

הרצאה 9: עוד אלגוריתמים אדפטיביים + תלויות במדגם

Source: Lecture notes by
Aaron Roth and Adam Smith

מרצה: אורי שטמר

הפרדה בין מודל הסטרימינג האדפטיבי והלא-אדפטיבי – המשך

משפט: במודל האדפטיבי, לכל אלגוריתם לבעיית OneFromMany שנכשל בהסתברות לכל היותר $\frac{3}{4}$ יש זיכרון לנארי ב $|X|$ (בהנחה שהסטרים מספיק ארוך)

לצורך הוכחת הטענה הזאת אנחנו נשתמש בטענה הקומבינטורית הבאה:

טענה קומבינטורית: קיימת קבוצה $B \subseteq 2^X$ עם התכונות הבאות:

1. לכל $Y \in B$ מתקיים $|Y| = |X|/2$
2. לכל $Y, Z \in B$ מתקיים $|Y \cap Z| < \frac{49|X|}{100}$
3. $|B| \geq \frac{2^{|X|/5}}{\sqrt{2|X|}}$

אתם יכולים להוכיח את הטענה הקומבינטורית הזאת ע"י טיעון ספירה פשוט יחסית. כרגע בואו נאמין לה.

הוכחת המשפט:

נניח בשלילה שקיים אלגוריתם \mathcal{A} שפותר את בעיית OneFromMany במודל האדפטיבי עם הסתברות שגיאה $\frac{3}{4}$ לכל היותר. תהי B הקבוצה המובטחת מהטענה הקומבינטורית. נתבונן בניסוי המחשבתי הבא:

Input: $Y \in B$

1. For every $x \in Y$, feed algorithm \mathcal{A} the update $(x, 1)$
2. Initiate $\hat{Y} = \emptyset$
3. Repeat $49|X|/100$ times:
 - a. Obtain an outcome $x \in X$ from \mathcal{A}
 - b. Add x to \hat{Y}
 - c. Feed the update $(x, -1)$ to \mathcal{A}
4. In there is an element $Z \in B$ such that $\hat{Y} \subseteq Z$ then return Z . Otherwise return \perp

כעת נשים לב שאם אלגוריתם \mathcal{A} לא נכשל, אז אחרי צעד 3 נקבל ש \hat{Y} היא תת קבוצה של Y בגודל $\frac{49|X|}{100}$.

מכיוון שכל שתי קבוצות ב B מסכימות על פחות מ $\frac{49|X|}{100}$ איברים, אז \hat{Y} לא יכולה להיות תת קבוצה של שום קבוצה ב B פרט ל- Y . לכן, אם אלגוריתם \mathcal{A} מצליח אז בסיום הניסוי המחשבתי הפלט המתקבל הוא $Z = Y$. במקרה כזה נאמר שהניסוי המחשבתי הצליח.

לפי ההנחה שלנו על \mathcal{A} , לכל קלט Y , הניסוי המחשבתי שלנו מצליח בהסתברות לפחות $\frac{1}{4}$

לכן, ישנה קביעה של האקראיות של \mathcal{A} עבורה הניסוי המחשבתי שלנו מצליח עבור לפחות $|B|/4$ מהקלטים האפשריים מתוך B

אחרת, אם נדגום Y יוניפורמית מתוך B אז נקבל ש-

$$\frac{1}{4} \leq \Pr_{r,Y}[\mathcal{A}_r(Y) \text{ succeeds}] = \sum_r \Pr[r] \cdot \Pr_Y[\mathcal{A}_r(Y) \text{ succeeds}] < \sum_r \Pr[r] \cdot \frac{1}{4} = \frac{1}{4}$$

סתירה... כאן r מסמן את האקראיות של \mathcal{A}

כלומר, אחרי שקבענו את האקראיות של \mathcal{A} , אז ישנה קבוצה $B_0 \subseteq B$ בגודל $|B_0| \geq \frac{|B|}{4}$ כך שלכל $Y \in B_0$, אם נריץ את הניסוי המחשבתי שלנו על Y אז הניסוי יצליח והפלט יהיה $Y = Z$.

קעת נשים לב שהמצב הפנימי של אלגוריתם \mathcal{A} בסיום שלב 1 קובע את הפלט של הניסוי המחשבתי שלנו. לכן, מכיוון שישנם לפחות $\frac{|B|}{4}$ פלטים שונים לניסוי, אז חייבים להיות ל \mathcal{A} לפחות $\frac{|B|}{4}$ מצבים פנימיים שונים ולכן הזיכרון שלו חייב להיות לפחות

$$\log\left(\frac{|B|}{4}\right) = \Omega(|X|)$$

דין: בהרצאות האחרונות דיברנו על המחיר בזיכרון הנדרש לאלגוריתמים במודל האדפטיבי לעומת במודל הלא-אדפטיבי. מה לגבי המחיר בזמן ריצה? אנחנו נראה שטרנספורמציה דומה לזאת שראינו עם DP-יציבות תעזור לנו גם לחסוך זמן ריצה. בשביל זה נצטרך את הכלי הבא:

DP-Stability Amplification

שינויי נושא – עכשיו נראה שיטה שמאפשרת לנו להגביר את הבטחת היציבות של אלגוריתם נתון. נניח שיש לנו אלגוריתם \mathcal{A} אשר פועל על דטהבייסים בגודל כלשהו מדומיין X ומבטיח $(1, \delta)$ -DP-יציבות. נגדיר את האלגוריתם הבא:

אלגוריתם \mathcal{B}	
קלט: דטהבייס $S \in X^n$	
(1) בנה דטהבייס $T \subseteq S$ על ידי לקיחת כל איבר $x_i \in S$ באופן בלתי תלוי הסתברות ε	
(2) החזר $\mathcal{A}(T)$	

משפט: אם אלגוריתם \mathcal{A} מקיים $(1, \delta)$ -DP-יציבות, אז אלגוריתם \mathcal{B} מקיים $(\varepsilon, \varepsilon\delta)$ -DP-יציבות.

הוכחה:

נקבע מאורע F ונקבע זוג דטהבייסים שכנים $S, S' = S \cup \{x'\}$

נחשוב על הריצה של \mathcal{B} על S' ונשים לב כי אם $x' \notin T$ אזי הפלט מתפלג בדיוק כמו הפלט בריצה של \mathcal{B} על S . מצד שני, אם $x' \in T$ אזי הפלט מתפלג "דומה" במובן של יציבות, עם פרמטרים $(1, \delta)$. כלומר,

$$\Pr[\mathcal{B}(S') \in F \mid x' \notin T] = \Pr[\mathcal{B}(S) \in F]$$

ובנוסף,

$$e^{-1} \cdot \Pr[\mathcal{B}(S) \in F] - \delta/e \leq \Pr[\mathcal{B}(S') \in F \mid x' \in T] \leq e \cdot \Pr[\mathcal{B}(S) \in F] + \delta$$

לכן

$$\Pr[\mathcal{B}(S') \in F] = (1 - \varepsilon) \cdot \Pr[\mathcal{B}(S') \in F \mid x' \notin T] + \varepsilon \cdot \Pr[\mathcal{B}(S') \in F \mid x' \in T]$$

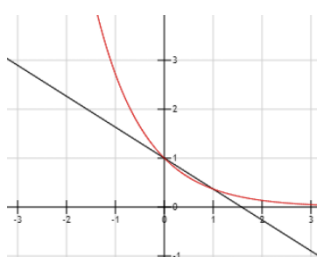
$$\begin{aligned} &\leq (1 - \varepsilon) \cdot \Pr[\mathcal{B}(S) \in F] + \varepsilon \cdot e \cdot \Pr[\mathcal{B}(S) \in F] + \varepsilon \cdot \delta \\ &= (1 + \varepsilon(e - 1)) \cdot \Pr[\mathcal{B}(S) \in F] + \varepsilon \cdot \delta \\ &\leq e^{2\varepsilon} \cdot \Pr[\mathcal{B}(S) \in F] + \varepsilon \cdot \delta \end{aligned}$$

הכיוון השני מתקיים באופן דומה:

$$\begin{aligned} \Pr[\mathcal{B}(S') \in F] &= (1 - \varepsilon) \cdot \Pr[\mathcal{B}(S') \in F \mid x' \notin T] + \varepsilon \cdot \Pr[\mathcal{B}(S') \in F \mid x' \in T] \\ &\geq (1 - \varepsilon) \cdot \Pr[\mathcal{B}(S) \in F] + \varepsilon \cdot e^{-1} \cdot \Pr[\mathcal{B}(S) \in F] - \varepsilon \cdot \delta/e \\ &= (1 - \varepsilon(1 - e^{-1})) \cdot \Pr[\mathcal{B}(S) \in F] - \varepsilon \cdot \delta/e \\ &\geq e^{-\varepsilon} \cdot \Pr[\mathcal{B}(S) \in F] - \varepsilon \cdot \delta/e \end{aligned}$$

(כאשר אי-השוויון האחרון נכון לכל $0 \leq \varepsilon \leq 1$)

מ.ש.ל.



זמן ריצה של אלגוריתמים במודל האדפטיבי

אנחנו חושבים על סיטואציה דומה לזאת של סטרימינג, רק שהפוקוס הוא על זמן ריצה ולא על זיכרון

נניח שיש לנו אלגוריתם \mathcal{A} (במודל הלא-אדפטיבי) עם התכונות הבאות:

- יהי X דומיין כלשהו (הקלטים של האלגוריתם שלנו יהיו איברים מהדומיין X) ותהי $f: X \rightarrow \mathbb{R}$ איזושהי פונקציה שאנחנו מעוניינים לחשב/להעריך על הקלטים שלנו
 - לדוגמה, אולי X הוא קבוצת כל הגרפים ובהינתן גרף $G \in X$ הפונקציה $f(G)$ מחזירה את גודל ה- $global\ min-cut$ בגרף G .
- בחילת הריצה, אלגוריתם \mathcal{A} מקבל את הקלט הראשוני $x_0 \in X$. נסמן את זמן האיתחול של האלגוריתם על הקלט הראשוני כ- t_{init}
 - לדוגמה, גרף
- בנקודת זמן i אנחנו מקבלים עדכון לקלט שלנו $u_i: X \rightarrow X$. אנחנו יכולים לחשוב על u_i כעל פונקציה שממפה קלטים לקלטים מעודכנים, כלומר $x_i \leftarrow u_i(x_{i-1})$. במילים אחרות, בכל שלב i הקלט הנוכחי שלנו מתעדכן. נסמן את זמן הריצה של האלגוריתם לכל עדכון כ- t_{update}
 - לדוגמה, בכל נקודת זמן קשת אחת נוספת/נמחקת מהגרף

- מדי פעם לאורך הריצה, אנחנו מתבקשים להעריך את $f(x_i)$ (כאשר x_i הוא הקלט הנוכחי). נאמר שאלו זמנים שבהם "מתשאלים" את האלגוריתם. נסמן את זמן הריצה של האלגוריתם לכל שאילתא כ t_{query} - לדוגמה, כאשר מתשאלים אותנו עלינו להחזיר קירוב לגודל ה $global\ min-cut$ בגרף הנוכחי
- האלגוריתם פועל במודל הלא-אדפטיבי. כלומר לכל סדרת עדכונים שנקבעה מראש, בה"ג האלגוריתם מחזיר תשובות מדויקות בכל נקודת זמן שבה מתשאלים אותו.

אז נניח שיש לנו אלגוריתם \mathcal{A} עם התכונות הנ"ל (שפועל במודל הלא-אדפטיבי). איך נוכל לתקן אותו כדי שיבטיח נכונות גם במול האדפטיבי? לשם פשטות נניח כי עלינו להחזיר תשובה אחרי כל עדכון.

Algorithm \mathcal{B}

Parameters: Let T be a parameter and denote $k \approx \sqrt{T}$. Let \mathcal{A} be an oblivious algorithm

1. Obtain the initial input x_0
2. Initiate k independent instances $\mathcal{A}_1, \dots, \mathcal{A}_k$ of the oblivious algorithm \mathcal{A} on the initial input x_0
3. For $i = 1, 2, \dots, T$:
 - a) Obtain an update u_i
 - b) Feed the update u_i to all of the copies of \mathcal{A}
 - c) Sample s of the copies of \mathcal{A} and query them to get intermediate outputs $y_{i,1}, \dots, y_{i,s}$
 % s is the number of points we need in order to find an approximate median in a DP-stable way with parameter $\epsilon_0 \approx \frac{1}{100}$. For simplicity let's think about it as a constant
 - d) Output $z_i = \text{DP_Stable_Median}(y_{i,1}, \dots, y_{i,k})$ using stability parameter $\epsilon_0 \approx 1/100$
4. Reset all copies of \mathcal{A} . That is, re-initialize each copy on the current input with fresh randomness, and goto Step 3

Analysis idea:

- We refer to each execution of Step 3 as a phase (consisting of T time steps)
- Fix a specific phase
- Let R denote the collection of random strings used by the copies of \mathcal{A} that are initiated before the beginning of the phase
- Similarly to what we had in the streaming setting, the sequence of all inputs obtained throughout the execution is DP-stable w.r.t. the collection of strings R
 % Here we also make use of the amplification lemma we saw before...
- This implies correctness in the adaptive setting, similarly to the streaming setting

What runtime do we get?

The total time needed for a T time steps is

$$\begin{aligned} &\approx k \cdot t_{\text{init}} + T \cdot k \cdot t_{\text{update}} + T \cdot s \cdot t_{\text{query}} \\ &= \sqrt{T} \cdot t_{\text{init}} + T^{1.5} \cdot t_{\text{update}} + T \cdot t_{\text{query}} \end{aligned}$$

And so the average time per step is

$$\approx \frac{t_{\text{init}}}{\sqrt{T}} + \sqrt{T} \cdot t_{\text{update}} + t_{\text{query}}$$

דוגמה לאפליקציה:

משפט (ללא הוכחה): קיים אלגוריתם לא-אדפטיבי לקירוב הגודל של ה- global min-cut בגרף $G = (V, E)$ עם זמני ריצה

$$\begin{aligned}t_{\text{init}} &= O(|E|) \\t_{\text{update}} &= O(\sqrt{|V|}) \\t_{\text{query}} &= O(1)\end{aligned}$$

מסקנה: קיים אלגוריתם לבעיה זו במודל האדפטיבי עם זמן ריצה ממוצע פר צעד

$$\approx \frac{|E| + T \cdot \sqrt{|V|}}{\sqrt{T}}$$

עבור בחירה של $T \approx \frac{|E|}{\sqrt{|V|}}$ נקבל זמן ריצה ממוצע פר צעד $\sqrt{|E|} \cdot |V|^{1/4}$

הערה: נכון להיום לא ידוע אלגוריתם יותר טוב מזה. צריך להשוות את זה לפתרון הטריטוריאלי שפותר מחדש את הבעיה בכל צעד, ומשיג זמן ממוצע $|E|$ פר צעד...

שינוי נושא:

מענה על שאלות אדפטיביות כאשר יש תלויות במדגם

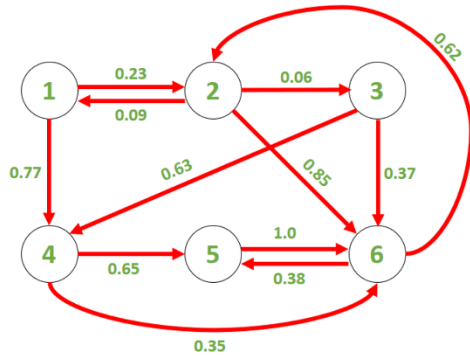
עד היום, כאשר דיברנו בקורס על מענה על שאלות אדפטיביות אז הנחנו שהדטה שלנו נדגם בצורה iid. בחיים האמיתיים, בהרבה מקרים ה- data שאנחנו אוספים איננו נדגם בצורה iid. למשל, סוקר מגיע לבית מסויים, ומבצע את הסקר שלו על כל אחד מתושבי הבית. מה נוכל להגיד ללא ההנחה שה- data נדגם בצורה iid?

ננסה לפרמל את השאלה הזאת. המשחק AdaptiveGame שלנו לא משתנה:

$\text{AdaptiveGame}_{n,k}(A, S, M)$	
1.	המכניזם M מקבל את המדגם S (האנליסט A לא מקבל את המדגם S)
2.	עבור $i = 1, 2, \dots, k$:
	<ul style="list-style-type: none">האנליסט A קובע שאלתא q_i (נניח ש-$q_i \in Q$ עבור משפחה של שאלות אפשריות Q)המכניזם M מקבל את השאלתא q_i ומחזיר תשובה a_iהאנליסט A מקבל את a_i
3.	החזר את הטרנסקריפט של האינטראקציה בין האנליסט למכניזם: $T = (q_1, a_1, q_2, a_2, \dots, q_k, a_k)$

קודם הנחנו שהמדגם S מכיל n דגימות iid מתוך התפלגות כלשהי \mathcal{D} מעל דומיין X . אנחנו עדיין רוצים להניח שהמדגם S מכיל n דגימות, אבל עכשיו ייתכנו תלויות בתוך המדגם. עכשיו אנחנו נניח שהתפלגות המטרה, נקרא לה \mathcal{P} , היא התפלגות מעל n -יות. כלומר \mathcal{P} היא התפלגות מעל X^n .

נשים לב שזה מכליל את ההצגה הקודמת שלנו, כי ייתכן ש- \mathcal{P} היא כן התפלגות מכפלה כאשר כל אחת מ- n הכניסות נדגמות iid מתוך איזושהי התפלגות \mathcal{D} מעל X .



דוגמה: אולי מוגדרת על ידי שרשרת מקרוב. למשל, כדי לדגום מ- \mathcal{P} , נתחיל במצב 1 וכל פעם נתקדם למצב הבא באקראי לפי ההסתברויות שרשומות במעברים. נבצע זאת במשך n צעדים ותוצאת הדגימה שלנו היא סדרת המצבים שביקרנו בהם.

דוגמה נוספת: אולי \mathcal{P} מוגדרת באופן הבא. כדי לדגום מ- \mathcal{P} , נתחיל מלדגום $x_1 \in [0,100]$ בהתפלגות אחידה. אח"כ עבור $i = 2, \dots, n$ נדגום את x_i בהתפלגות אחידה מתוך $\left[\frac{1}{i-1} \sum_{j=1}^{i-1} x_j, \frac{2}{i-1} \sum_{j=1}^{i-1} x_j \right]$.

המסר כאן הוא שעכשיו התפלגות המטרה \mathcal{P} היא התפלגות מעל n -יות ולא מעל יחידונים.

אם כך, בהינתן שאילתא, איך נגדיר את הערך שלה על פני ההתפלגות \mathcal{P} , כלומר, בהינתן שאילתא, מה היינו רוצים להחזיר?

הגדרה 1: עבור התפלגות \mathcal{P} מעל X^n ושאילתא סטטיסטית $q: X \rightarrow [0,1]$ נסמן

$$q(\mathcal{P}) = \mathbb{E}_{S \sim \mathcal{P}}[q(S)] = \mathbb{E}_{S \sim \mathcal{P}} \left[\frac{1}{|S|} \sum_{x \in S} q(x) \right]$$

הערה: היינו יכולים לדבר גם על שאילתות שפועלות על על n -יות, למשל שאילתות עם רגישות נמוכה או שאילתות אחרות. אנחנו נשאר כאן עם שאילתות סטטיסטיות לצורך פשטות.

הצעה: אולי נדרוש את אותו הדבר שדרשנו במקרה ה- iid , רק תחת ההנחה שהמכניזם מקבל מדגם $S \sim \mathcal{P}$. כלומר,

הצעה להגדרה: מכניזם M הוא (α, β) -מדוייק-סטטיסטית-עם-תלויות עבור k שאילתות אדאפטיביות מעל X אם לכל התפלגות \mathcal{P} מעל X^n ולכל אנליסט A מתקיים

$$\Pr_{S \sim \mathcal{P}} \left[\exists i \text{ s.t. } |a_i - q_i(\mathcal{P})| > \alpha \right] \leq \beta$$

$AG_{n,k}(A,S,M)$

האם זאת הגדרה טובה?

זאת הגדרה שאי אפשר לעמוד בה. למשל, אולי \mathcal{P} היא אחת משתי ההתפלגויות הבאות:

התפלגות \mathcal{P}_1	התפלגות \mathcal{P}_0
• בהסתברות 0.5 החזר $(\frac{1}{2}, \frac{1}{2}, \dots, \frac{1}{2})$	• בהסתברות 0.5 החזר $(\frac{1}{2}, \frac{1}{2}, \dots, \frac{1}{2})$
• בהסתברות 0.5 החזר $(1, 1, \dots, 1)$	• בהסתברות 0.5 החזר $(0, 0, \dots, 0)$

מה נענה כאשר אנו מחזיקים את המדגם $(\frac{1}{2}, \frac{1}{2}, \dots, \frac{1}{2})$ ומקבלים את השאילתא $q(x) = x$?

מסקנה: אנחנו חייבים להגביל את הסיפור איכשהו.

שתי גישות עיקריות:

1. נגביל בצורה מפורשת את התלויות בתוך המדגם. אינטואיטיבית, בדוגמה האחרונה היו תלויות "חזקות" מאוד בתוך המדגם. תחת הגבלה "מתאימה" של התלויות, והנחה שהמדגם אכן מגיע מהתפלגות שעומדת בהגבלות האלה, ניתן לתכנן מכניזמים שעונים על שאילתות אדפטיביות.

2. נשווה בין שני העולמות – העולם האדפטיבי והעולם הלא-אדפטיבי. נרצה להצליח לענות על שאילתות אדפטיביות בצורה דומה למה שאנחנו יכולים לעשות בעולם הלא-אדפטיבי. נשים לב שאת הדוגמה האחרונה בעצם גם לא ניתן לפתור בעולם הלא-אדפטיבי ולכן במובן מסויים זה "לא פייר" שננסה בכלל לפתור אותה בעולם האדפטיבי. "אם אי אפשר לפתור משהו בעולם הלא-אדפטיבי, אז איך נפתור אותו בעולם האדפטיבי?"

אנחנו נתרכז היום בגישה 2.

הערה: גם גישה 2 בפועל מגבילה את התפלגות המטרה, אבל בצורה פחות מפורשת ולא דרך מדד תלויות ספציפי.

הגדרה 2: תהי \mathcal{P} התפלגות מעל X^n ותהי $q: X \rightarrow [0,1]$ שאילתא סטטיסטית. נאמר ש- q היא (α, β) -מרוכזת ביחס ל- \mathcal{P} אם

$$\Pr_{S \sim \mathcal{P}} [|q(S) - q(\mathcal{P})| > \alpha] < \beta$$

כלומר, שאילתא q היא מרוכזת ביחס להתפלגות \mathcal{P} אם כאשר דוגמים מדגם $S \sim \mathcal{P}$ אז בהסתברות גבוהה (מעל הגרלת S) מתקיים שהממוצע האמפירי קרוב לתוחלת. אינטואיטיבית, באופן כללי, על שאילתות שהן לא מרוכזות אין לנו דרך טובה לענות גם בעולם הלא-אדפטיבי ולכן זה הגיוני להתמקד בשאילתות מרוכזות.

דוגמה: אם \mathcal{P} היא התפלגות מכפלה כלומר $\mathcal{P} = \mathcal{D}^n$, אזי כל שאילתא סטטיסטית q היא מרוכזת ביחס ל- \mathcal{P} לפי חסם הופדינג.

עוד דוגמה (lazy random walk): אם \mathcal{P} היא ההתפלגות הבאה מעל X^n , אזי כל שאילתא סטטיסטית q מרוכזת ביחס ל- \mathcal{P} .

- תהי \mathcal{D} התפלגות כלשהי מעל X
- כדי לדגום מ- \mathcal{P} נבצע את התהליך הבא:
 - נגדיל $x_1 \sim \mathcal{D}$
 - עבור $i = 2, \dots, n$, בהסתברות $\frac{1}{2}$ נקבע $x_i = x_{i-1}$ ובהסתברות $\frac{1}{2}$ נגדיל $x_i \sim \mathcal{D}$.
 - נחזיר את (x_1, x_2, \dots, x_n) .

הגדרה 3: מכניזם M הוא $(\alpha, \hat{\alpha}, \beta, \hat{\beta})$ -מדוייק-סטטיסטית עבור k שאילתות מרוכזות אדאפטיביות מעל X אם לכל התפלגות \mathcal{P} מעל X^n ולכל אנליסט A ששואל שאילתות $(\hat{\alpha}, \hat{\beta})$ -מרוכזות ביחס ל- \mathcal{P} מתקיים:

$$\Pr_{S \sim \mathcal{P}} [\exists i \text{ s.t. } |a_i - q_i(\mathcal{P})| > \alpha] \leq \beta$$

$AG_{n,k}(A,S,M)$

איך נתכנן מכניזמים שעומדים בהגדרה 3?

הערה: נשים לב שעכשיו אפילו לא ברור שאפשר להשתמש בשיטת ה *Sample Splitting*, מכיוון שבגלל התלויות לא נוכל להפעיל את חסם הופדינג. למשל, אולי כשדוגמים $S \sim \mathcal{P}$ אזי כל S מקודד בתוך ה *low order bits* של איברים ב- S , כך שממוצע אמפירי (מדוייק) על חלק מ S חושף את כל S . דוגמה נוספת: אולי ההתפלגות \mathcal{P} מייצרת t בלוקים משוכפלים, כלומר דוגמת בלוק אחד ואז משכפלת אותו t פעמים...

היום נראה איך אפשר להשתמש בכלים שלמדנו עבור דחיסת-טרנסקריפט גם כאשר יש תלויות במדגם.

תזכורת:

הגדרה 3: נאמר שמכניזם M מאפשר דחיסת-טרנסקריפט ל- $b(n, k)$ ביטים אם לכל אנליסט A קיימת קבוצת טרנסקריפטים אפשריים H_A בגודל $|H_A| \leq 2^{b(n,k)}$ כך שלכל מדגם S מתקיים

$$\Pr[AG_{n,k}(A, S, M) \in H_A] = 1$$

אנחנו נגביל את האנליסט לשאול רק שאילתות שהן מרוכזות ביחס להתפלגות המטרה.

משפט 4 (מקביל למשפט שראינו במקרה ה iid): יהי M מכניזם המאפשר דחיסת-טרנסקריפט ל- $b(n, k)$ ביטים. אזי לכל התפלגות \mathcal{P} ולכל אנליסט A ששואל שאילתות $(\hat{\alpha}, \hat{\beta})$ -מרוכזות ביחס ל- \mathcal{P} מתקיים:

$$\Pr_{S \sim \mathcal{P}}^{AG_{n,k}(A, S, M)} \left[\exists i \text{ s.t. } |q_i(S) - q_i(\mathcal{P})| > \hat{\alpha} \right] \leq \hat{\beta} \cdot k \cdot 2^{b(n,k)}$$

הוכחה: נקבע התפלגות \mathcal{P} ונקבע אנליסט A . נזכור שלאחר שקבענו את A ישנה קבוצה H_A של לכל היותר $2^{b(n,k)}$ טרנסקריפטים אפשריים באינטראקציה בין A ל- M . בכל טרנסקריפט כזה ישנם לכל היותר k שאילתות סטטיסטיות שונות, ולכן ישנם לכל היותר $k \cdot 2^{b(n,k)}$ שאילתות סטטיסטיות אפשריות במהלך הריצה. נסמן קבוצת שאילתות זהו כ- Q_A . מכיוון שכ"א מהשאילתות האלה היא מרוכזת, לפי חסם האיחוד מתקיים

$$\Pr_{S \sim \mathcal{P}} \left[\max_{q \in Q_A} |q(S) - q(\mathcal{P})| > \hat{\alpha} \right] \leq \hat{\beta} \cdot k \cdot 2^{b(n,k)}$$

בפרט,

$$\Pr_{S \sim \mathcal{P}}^{AG_{n,k}(A, S, M)} \left[\exists i \text{ s.t. } |q_i(S) - q_i(\mathcal{P})| > \hat{\alpha} \right] \leq \hat{\beta} \cdot k \cdot 2^{b(n,k)}$$

מ.ש.ל.

משפט 5 (מקביל למשפט שראינו עבור המקרה ה iid): יהי M מכניזם אשר

א. מאפשר דחיסת-טרנסקריפט ל- $b(n, k)$ ביטים

ב. M הוא (α', β') -מדוייק-אמפירית עבור k שאילתות אדאפטיביות

אזי, לכל $\hat{\alpha}, \hat{\beta}$ מתקיים ש- M הוא $(\alpha, \hat{\alpha}, \beta, \hat{\beta})$ -מדוייק-סטטיסטית עבור

$$\alpha = \alpha' + \hat{\alpha} \quad \text{ועבור} \quad \beta = \beta' + \hat{\beta} \cdot k \cdot 2^{b(n,k)}$$

ההוכחה כמעט זהה להוכחה המקבילה למקרה ה iid (פשוט אי-שוויון המשולש...)

קיים מכניזם (Median Mechanism) המקיים:

א. מאפשר דחיסת-טרנסקריפט ל- $b(n, k) = \tilde{O}(\sqrt{n} \cdot \log|X| \cdot (\log k)^{3/2})$

ב. מקיים $(\alpha', 0)$ -דיוק אמפירי עבור $\alpha' = \left(\frac{\ln(4k)}{2n}\right)^{1/4}$

מסקנה 6: לכל $\hat{\alpha}, \hat{\beta}$ קיים מכניזם $(\alpha, \hat{\alpha}, \beta, \hat{\beta})$ -מדוייק-סטטיסטית עבור

$$\alpha = \alpha' + \hat{\alpha} \quad \text{ועבור} \quad \beta = 0 + \hat{\beta} \cdot k \cdot 2^{\tilde{O}(\sqrt{n} \cdot \log|X| \cdot (\log k)^{3/2})}$$

דיון לגבי מסקנה 6:

בעצם, הקנס הגדול שאנחנו משלמים כאן הוא בכך שהסתברות השגיאה β "מתפוצצת". זאת אומרת שכדי לקבל משהו מועיל אנחנו צריכים להניח שהשאלות שהאנליסט שואל הן מאוד מרוכזות ביחס להתפלגות המטרה. בפרט אנחנו צריכים שיתקיים $\dots \hat{\beta} \ll 2^{-\sqrt{n}}$

זה נכון כאשר התפלגות המטרה היא התפלגות מכפלה: אם $\mathcal{P} = \mathcal{D}^n$ אזי לפי הופדינג, לכל שאלתא סטטיסטית $q: X \rightarrow [0,1]$ ולכל $\hat{\beta} > 0$ מתקיים

$$\Pr_{S \sim \mathcal{P}} \left[|q(S) - q(\mathcal{P})| > \sqrt{\frac{\ln(2/\hat{\beta})}{2n}} \right] \leq \hat{\beta} \quad (1)$$

כלומר במקרה ה iid כל שאלתא סטטיסטית היא $(\hat{\alpha}, \hat{\beta})$ -מרוכזת עבור $\hat{\alpha} = \sqrt{\frac{\ln(2/\hat{\beta})}{2n}}$

כדי לקבל משהו משמעותי ממסקנה 6 נרצה לדאוג ש- $\hat{\beta}$ יהיה מספיק קטן. ספציפית נרצה

$$1 \gg \hat{\beta} \cdot k \cdot 2^{\tilde{O}(\sqrt{n} \cdot \log|X| \cdot (\log k)^{3/2})}$$

כלומר נרצה (בערך)

$$\hat{\beta} \ll \frac{1}{k} \cdot 2^{-\sqrt{n} \cdot \log|X| \cdot (\log k)^{3/2}}$$

נציב את הדרישה הזאת ב- (1) ונקבל שכל שאלתא סטטיסטית היא מרוכזת עבור

$$\hat{\alpha} \approx \sqrt{\frac{\ln(2k) + \sqrt{n} \cdot \log|X| \cdot (\log k)^2}{2n}} \approx \sqrt{\frac{\log|X| \cdot (\log k)^2}{\sqrt{n}}}$$

לפי מסקנה 6, זה נותן לנו דיוק

$$\alpha \approx \left(\frac{\ln(4k)}{2n}\right)^{1/4} + \sqrt{\frac{\log|X| \cdot (\log k)^2}{\sqrt{n}}}$$

כלומר, השגיאה יורדת עם n .

הערות:

1. זה בדיוק מה שקיבלנו עבור ה *MedianMechanism* גם כשניתחנו אותו ישירות עבור המקרה ה *iid*. כלומר הכללנו את האנליזה בלי להפסיד כלום (כאשר מנוונים את התוצאה חזרה למקרה ה *iid*).
2. אמנם החשבון הזה היה רק עבור התפלגויות מכפלה $\mathcal{P} = \mathcal{D}^n$, אבל הדבר היחיד שהשתמשנו בו כאן היה ההנחה שהשאלות שהאנליסט שואל הן ממש מרוכזות. המסקנה היא שכל עוד השאלות שהאנליסט שואל הן ממש מרוכזות אז נוכל להבטיח שגיאה לא טריוויאלית גם כאשר יש תלויות במדגם.