\min_{h\in\mathcal{H}}L_D(h)+\epsilon$ אז A הוא $(2\epsilon,1/2)$ לומד חלש (משתמשים בא"ש מרקוב על השגיאה של האלג' פחות השגיאה המינימלית).

- אלג' AdaBoost (עושה בוסטינג לדיוק) מקבל מדגם S, לומד חלש WL ומספר סיבובים T: מאתחל התפלגות אחידה $D^{(1)}$ ובכל סיבוב t מחשב $h_t = \mathrm{WL}\left(D^{(t)}, S\right)$ ואז $\epsilon_t = L_{D^{(t)}}\left(h_t\right) = \sum_{i=1}^{m} D_i^t \mathbf{1}_{[y_i \neq h_t(x_i)]}$ ואז מעדכן לכל $i = [m]$ את ההתפלגות להיות $D_i^{(t+1)} = D_i^{(t)} \exp\left(-w_t y_i h_t\left(x_i\right)\right)$ (עם גורם נרמול כדי שתהיה התפלגות חוקית) כאשר $w_t = \frac{1}{2}\log\left(\frac{1}{\epsilon_t} - 1\right)$, ואז מחזיר את הסימן של הממוצע המשוקלל של ההיפותזות בכל שלב. אינטואיטיבית האלג' מכריח את ההיפותזה הבאה להתמקד בדוגמאות שהוא טעה עליהן פעם קודמת. השגיאה של h_t על ההתפלגות $D^{(t+1)}$ היא בדיוק חצי (כך נבחרו המשוואות).
- האלג' מבטיח שאם WL הוא $\left(1/2 - \gamma, \delta\right)$ לומד חלש, אז בסיכוי $1 - \delta T$ מתקיים שעבור h_s שהאלג' מחזיר $L_S\left(h_s\right) \leq \exp\left(-2\gamma^2 T\right)$. אם ניקח $T \geq \frac{\log(1/\epsilon)}{2\gamma^2}$ נקבל שגיאה קטנה מ ϵ, ואם נבחר $\epsilon = 1/2m$ נקבל שגיאת אימון 0 (כי היא בדידה). לרוב δ לא מעניין כי ניתן לעשות לו בוסטינג במחיר לא גדול (בערך פי 4 יותר סיבובים של האלג').
- שגיאת האפרוקסימציה קטנה עם T (≈סיבוכיות המחלקה) ושגיאת האסטימציה גדלה עם T. ממד ה VC של מחלקת כל ההיפותזות שהאלג' מחזיר $L\left(B, T\right)$ (כל ההרכבות של T היפותזות מ B שה WL מחזיר) הוא קטן מ $O\left(T \cdot \mathrm{VCdim}\left(B\right)\right)$. בנוסף, אם $m \geq \Omega\left(\frac{\log(1/\delta)}{\gamma^2 \epsilon}\right), T = \log\left(m\right)/\left(2\gamma^2\right)$ אז בהסתברות של לפחות $1 - \delta$ נקבל כי $L_{D,f}\left(h_s\right) \leq \epsilon$.

Nonuniform learning, MDL, SRM, Decision Trees, Nearset neighbors

- סוגים שונים של ידע מוקדם יכולים להיות שהיפותזות "קצרות" הן יותר טובות, או שדברים שנראים דומים הם באמת דומים.
- בהינתן מחלקת היפותזות בת מניה $\mathcal{H}$ נגדיר $w : \mathcal{H} \to \mathbb{R}$ כך ש $\sum_{h \in \mathcal{H}} w\left(h\right) \leq 1$. למשל עבור $w\left(h\right) = 2^{-|h|}$ כאשר $|h|$ הוא האורך של המילה שמתאימה להיפותזה (בשפה שהיא חסרת רישות) מתקיים $\sum_h w\left(h\right) \leq 1$.
- אם יש לנו מחלקת היפותזות ופונקציית משקל כנ"ל w, אזי בהסתברות לפחות $1 - \delta$ לכל היפותזה מתקיים $L_D\left(h\right) \leq L_S\left(h\right) + \sqrt{\frac{-\log(w(h)) + \log(2/\delta)}{2m}}$ (בניגוד לחסם ה VC שם מחליפים את הlog ב $\mathrm{VCdim}\left(\mathcal{H}\right)$).
- מזעור חסם ה VC מוביל לאלג' ERM ומזעור חסם ה MDL מוביל לכלל הלמידה $\mathrm{MDL}\left(S\right) \in \arg\min_{h \in \mathcal{H}} \left[L_S\left(h\right) + \sqrt{\frac{-\log(w(h)) + \log(2/\delta)}{2m}}\right]$. כלל הלמידה הזה נותן לנו Bias-complexity tradeoff.
- MDL הוא לומד אוניברסלי, אך לומד Nonuniform. בלמידה לא יוניפורמית, מספר הדוגמאות תלוי גם ב h ולא רק ב ϵ, δ. למשל מחלקת כל הפונקציות החשיבות היא למידה לא יוניפורמית אך לא למידת PAC. מחלקה היא למידה לא יוניפורמית אם ורק אם היא איחוד בן מנייה של מחלקות שהן למידות PAC.
- ניתן ללמוד באופן NU את מחלקת כל הפונקציות החשיבות, אבל בPAC לא ניתן.
- $\mathrm{SRM}\left(S\right) \in \arg\min_{h \in \mathcal{H}} \left[L_S\left(h\right) + \min_{n: h \in \mathcal{H}_n} \sqrt{C \frac{d_n - \log(w(n)) + \log(1/\delta)}{m}}\right]$ וכן לכל $h \in \mathcal{H}$ מתקיים $L_D\left(\mathrm{SRM}\left(S\right)\right) \leq L_S\left(h\right) + \min_{n: h \in \mathcal{H}_n} \sqrt{C \frac{d_n - \log(w(n)) + \log(1/\delta)}{m}}$. כלל ה SRM הוא לומד לא יוניפורמי גנרי. מחלקת כל הפונקציות מעל דומיין אינסופי היא לא למידה לא־יוניפורמית. סיבוכיות מדגם של SRM: $m_H \leq \min_{n: h \in \mathcal{H}_n} C \frac{d_n - \log(w(n)) + \log(1/\delta)}{\epsilon^2}$
- מחלקת עצי ההחלטה עם k עלים היא בעלת $\mathrm{VCdim} = k$. מחלקת כל עצי ההחלטה מעל $\{0,1\}^d$ היא מממד $\mathrm{VCdim} = 2^d$. נרצה להעדיף עצים קצרים יותר ע"י שפת תיאור לעצים ואז שימוש בכלל הלמידה של MDL על מנת לחפש עץ עם n צמתים שממזער , אך בעיה זו היא NP קשה. נשתמש באלג' המקרב שעובד בצורה חמדנית שנקרא ID3 - בכל שלב הוא מפצל לפי המאפיין שממקסם את ה $\mathrm{Gain}\left(S, i\right) = C\left(\mathbb{P}\left[y\right]\right) - \left(\mathbb{P}\left[x_i\right] C\left(\mathbb{P}\left[y|x_i\right]\right) + \mathbb{P}\left[\neg x_i\right] C\left(\mathbb{P}\left[y|\neg x_i\right]\right)\right)$ וממשיך רקורסיבית עד שנגמרים המאפיינים. ניתן גם לגזום, או לבנות יער של הרבה עצים (כמעט תמיד עובד), וגם להתמודד עם Real valued features (ספים).
- בשיטת ה Nearest neighbors אין מחלקת היפותזות. תהליך הלמידה הוא פשוט שמירת דוגמאות ובזמן הפרידקציה מחזירים את הצבעת הרוב של k השכנים הכי קרובים לדוגמא. אם מניחים c־ליפשיציות של $\eta\left(x\right) = \mathbb{P}\left[y = 1|x\right]$, אזי $\mathbb{E}\left[L_D\left(h_S\right)\right] \leq \left(1 + \sqrt{\frac{8}{k}}\right) L_D\left(h^*\right) + \left(6c\sqrt{d} + k\right) m^{-1/(d+1)}$. הפרמטר k קשור ל Bias-complexity tradeoff. בנוסף, מספר הדוגמאות גדל אקספ' עם ממד הדוגמאות. בנוסף, מספר הדוגמאות תלוי בהתפלגות D. אלגוריתם Nearest neighbors הוא Universal consistent.
- PAC חזק יותר מ Non uniform שחזק יותר מ Universal consistent.

6. קמירות ו SGD

- קבוצה $C \subset V$ קמורה אם ורק אם לכל שני וקטורים $u, v \in C$ וקבוע $\lambda \in [0,1]$ מתקיים $\lambda u + \left(1 - \lambda\right) v \in C$ כלומר הקטע המחבר בין הנקודות נמצא כולו בתוך הקבוצה. כל תת מרחב וקטורי $U \subseteq V$ הוא קבוצה קמורה. חיתוך של מס' סופי של קבוצות קמורות היא קבוצה קמורה. חיבור של שתי קבוצות קמורות הוא קבוצה קמורה. הכפלה של קבוצה קמורה בסקלר כלשהו היא קבוצה קמורה. כל על מישור $W = \{v \in V : \langle w, v\rangle = b\}$ הוא קבוצה קמורה.
- משפטים לא קשורים: תהי C קבוצה קמורה וסגורה ב V, ו $u \in V$, אזי קיימת נקודה $w \in C$ יחידה (שנקראת ההטלה של u ל C) כך ש $||u - w|| \leq ||u - v||$. כמו כן, $\langle u - w, v - w\rangle \leq 0$. לכל קבוצה קמורה ונקודה שלא בקבוצה קיים על מישור מפריד ביניהם
- פונקציה $f : C \to \mathbb{R}$ היא קמורה אם ורק אם לכל שני וקטורים $u, v \in C$ מתקיים $f\left(\lambda u + \left(1 - \lambda\right) v\right) \leq \lambda f\left(u\right) + \left(1 - \lambda\right) f\left(v\right)$. תנאי שקול לכך הוא שה $\mathrm{epigraph}\left(f\right) = \{\left(x, \beta\right) : f\left(x\right) \leq \beta\}$ היא קבוצה קמורה. פונקציית נורמה היא קמורה. כל פונקציה אפינית היא קמורה. כל צירוף לינארי של פונקציות קמורות עם סקלרים חיוביים הוא פונקציה קמורה. הרכבה של פונקציה קמורה על פונקציה אפינית כלשהי היא פונקציה קמורה גם כן. סופרימום של מס' סופי של פונקציות קמורות היא פונקציה קמורה. פונקציה חד ממדית ודיפרנציאבילית f היא קמורה אם ורק אם הנגזרת שלה מונוטונית עולה, ואם היא דיפרנציאבילית פעמיים אז היא קמורה אם ורק אם הנגזרת השנייה אי שלילית. הפונקציה הלוגיסטית $f\left(w\right) = \log\left(1 + \exp\left(-y \langle w, x\rangle\right)\right)$ היא קמורה.
- אם f היא קמורה, אזי כל מינימום לוקאלי הוא מינימום גלובאלי. בנוסף, המשיקים לפונקציה עוברים מתחת לפונקציה כלומר לכל u מתקיים $f\left(u\right) \geq f\left(w\right) + \langle \nabla f\left(w\right), u - w\rangle$.
- v נקרא סאב־גרדיאנט של f ב w אם לכל u מתקיים $f\left(u\right) \geq f\left(w\right) + \langle v, u - w\rangle$. קבוצת כל ה SG של f בנק' w נקראת דיפרנציאל ומסומנת $\partial f\left(w\right)$. פונקציה f קמורה אם ורק אם לכל w יש לפחות SG אחד (כלומר $\partial f\left(w\right) \neq \emptyset$. אם פונקציה f קמורה ודיפרנציאבילית ב x אזי $\partial f\left(x\right) = \{\nabla f\left(x\right)\}$אם וקטור ה 0 הוא SG של f בנקודה w אם ורק אם w היא מינימום גלובאלי.
- פונקציה $f : C \to \mathbb{R}$ נקראת ρ־ליפשיצית אם לכל w_1, w_2 מתקיים $|f\left(w_1\right) - f\left(w_2\right)| \leq \rho ||w_1 - w_2||$. אם פונקציה f קמורה, אז היא ρ־ליפשיצית אם ורק אם הנורמה של כל ה SG היא לכל היותר ρ.
- באופטימיזציה קמורה רוצים למזער פונקציה קמורה $f\left(w\right)$ עבור $w \in C$. מקרים מיוחדים: פיזיביליות - f היא פונקציה קבועה, כלומר צריך למצוא $w \in C$. Unconstrained minimization - $C = \mathbb{R}^d$.
- אלגו' האליפסואיד לבעיית הפיזיביליות מניח ש C חסומה מלמעלה ומלמטה וכן שניתן להפריד בין w, C (אם $w \notin C$ קיים על מישור מפריד), ובכל שלב מעדכן את האליפסואיד לפי אוב ההפרדה. מתכנס אחרי $2d\left(2d + 2\right)\log\left(R/r\right)$ צעדים לכל היותר.
- מימוש של האוב הפרדה: בהינתן ש C נתונה בתור חיתוך של $level - set$ של פונקציות קמורות נבדוק אם $f_i\left(w\right) \leq 0$ לכל i. אם כן, $w \in C$, אחרת נחזור מינוס של SG של $f_i\left(w\right)$.
- אם רוצים למזער פונקציה קמורה וליפשיצית אפשר להשתמש באליפסואיד עם $2d\left(2d + 2\right)\log\left(\frac{\rho ||w^*||}{\epsilon} + 1\right)$ צעדים.
- אלג' gradient descent מתחיל מווקטור משקולות התחלתי $w^{(1)}$ ומעדכן בכל שלב $w^{(t+1)} = w^{(t)} - \eta \nabla f\left(w^{(t)}\right)$ ($\eta > 0$ הוא Bias-complexity tradeoff). לבסוף מחזיר w ממוצע. ב SubGD מחליפים את הגרדיאנט ב SG. SubGD מתכנס תוך $\frac{||w^*||^2 \rho^2}{\epsilon^2}$ צעדים עבור פונקציה f שהיא ρ־ליפשיצית וקמורה.
- על מנת להביע את הבעיה של מציאת מישור מפריד כבעיית אופטימיזציה קמורה נכתוב אותה כ $f\left(w\right) = \max_i -y_i \langle w, x_i\rangle$. f קמורה ו 1־ ליפשיצית. $w^{(t)}$ ש SubGD מחזיר על מישור מפריד. כמעט זהה ל Batch perceptron. הגורם η שחסר בפרספטרון לא משנה את הסימן לכן לא חשוב.
- SubGD לא תלוי ב d (רק שלב העדכון עולה לינארית עם d) לעומת האליפסואיד שנותן דיוק הרבה יותר טוב (תלוי לוגריתמית בדיוק) אך תלוי ריבועית ב d^2. אם מדברים על מציאת על מישור מפריד אז ככל שהמדגם שלנו מופרד יותר טוב ככה נעדיף שיטות של SubGD ואם הוא מופרד גרוע נרצה לעבוד עם האליפסואיד (כי הוא תלוי רק לוגריתמית בשוליים) אך התלות היא בממד.
- בעיית למידה היא קמורה אם $\mathcal{H}$ היא קבוצה קמורה ואם לכל z מתקיים ש $\ell\left(\cdot, z\right)$ היא קמורה. $\mathcal{H}$ תהיה פרמטריזציה של הפונקציות. פתרון ה ERM של בעיות למידה קמורות הוא בעיית אופטימיזציה קמורה. לא כל בעיות הלמידה הקמורות הן למידות. נצטרך להוסיף אילוץ שאומר ש $\mathcal{H}$ חסומה (לכל $w \in \mathcal{H}$ מתקיים $||w|| \leq B$) וש ℓ ליפשיצית. ניתן להחליף את הדרישה לליפשיציות בדרישה ל"חלקות" (הנגזרת של הפונקציה היא ליפשיצית) ואי שליליות של ℓ.
- אם פונקציית ה loss היא לא קמורה נשתמש בפונקציית Surrogate loss שהיא כן קמורה והיא חוסמת מלמעלה את הפונקציה המקורית. למשל עבור $0 - 1\mathrm{loss}$ נגדיר $\ell^{hinge}\left(w, \left(x, y\right)\right) = \max\{0, 1 - y \langle w, x\rangle\}$.
- באלג' SGD מעדכנים כל פעם ע"פ גרדיאנט בנקודה אחת שנבחרת אקראית מתוך D (במטרה למזער את L_D, אם רוצים למזער את L_S בוחרים נקודה מתוך מדגם נתון) וממזערים את $L_D\left(h\right)$ במקום $L_S\left(h\right)$. אין פה סתירה למשפט היסודי של הלמידה כי לא מדובר כאן על בעיית קלסיפיקציה בינארית (עם $0 - 1\mathrm{loss}$ שהוא לא קמור). בסוג בעיות כאלה ERM הוא לא בהכרח הפרדיגמה הכי טובה. אם יש לנו בעיית למידה שהיא קמורה חסומה וליפשיצית, אם נבחר $T \geq \frac{B^2 \rho^2}{\epsilon^2}$ צעדים של SGD עם $\eta = \sqrt{\frac{B^2}{\rho^2 T}}$ נקבל $\mathbb{E}\left[L_D\left(\overline{w}\right)\right] \leq \min_{w \in \mathcal{H}} L_D\left(w\right) + \epsilon$.
- קמירות לבדה לא מבטיח למידות. בעית רגרסיה לינארית עם loss ריבועי היא לא למידה, אלא אם מניחים חסימות של $X, \mathcal{H}$.
- כשמסתכלים על סיווג מסמכי טקסט לנושא למשל, ומשתמשים בשיטת Bag of words, אם ננתח לפי חסמי VCdim נקבל Sample complexity מאוד גבוה, לעומת זאת אם ננתח לפי חסמי נורמה נקבל חסם יחסית נמוך יותר ($R^2 ||w||^2$) שלא תלוי בממד של המילון. ישנן בעיות הפוכות שבהן הממד הרבה יותר קטן מהנורמה ואז הוא עדיף.

7. SVM, RLM, Kernels

- עבור על מישור $L=\{v:\langle w,v\rangle=0\}$ המרחק של נקודה מהמישור אם $||w||=1$ הוא $d(x,L)=|\langle w,x\rangle|$. השוליים של על מישור הוא המרחק לנקודה הכי קרובה אליו $\min_i |\langle w,x_i\rangle+b|$ (וקטורים תומכים).

- אלגוריתם Hard-SVM מחפש על מישור עם שוליים הכי גדולים. הבעיה הזאת היא שקולה לבעיה של למזער את $||w||^2$ כך שלכל i מתקיים $y_i(\langle w,x_i\rangle+b)\geq 1$.

- בבעיית Soft-SVM היא $\min_w \frac{\lambda}{2}||w||^2+\frac{1}{m}\sum \ell^{hinge}(y_i\langle w,x_i\rangle)$. בבעיית Soft-SVM ניתן להחליף את גורם ה loss במזעור של ממוצע ה ξ_i כך שלכל i מתקיים $y_i\langle w,x_i\rangle \geq 1-\xi_i$ וכן $\xi_i\geq 0$, כלומר $\min_w \frac{\lambda}{2}||w||^2+\frac{1}{m}\sum\xi_i$.

- באופן כללי, רגולריזציה (RLM) היא הוספת גורם שאינו תלוי במדגם למזעור $\min_w R(w)+L_S(w)$. Soft-SVM היא בעיית רגולריזציה. בעיה נוספת היא Ridge regression שהיא RLM עם $\frac{\lambda}{2}||w||^2$.

- למה לעשות רגולריזציה? יכול להוריד את ה sample complexity - כמו ב MDL יש לנו ידע מוקדם שהיפותזות קצרות הן יותר טובות, וכן אלגוריתמים כאלה הם "יציבים". בנוסף, זה עוזר להוריד את סיבוכיות החישוב (גם לבעיות לא קמורות).

- עבור פונקציה $\epsilon:\mathbb{N}\to\mathbb{R}$ מונוטונית יורדת, אלגוריתם נקרא "בממוצע יציב להחלפה אחת" בקצב $\epsilon(m)$ אם לכל הסתברות מתקיים $\mathbb{E}\left[\ell\left(A\left(S^{(i)}\right),z_i\right)-\ell\left(A\left(S\right),z_i\right)\right]\leq\epsilon(m)$. אם אלג' הוא "בממוצע יציב להחלפה אחת" עם קצב $\epsilon(m)$ אז $\mathbb{E}\left[L_D\left(A\left(S\right)\right)-L_S\left(A\left(S\right)\right)\right]\leq\epsilon(m)$.

- משפט: רגולריזציית תיכונוב ($\lambda||w||^2$) היא מייצבת - בהנחה שפונקציית ה loss קמורה וליפשיצית, אז כלל ה RLM עם רגולריזציית תיכונוב היא "בממוצע יציב להחלפה אחת" עם $\epsilon(m)=\frac{2\rho^2}{\lambda m}$. את λ אפשר לבחור עם ולידציה.

- שיטת הקרנלים: מגדירים פונקציה $\psi:X\to F$ כאשר F הוא מרחב מאפיינים כלשהו, ואז נלמד חצאי מרחב על $(\psi(x_i),y_i)$. איך נבחר את המיפוי? דורש ידע על הבעיה.

- הקרנל של מיפוי ψ היא פונקציה שמממשת מכפלה פנימית במרחב המאפיינים: $K(x,x')=\langle\psi(x),\psi(x')\rangle$. לפעמים קל לחשב את K בלי לחשב את ψ באופן מפורש.

- עבור כל כלל למידה מהצורה של $w^*=\arg\min_w\left(f\left(\langle w,\psi(x_1)\rangle,\ldots,\langle w,\psi(x_m)\rangle\right)+\lambda||w||^2\right)$ קיימים α_i כך ש $w^*=\sum\alpha_i\psi(x_i)$. לכן, אם אנחנו יודעים את $G_{i,j}=\langle\psi(x_i),\psi(x_j)\rangle$ נוכל לחשב את $\langle w,\psi(x_i)\rangle=(G\alpha)_i$ וכן את $||w||^2=\alpha^TG\alpha$ וכך לקבל את בעיית האופטימיזציה: $\arg\min_\alpha\left(f(G\alpha)+\lambda\alpha^TG\alpha\right)$ (שהיא בממד הרבה יותר קטן לרוב). אם אכן משתמשים בטריק הזה, כאשר רוצים לבצע פרדיקציה על x מחשבים את $\langle w,\psi(x)\rangle=\sum_j\alpha_jK(x_j,x)$.

- קרנל פופולרי הוא קרנל גאוסי (RBF) - לומדים פולינומים ממעלה אינסופית ע"י קרנל יעיל.

- ע"פ תנאי מרסר - אם לכל בחירה של m דוגמאות המטריצה G היא מוגדרת אי שלילית אזי היא פונקציית קרנל חוקית. בנוסף, מכיוון שכל פונקציית קרנל היא מכ"פ, אזי היא חייבת לקיים את התנאים של מכ"פ (חיוביות בהחלט - לכל x מתקיים $\langle x,x\rangle\geq 0$ ושוויון רק ב 0, לינאריות וסימטריות).

8. צברור ו Features

- במודל צברור יש קבוצת אובייקטים ופונקציית מרחק d, בעוד הפלט הרצוי היא חלוקה של קבוצת האובייקטים ל k קבוצות זרות. הפלט יכול להיות גם עץ של קבוצות כך שבכל רמה יש חלוקה גסה יותר (ברמה הכי נמוכה יש את הדוגמאות וברמה הכי גבוהה יש קבוצה אחת שמכילה את כולם).

- בשיטות מבוססות קישורים נבנה בעצם דנדוגרם (עץ כמתואר לעיל) כאשר נתחיל מסט של $|X|$ קבוצות וכל פעם נחבר את שתי הקבוצות שהכי דומות לפי פונקציית המרחק. יכולות להיות כמה הרחבות של d לפונ' D שמאחדת בין קבוצות ובין נקודה לקבוצה (המרחק המינימלי בין הקבוצות, המרחק הממוצע, המרחק המקסימלי וכו').

- בשיטות מבוססות מזעור מחיר מגדירים פונקציית מחיר G אשר מגדירה עבור קלט של דוגמאות ופונקציית מרחק ביחד עם צברור מוצע את המחיר של הצברור, ואז מנסים למזער את הפונקציה G.

- במשפחת $k-means$ מגדירים את $G=\min_{\mu_i\in X'}\sum_{i=1}^k\sum_{x\in C_i}d(x,\mu_i)^2=\sum_{i=1}^k\sum_{x\in C_i}d(x,\mu_i(C_i))^2$ כאשר $\mu_i(C_i)=\arg\min_{\mu\in X'}\sum_{x\in C_i}d(x,\mu)^2$ הוא המרכז של הצבר ה i. ניתן גם להסתכל על מרכזים שהם רק בתוך הקבוצה המקורית ($k-medoids$) או להסתכל על מרחק בלי ריבוע ($k-median$).

- בעיית $k-means$ היא NP קשה, לכן משתמשים באלג' קירוב שנקרא $k-means$ שמתחיל ממרכזים אקראיים, ובכל שלב לוקח לצבר ה i את כל הנקודות שהכי קרובות למרכז הזה ואז מעדכן כל מרכז להיות המרכז של כל הנקודות שמשויכות אליו.

- איך מייצגים אובייקטים מהעולם האמיתי בתור וקטור של Features? בהנתן שיש לנו ייצוג של Features, האם זה הייצוג הכי טוב שיש?

- בהנתן מרחב דוגמאות ממימד d נרצה לייצג וקטור Features שמכיל רק k Features (עבור $k\ll d$) על מנת להוריד את טעות האסטימציה, להוריד את העלות החישובית וגם את עלות השגת המידע.

- בשיטת הפילטרים נעריך כל Feature בנפרד ע"י פונקציית מחיר כלשהי (למשל ה Feature שנותן את ה loss המינימלי הכי גבוה, או מה שממזער את ה loss הריבועי שזה שקול למקדם הקורלציה של פירסון שהוא $\frac{|\langle v-\overline{v},y-\overline{y}\rangle|}{||v-\overline{v}||\,||y-\overline{y}||}$, או להפעיל את מקדם פירסון על הציון של ה Feature v) וניקח את k ה Features הכי טובים.

- בשיטה החמדנית נתחיל מקבוצת Features ריקה וכל פעם נוסיף את ה Feature שייתן את הפרדיקטור הכי טוב ביחד עם ה Features שכבר יש. אלג' OMP עושה זאת בצורה יעילה ע"י מציאת ה Feature שממקסם את הקורלציה עם y בכל שלב. שיטות נוספות הן ע"י בחירת ה Feature שממזער את הגרדיאנט של ה Risk בכל שלב. בנוסף, אלג' AdaBoost הוא בעצם אלג' בחירת Features חמדני עבור $R(w)=\log\left(\sum_{i=1}^m\exp\left(-y_i\sum_jw_jh_j(x_j)\right)\right)$.

- שיטות מזעור הנורמה מתבססות על כך שאנחנו רוצים למזער את $L_S(w)$ תחת אילוץ שאומר שהנורמת-0 של הפתרון שלנו קטנה מ k. כאשר מחליפים את נורמת-0 בנורמת-1 (שהיא באמת נורמה) מקבלים בעית אופטימיזציה קמורה. למה דווקא נורמת-1? כי היא במידה מסוימת הכי "קרובה" לנורמת-0, וגם כי בהנתן פתרון עם נורמת-1 מינימלית אפשר לבנות פתרון אחר עם נורמת-0 קטנה וערך דומה של L_S. בנוסף, נורמת-1 לרוב מעודדת דלילות של הפתרון: מזעור תחת אילוץ על נורמת-1 קטנה הוא כמו רגולריזצית L_1, ואז נוצר מצב ש Feature מקבל משקל שהוא לא 0 אם הקורלציה שלו עם הלייבל היא משמעותית.

- ברגע שיש לנו כבר Features ניתן לעשות מניפולציות קטנות שמורידות את העלות החישובית ולרוב משפרות את השגיאה: אם ל Features יש סקאלות מאוד שונות, נרמול שלהם יתן פתרון הרבה יותר טוב (חלוקה במקסימום או בסטיית תקן).

- סוגים שונים של טרנספורמציות הם: מרכוז - החסרת הממוצע של ה Feature מסוים, הורדת הערכים לקטע $[-1,1]$, הפיכה למשתנה נורמלי, Clipping שהיא בעצם הטרנספורמציה $x\to\text{sign}(x)\min\setminus\max\{b,|x|\}$, טרנספורמציה סיגמואידית כלומר אם Feature מסוים מקבל ערכים בין $(-\infty,\infty)$ ניתן להעביר אותו לסיגמואיד, טנספורמציה לוגריתמית, או טרנספורמציה אונארית (אם ה Feature מקבל רק סט ערכים קבוע).

9. רשתות נוירונים ולמידה מבוססת גרדיאנט

- הפעלת SGD גם על בעיות שהן לא קמורות לרוב עובד בצורה טובה ומביא לפתרון כמעט טוב.

- גרף חישוב היא שפה שנותנת לנו להביע חישוב ולהביע גזירה בצורה נוחה על סמך רכיבים בסיסיים. גרף חישוב הוא גרף DAG בעל מיון טופולוגי שלפיו מוצאים סדר לחישוב.

- כל Node בגרף נקרא שכבה, ובכל שכבה יש פונקציה גזירה (יכולה להיות רב מימדית). דוגמאות לסוגי פונקציות הן: פונקציה אפינית (הכפלה במטריצה W והוספת וקטור b), פונקציה אונארית (לכל איבר מפעילים פונקציה חד מימדית כלשהי) למשל: פונקציית סיגמואיד, פונקציית ReLU שהיא פשוט $\max\{0,x\}$, פונקציה בינארית (למשל חיבור, ℓ^{hinge}, או לוגיסטי $(\log(1+\exp(-y_ix_i))$).

- אלגוריתם Backpropagation מחשב את הנגזרות ע"י מעבר איטרטיבי על כל השכבות מלמטה למעלה ואז מלמעלה למטה.

- שכבה חבויה בגרף חישוב היא שכבה שאין עליה supervision ישיר אלא רק עקיף.

- אם שכבה מסוימת מחשבת פונקציה n מימדית, אז השכבה מכילה n נוירונים.

- רשת עם 0 שכבות חבויות (d נוירוני קלט) ופונקציית סימן יכולה להביע את מחלקת חצאי המרחב על $\mathbb{R}^d$ (חצאי מרחב אפינים אם יש נוירון קלט קבוע נוסף). רשת עם שכבה חבויה אחת יכולה להביע כל פונקציה בוליאנית, אך המחיר הוא בכמות הנוירונים בשכבה החבויה שיכולה להיות אקספ'. רשת עם שכבה חבויה אחת יכולה להביע חיתוכים של חצאי מרחב, בעוד רשת עם שתי שכבות חבויות יכולה להביע איחוד של חיתוכים של חצאי מרחבים.

- באופן כללי, עבור רשת עם k נוירונים סה"כ ופונקציית סימן, העומק לא משפיע על מימד ה VC, ומתקיים $\text{VCdim}(\mathcal{H})=O\left(k^2\log k\right)$.

- רשתות בעומק T (כלומר T שכבות חבויות) עם מספר משתנים שהוא T^2 יכולות ללמוד כל מ"ט שרצה בזמן $T(n)$ עם Sample complexity שהוא פולי' ב T.

- הבעיה של מימוש ERM עבור רשת עם שכבה חבויה אחת בעלת 4 נוירונים היא NP-קשה. אלג' SGD לרוב עובד לחישוב הרשתות גם בבעיות שהן לא קמורות, אך זמן החישוב שלו לרוב מאוד גבוה ואין לו שום הבטחות אופטימליות.

- ניתן לבצע כמה אופטימיזציות לגרף חישוב: נורמליזציה של הקלט, אתחול אקראי של מטריצות המשקולות, mini-batch כלומר בכל איטרציה של SGD נרוץ על k דוגמאות ולא רק 1 (לרוב מימוש מקבילי), הקטנה של η בכל פעם שהחישוב מתחיל "להתקע".

- ישנם כמה סוגים של בעיות שבהן SGD נכשל ולא מצליח ללמוד למרות שהפתרון נמצא במחלקת ההיפותזות.

- ברשתות קונבולוציוניות נעשה קודם כל טרנספורמציה כלשהי לתמונת קלט, כלומר תהיה שכבה של קונבולוציה שמקבלת c תמונות ומוציאה c' תמונות. שכבה נוספת אפשרית היא שכבת $pooling$ (הורדת רזולוציה).

10. למידת חיזוק

- המטרה של למידת חיזוק, שהיא הכללה של למידה מפוקחת, היא ללמוד policy - פונקציה שממפה ממרחב מצבים S למרחב פעולות A. בכל שלב t בתהליך הלמידה המשתמש רואה את המצב $s_t\in S$ ומחליט על פעולה $a_t\in A$ לפי ה policy. הסביבה מספקת reward שהוא $r_t\in\mathbb{R}$ שמתאר כמה ההחלטה הנוכחית טובה או לא (לרוב נניח כי הוא חסום). בסוף השלב זזים למצב הבא s_{t+1} והתהליך ממשיך.

- שני מודלים של regret: או שמנסים למקסם את הגמול הממוצע $R_T(\pi)=\mathbb{E}\left[\frac{1}{T}\sum r_t\right]$, או שמסתכלים על גמול שהולך ויורד עם הזמן, כלומר $R_\gamma(\pi)=\mathbb{E}\left[\sum_{t=1}^\infty\gamma^tr_r\right]$ עבור $\gamma\in(0,1)$.

- בבעיית Multi armed bandit אין מצבים, $A=\{1,\ldots,n\}$. לכל "מכונת מזל" יש התפלגות שונה. $\mu_i=\mathbb{E}[r_t|a_t=i]$, $i^*=\arg\max_i\mu_i,\mu^*=\mu_{i^*},\Delta_i=\mu^*-\mu_i$. ה $Regret$ הוא $\mu^*-\mathbb{E}[R_T(\pi)]$ ורוצים שהוא ילך ל0 ככל ש T גדל. בכל שלב יש שתי אופציות - לבחור במכונה שעד כה נראית הכי טובה, או לבחור במכונה חדשה שעוד לא ראינו.

- השיטה הכי פשוטה להתמודד עם ה tradeoff היא להתחיל מ m צעדים של חיפוש אקראי, ואז בשאר ה $T-m$ צעדים לבחור במכונה שנתנה את ה reward הממוצע הכי טוב. בשיטה זו, אם $m=O\left(n\frac{\log n}{\epsilon^2}\right)$ אזי לכל i מתקיים $|\mu_i-\mu^*|\leq\epsilon$ (הופדינג וחסםהאיחוד).
- בנוסף, ה regret הוא $\mu^*-\mathbb{E}\left[R_T\left(\pi\right)\right]\leq 2\epsilon+\frac{n\log n}{T\epsilon^2}$. עבור ה ϵ הכי טוב מקבלים שה $\text{regret}=\left(\frac{n\log n}{T}\right)^{1/3}$.
- טכניקה נוספת היא חיפוש ϵ-greedy. בעצם אנו רוצים למזער את הפונקציה $L\left(w\right)=-w^T\mu$ עם האילוץ ש $\sum w_i=1$ כאשר $w=\left[0,1\right]^n$. הבעיה היא קמורה עם אילוצים קמורים, הבעיה היא שלא יודעים לחשב את ∇L לכן נבנה משערך בלתי מוטה לגרדיאנט. לכל וקטור התפלגויות p נבחר $i_t\sim p$ ונקבע $\hat{\nabla}L\left(w^{(t)}\right)=\frac{-r_t}{p_{i_t}}e_{i_t}$. נבחר את $p=\left(1-\epsilon\right)w^{(t)}+\epsilon\cdot\frac{1}{n}$ כלומר בסיכוי ϵ נבחר מכונה אקראית ובסיכוי $1-\epsilon$ נשתמש בידע הקודם שלנו. הדבר מבטיח שהנורמה של הגרדיאנט בריבוע לא תהיה גדולה מידי וגם נקבל $\text{regret}=\left(\frac{n}{T}\right)^{1/3}$ (רווח לוגריתמי).
- אלג' נוסף, שדומה ל SGD הוא EXP3: מאתחלים את w להיות וקטור התפלגות אחידה ונעדכן כמו ב SGD רק באקספוננט - $w_i^{(t+1)}=\frac{1}{Z_t}w^{(t)}\exp\left(-\eta\hat{\nabla}L\left(w^{(t)}\right)[i]\right)$. באלג' זה ה $\text{regret}=\left(\frac{n\log n}{T}\right)^{1/2}$. עובד גם במודל adversarial (כשטבע מתנכל לנו).
- טכניקה נוסף היא UCB: ה $\text{UCB}_i\left(t\right)=\hat{\mu}_i+\sqrt{\frac{2\log(T)}{N_i(t)}}\geq\mu_i$, בכל שלב נבחר את הזרוע שממקסמת את ה $\text{UCB}_i\left(t\right)$. ה regret בטכניקה הזאת מתנהג כמו $\frac{\log(T)}{T}$.
- מודל יותר ריאליסטי הוא MDP: לכל t מתקיים $s_{t+1}\sim\tau\left(s_t,a_t\right)$ כאשר τ דטרמיניסטית על $S\times A$, וכן r_t מ"מ מעל $[0,1]$ שתלוי רק ב $\left(s_t,a_t\right)$ וכן $\mathbb{E}\left[r_t\right]=\rho\left(s_t,a_t\right)$. הנחת המרקוביות אומרת שהעבר והעבר בלתי תלוים בהנתן ההווה (ההווה מסכם את כל מה שצריך לדעת על העבר).
- אלג' Value iteration מוצא את π האופטימלי כאשר τ,ρ ידועים כבר (כאשר למשל אפשר להסיק אותם מתוך הפיזיקה). Q-Learning הוא כאשר לא יודעים את τ,ρ.
- באלג' Value iteration מגדירים את ה $V^*:S\to$ optimal value function $\mathbb{R}$ להיות מה תוחלת הrewardשל הפוליסי האופטימלי בהנתן שהתחלנו ממצב s, כלומר $V^*\left(s\right)=\mathbb{E}\left[\sum_{t=1}^{\infty}\gamma^t r_t|s_1=s\right]$. מכאן ניתן להסיק כי מתקיים $V^*\left(S\right)=\max_{a\in A}\left[\rho\left(s,a\right)+\gamma\mathbb{E}_{s'\sim\tau(s,a)}\left[V^*\left(s'\right)\right]\right]$. הגורם הפנימי הוא בעצם $Q^*\left(s,a\right)=\rho\left(s,a\right)+\gamma\mathbb{E}_{s'\sim\tau(s,a)}\left[V^*\left(s'\right)\right]$ שמתאר מה יהיה ה reward אם אני נמצא כרגע במצב s ואני הולך לעשות את הפעולה a ומשם את הפעולות לפי הפוליסי האופטימלי. כלומר ה $\pi^*\left(s\right)=\arg\max_a Q^*\left(s,a\right)$ הפוליסי האופטימלי הוא מה שממקסם את ה Q^* בהנתן המצב הנוכחי. כלומר הפוליסי האופטימלי הוא דטרמיניסטי ביחס ל s_t. האלג' אם כך פשוט מתחיל מווקטור אקראי V_0 ומעדכן $V_{t+1}\left(s\right)=\max_{a\in A}\left[\rho\left(s,a\right)+\gamma\mathbb{E}_{s'\sim\tau(s,a)}\left[V^*\left(s'\right)\right]\right]$ עד להתכנסות. כאשר $V_t=V_{t+1}$ אזי הוא האופטימלי (V^*). $||V_t-V^*||_\infty\leq\gamma||V_0-V^*||_\infty$ כלומר הוא מתכנס מהר יחסית. ההוכחה: מגדירים אופרטור $T^*:\mathbb{R}^{|S|}\to\mathbb{R}^{|S|}$ להיות $V_{t+1}=T^*\left(V_t\right)$ ואז מראים שהאופרטור הוא אופרטור כיווץ (כלומר לכל שני וקטורים u,v מתקיים $||T^*\left(u\right)-T^*\left(v\right)||_\infty\leq\gamma||u-v||_\infty$) ואז משתמשים במשפט נקודת השבת של בנך. כעת לאחר שהתכנסנו ל V^* נוכל לחשב את Q^* ומשם את π^*.
- שיטה נאיבית בהנתן שלא יודעים את τ,ρ היא לשערך את τ,ρ בעזרת "חיפוש אקראי" בעולם, כלומר פוליסי אקראי.
- באלג' Q-Learning מנסים לשערך את פונקציית ה Q^* בלבד (model-free) ע"י ניסיון לפתור את משוואת בלמן המתאימה לה: $Q^*\left(s,a\right)=\rho\left(s,a\right)+\gamma\mathbb{E}_{s'\sim\tau(s,a)}\max_{a'}Q^*\left(s',a'\right)$. בהנתן $\left(s_t,a_t,s_{t+1},r_t\right)$ נגדיר $\delta_{s_t,a_t}\left(Q\right)=Q\left(s_t,a_t\right)-\left(r_t+\gamma\max_{a'}Q\left(s_{t+1},a'\right)\right)$ ואז נאתחל Q_1 ונעדכן $Q_{t+1}\left(s_t,a_t\right)=Q_t\left(s_t,a_t\right)-\eta_t\delta_{s_t,a_t}\left(Q_t\right)\mathbf{1}_{[s=s_t,a=a_t]}$ בניסיון להתכנס למשוואת בלמן. עובד פחות טוב מ Value iteration. בחירת a_t בכל שלב יכולה להעשות באותן טכניקות כמו ב Multi armed bandit.
- האלג' Q-Learning בצורה יותר מפורשת: מאתחלים $Q_0=0$ ואז לכל רביעייה $\left(s_t,a_t,r_t,s_{t+1}\right)$ מעדכנים $Q_{t+1}\left(s_t,a_t\right)=\left(1-\eta_t\right)Q_t\left(s_t,a_t\right)+\eta_t\left(r_t+\gamma\max_a Q_t\left(s_{t+1},a\right)\right)$.
- הבעיה בשני האלג' האלה היא שהסיבוכיות עולה אקספ' עם הממד (גם sample complexity וגם סיבוכיות החישוב). נוכל לעשות אפרוקסימציה לפונקציית ה Q ע"י מחלקת היפותזות שתאופיין בצורה פרמטרית עם פרמטר θ ואז נוכל לבצע את האלג' בצורה יעילה חישובית (למשל עם GD). Deep-Q-Learning זה בעצם לקחת את מחלקת ההיפותזות להיות רשת נוירונים. למשל DeepMind לקחו את הפונקציה $Q_\theta:S\to\mathbb{R}^{|A|}$ להיות רשת שמקבלת כקלט ייצוג d-ממדי (ממשי) של המצב הנוכחי ועם שכבת פלט בגודל $|A|$ מוציאה את הערך של Q לכל אחת מהפעולות האפשריות.
- פתרון אחר לבעיית הממד הוא באמצעות temporal abstraction המשתמש במנגנון שנקרא אופציות: כל פעם בוחרים אופציה (למצוא את הטלפון) שבתורה קוראת לאופציות מתחתיה בתהליך הירכי, עד שהיא מסתיימת ואז חוזרים לתהליך העליון, ועוברים לאופציה הבאה. כלומר מחלקים את הבעיה לתתי בעיות, מייצרים מבנה הירכי על מרחב המצבים ומתמודדים כל פעם רק עם חלק קטן יותר מהמרחב.

11. הורדת ממד

- לוקחים מידע שהוא בממד גבוה ומורידים לממד נמוך: יכול להקטין את זמן האימון וזמן הריצה, מוריד את טעות האסטימציה. טכניקות לינאריות - מכפילים $x\in\mathbb{R}^d$ במטריצה כך ש $Wx\in\mathbb{R}^n,n<d$.
- בשיטות PCA משתמשים לרוב לפני שלומדים על המידע, כשלב עיבוד מקדים. בשיטות הורדת ממד נרצה שלאחר שחזור של $\tilde{x}$ נקבל $x\approx\tilde{x}$. PCA מניח שגם השחזור הוא לינארי. בעצם רוצים למצוא מטריצות W,U המתאימות ל $\arg\min\sum_{i=1}^m||x_i-UWx_i||^2$. הפתרון של PCA טוען כי אם $A=\sum_{i=1}^m\mathbf{x}_i\mathbf{x}_i^T$ אז U (מטריצת השחזור) היא מטריצה שהעמודות שלה הן הווקטורים $u_1,\ldots,u_n$ שהם n הווקטורים העצמיים המתאימים לע"ע הגדולים ביותר של A.
- בהוכחה מראים קודם כל ש $W=U^T$ והעמודות של U הן בסיס א"נ ל $S=\operatorname{Im}\left(WU\right)$ ע"י כך שרואים כי הנקודה $\tilde{x}$ שהכי קרובה ל x על S היא VV^Tx כאשר העמודות של V הן בסיס א"נ, לכן אם ההנחה לא מתקיימת נוכל להחליף את W,U ורק לשפר את מצבנו.
- אנליזה של האלג' מראה לנו גם ששגיאת השחזור שלנו היא בדיוק סכום הע"ע שלא לקחנו.
- ברוב המקרים מורידים את הממוצע של הדוגמאות לפני שמפעילים עליהן PCA.
- במקרה שבו d גדול ממש עושים את הטרנספורמציה הבאה: מגדירים X להיות מטריצה שהשורות שלה הן הדוגמאות, ואז $A=X^TX$. נגדיר את $B=XX^T$ שהיא מממד $m\times m$ שהוא יותר קטן מ $d\times d$. כעת אם $Bu=\lambda u$ אז נקבל כי $A\left(X^Tu\right)=\lambda\left(X^Tu\right)$ לכן $\frac{X^Tu}{||X^Tu||}$ הוא ו"ע של A עם אותו ע"ע. בגלל ש $B_{ij}=\langle x_i,x_j\rangle$ תלויה רק במכ"פ ניתן להחליף אותה בפונקציית קרנל. במקרה שמשתמשים ב B במקום ב A זמן הריצה הוא $O\left(m^3+m^2d\right)$.
- לפעמים הורדת ממד עוזרת גם ליצירת צברים (כמו בדוגמא עם התמונות של האנשים).
- ב Random projections לא אכפת לנו משחזור מוצלח, אלא מעניין אותנו לשמור מרחקים, כלומר שהיחס בין המרחק לפני למרחק אחרי יהיה בערך 1 $\frac{||Wx_i-Wx_j||}{||x_i-x_j||}\approx 1$. נייצר מטריצה $W_{ij}\sim N\left(0,1/n\right)$ ונראה מה יקרה להורדת ממד שתתבצע איתה. התוחלת $\mathbb{E}\left[||Wx||^2\right]=\sum_{i=1}^n\mathbb{E}\left[\left(\langle w_i,x\rangle\right)^2\right]=\sum_{i=1}^n x^T\mathbb{E}\left[w_iw_i^T\right]x=nx^T\left(\frac{1}{n}I\right)x=||x||^2$. יש משפט שאומר שאם $n\geq\frac{\log(|Q|)-\log(\delta)}{\log(\epsilon^2)}$ (סדר גודל, עד קבועים) אז הסיכוי שהורדת ממד מקרית תעוות את המרחקים ביותר מ ϵ קטן מ δ. באופן כללי אם יש לנו n דוגמאות בממד מגודל d אז בפועל הדוגמאות יושבות בתת מרחב מממד n.
- הלמה של ג'ונסון ולינדרשטראוס: תהי Q קבוצה סופית ב $\mathbb{R}^d$, $\delta\in\left(0,1\right)$ ו n מס' טבעי כך ש $\epsilon=\sqrt{\frac{6\log(2|Q|/\delta)}{n}}\leq 3$ אז בסיכוי לפחות $1-\delta$ על בחירת מטריצה מקרית $W\in\mathbb{R}^{n,d}$ עם $W_{i,j}\sim N\left(0,1/n\right)$ מתקיים: $\max_{x\in Q}\left|\frac{||Wx||^2}{||x||^2}-1\right|<\epsilon$. כלומר הטלה מקרית משמרת מרחקים עבור קבוצה סופית של וקטורים. בנוסף, אם הוקטורים יושבים בתוך כדור יחידה, גם המכפלות הפנימיות שלהם נשמרות.
- בשיטת Compressed sensing מניחים כי $x\approx U\alpha$ עבור U אורתונורמלית כך שמספר האינדקסים השונים מ0 ב x הוא s $d>>s$. נשמור את המידע אם כך ע"י מציאת $\alpha=U^Tx$ ושמירת כל האלמנטים שהם לא 0 בוקטור α ואת האינדקסים שלהם (דורש בערך $s\log\left(d\right)$ ביטים).
- Compressed sensingמבטיחים שחזור מושלם לכל הוקטורים שהם $O\left(\frac{n}{\log(d)}\right)$-דלילים. PCA לעומת זאת מבטיח שחזור מושלם אם הדוגמות יושבות בדיוק בתמ"ו מממד n (מוצא תמ"ו מממד n שהכי קרוב לדוגמאות). אם עושים בדיוק compressed sensing $n\log\left(d\right)$ יכול להיות שהם יכשלו הדוגמה הזאת, אך אם נעשה compressed sensing - מכיוון שיש בסיס שבו המידע הזה הוא n דליל, לכן אם ניקח קצת יותר, כלומר $n\log d$ אז נצליח. באופן פרקטי PCA יותר מוצלח מ compressed sensing ברוב המקרים.
- Compressed sensingיותר טוב בדוגמה לעיל מכיוון שבדוגמה זו השחזור הוא לא לינארי, כי זאת אחת ההנחות של PCA (בשחזור לינארי PCA הוא אכן האופטימלי).
- מטריצה W היא $\text{RIP}-\left(\epsilon,s\right)$ אם לכל $0\neq x$ כך ש $||x||_0\leq s$ מתקיים $\left|\frac{||Wx||_2^2}{||x||_2^2}-1\right|\leq\epsilon$. משפט: עבור $\epsilon<1$ תהי W מטריצה שהיא $\text{RIP}-\left(\epsilon,2s\right)$, ו x וקטור כך ש $||x||_0\leq s$ וכן $y=Wx$ ו $\tilde{x}\in\arg\min_{v:Wv=y}||v||_0$ אזי $\tilde{x}=x$. כלומר לפי המשפט נסיק כי תכונת ה RIP משמרת את השחזור המושלם לכל הווקטורים הדלילים. בנוסף, אם נוסיף דרישה ש $\epsilon<\frac{1}{1+\sqrt{2}}$ אז גם נורמת-1 מקיימת את התכונה ל $\tilde{x}$ (כלומר אם $\tilde{x}$ הוא הווקטור עם הנורמה-1 הכי קטנה מבין אלה שמשוחזרים ל x). את הבעיה עם נורמת-1 ניתן לכתוב כבעיית LP. כלומר אפשר לשחזר את כל הווקטורים הדלילים בצורה יעילה אם מורידים אותם מקרית ל $s\log d$ מדידות.

12. כללי

- חוקי log: $\log_b\left(x\right)=\frac{\log_c(x)}{\log_c(b)}$, $\log\left(xy\right)=\log\left(x\right)+\log\left(y\right)$
- מטריצה סימטרית ממשית A נקראת PSD אם לכל $v\in\mathbb{R}^d$ מתקיים $v^TAv\geq 0$, או אם קיימת B כך ש $A=BB^T$, או אם כל הע"ע הם אי"ש.
- bayes optimal classifier בהנתן D הוא $f_D\left(x\right)=\begin{cases}1 & \mathbb{P}\left[y=1|x\right]\geq 0.5\\0 & \text{o.w}\end{cases}$
- ממבחן: SVM עם קרנל RBF יכול להביע את כל הפונקציות מ $\left\{\pm 1\right\}^d\to\left\{\pm 1\right\}$
- עבור r מחלקות היפותזות, ו $d\geq 3$ הוא ה VCdim המקסימלי, אזי $\text{VCdim}\left(\bigcup_{i=1}^r\mathcal{H}_i\right)\leq 4d\log\left(2d\right)+2\log\left(r\right)$. אם $r=2$ נקבל $2d+1\geq$.
- $PAC<=NUL<=consistent$