21.06.2023 **ניתוח מידע אדפטיבי**

הרצאה 10: תוצאות שליליות

Source: Lecture notes by
Aaron Roth and Adam Smith

מרצה: אורי שטמר

בקורס שלנו ראינו כמה שיטות שמאפשרות לענות על שאילתות סטטיסטיות אדפטיביות. בפרט, ראינו:

- אלגוריתם יעיל חישובית אשר עונה על k שאילתות אדאפטיביות בעזרת מדגם בגודל
$$n \gtrsim \sqrt{k}$$
- אלגוריתם לא יעיל חישובית אשר עונה על k שאילתות אדפטיביות בעזרת מדגם בגודל
$$n \gtrsim (\log k)^3 \cdot (\log|X|)^2$$

האם יש אלגוריתם יעיל אשר משיג תוצאות טובות משמעותית מהאלגוריתם שראינו? בהרצאה הזאת נראה שבאופן כללי התשובה היא לא.

<u>הרעיון:</u> אנחנו נראה תוקף (אנליסט) אשר מדבר עם מכניזם יעיל שעונה על שאילתות. בעזרת (בערך) n^2 שאילתות, התוקף שלנו יצליח לשחזר כמעט את כל המדגם שהמכניזם מחזיק. כפי שכבר ראינו בקורס שלנו, לאחר שהתוקף הצליח לשחזר את המדגם הוא יוכל בקלות למצוא שאילתא "רעה".

בשביל זה, נצטרך להכיר את הכלי הקריפטוגרפי הבא:

<u>Fingerprinting codes (FPC)</u>

זהו כלי קריפטוגרפי שתוכנן כדי לנסות ולמנוע הפצה לא חוקית של תוכן דיגיטלי. נניח שכל יום נטפליס משדרת למנויים שלה פרק חדש בסדרה מסויימת. נטפליקס לא רוצים שמשתמשים רעים יורידו את הפרק ויעלו אותו לאינטרנט. בנסיון למנוע זאת, הם יכולים לשתול סימנים בלתי נראים (watermarks) בפרקים שהיא משדרת, כך שכל מנוי יקבל עותק שונה עם סימונים שונים. כעת, אם נטפליקס תראה את הפרק באינטרנט, היא תוכל לדעת מי מהמשתמשים הפיץ את הפרק.

אבל מה קורה אם <u>כמה</u> מנויים מחליטים לשתף פעולה ולנסות להשוות בין העותקים שלהם על מנת להסיר את הסימונים, ואז מעלים עותק "מעובד" לאינטרנט?

את הבעייה הזאת FPC מנסה לפתור. הרעיון מסתמך על ההנחה הבאה:

- אם שני משתמשים רעים מנסים להשוות את העותקים כדי להסיר סימונים, אז הם יוכלו לזהות רק מיקומות שבהם הסימונים שלהם <u>שונים</u>. לעומת זאת, אם במיקום מסויים לשניהם יש את אותו הסימון בדיוק, אז הם לא יוכלו לשים לב שיש שם סימון בכלל ולא יסירו את הסימון.
- באותו אופן, אם קבוצה של n משתמשים רעים מנסים לשלב כוחות ולהסיר את הסימונים מהעותקים שלהם, הם לא יוכלו לזהות סימונים אשר משותפים <u>לכולם</u>. כלומר, אם בנקודה מסויימת בקובץ יש <u>לכולם</u> בדיוק את אותו הסימון – אז הסימון הזה ישאר גם בעותק "המעובד" שהם ינסו לייצר.

תחת ההנחות האלה, FPC מאפשר לנטפליקס לזהות את המשתמשים הרעים.

עם קצת יותר פרטים:

- ישנם N משתמשים, מתוכם n משתמשים החליטו לשתף פעולה ולהעלות פרקים לאינטרנט
- בכל יום:
  - נטפליקס משדרת לכ"א מהמשתתפים פרק חדש עם סימונים שונים
  - n המשמשים הרעים משלבים כוחות ומנסים להסיר/לשנות את הסימונים בעותקים שלהם, תחת ההגבלה שסימונים אשר זהים אצל כל n המשתמשים הרעים לא ימחקו, ומעלים את העותק המעובד שהם יצרו לאינטרנט.
    - למעשה, פורמלית, אנחנו נדרוש (לכאורה) פחות מהמשתמשים הרעים: סימונים אשר זהים אצל כל N המשתמשים לא ימחקו (בפרט, אם סימונים אשר זהים אצל n הרעים לא ימחקו, אז ברור שגם סימונים שזהים אצל כל N המשתמשים לא ימחקו).
  - נטפליקס רואה את הפרק המעובד באינטרנט
- לאחר ℓ אינטציות כאלה, נטפליקס מסוגלים לזהות (כמעט) כל אחד מהמשתמשים הרעים

באופן פורמלי, נחשוב על המשחק הבא בין תוקף $\mathcal{P}$ ששולט על n מתוך N משתמשים לבין אלגוריתם $\mathcal{F}$ (זהו אלגוריתם ה FPC) אשר מנסה לזהות את n המשתמשים הנשלטים ע"י היריב.

Algorithm $\boldsymbol{IFPC(\mathcal{P},\mathcal{F})}$

(1) Denote $\mathrm{ALIVE}^1 = [N]$
(2) The Adversary $\mathcal{P}$ *selects a subset of users* $S^1 \subseteq [N]$ *(unknown to* $\mathcal{F}$*)*
(3) For $j = 1$ *to* ℓ *do*
  a. *Algorithm* $\mathcal{F}$ *outputs a vector* $c^j \in \{-1,1\}^N$
  b. *Let* $c^j_{S^j}$ *be the restriction of* c^j *to coordinates in* S^j*. This is given to* $\mathcal{P}$
  c. *The adversary* $\mathcal{P}$ *outputs* $a^j \in \{-1,1\}$*. This is given to* $\mathcal{F}$
  d. *Algorithm* $\mathcal{F}$ *accuses a set of users* $I^j \subseteq [N]$*.*
  e. *Let* $\mathrm{ALIVE}^{j+1} = \mathrm{ALIVE}^j \setminus I^j$*, and let* $S^{j+1} = S^j \setminus I^j$
(4) The Adversary $\mathcal{P}$ *wins if it was consistent throughout the game. Formally,* $\mathcal{P}$ *is consistent if for every* $j \in [\ell]$ *there exists an index* $i \in \mathrm{ALIVE}^j$ *such that* $a^j = c^j_i$

<u>הגדרה 1:</u> אלגוריתם $\mathcal{F}$ הוא *FPC* עם פרמטרים n, N, ℓ, ε אם לכל תוקף $\mathcal{P}$ מתקיימים התנאים הבאים:

$$\Pr\begin{bmatrix}\text{משתמש טוב}\\ \text{מואשם כרע}\end{bmatrix} \leq \varepsilon$$

$$\Pr[\text{התוקף } \mathcal{P} \text{ מנצח}] = \Pr\begin{bmatrix}\text{התוקף } \mathcal{P} \text{ מצליח}\\ \text{להיות עקבי לאורך}\\ \text{כל הריצה}\end{bmatrix} \leq \varepsilon$$

כאשר ההסתברויות הן מעל הרצת אלגוריתם *IFPC*

מה קורה פה?
לאורך הריצה אנחנו מאשימים משתמשים כרעים ואז הם "יוצאים מהמשחק" (ובפרט יוצאים מהקבוצה S^j).
אנחנו דורשים שבהסתברות גבוהה, אף משתמש הגון לא יואשם בתור משתמש רע.
בנוסף, אנחנו דורשים שמתישהו במהלך הריצה, מכיוון שמשתמשים רעים נמחקים, התוקף $\mathcal{P}$ לא יוכל לעמוד יותר בדרישת העקביות. מכיוון שאנחנו מניחים שתוקפים לא יכולים ייצר "פרקים מעובדים" מבלי לעמוד בדרישת העקביות, זאת אומרת שהתוקפים לא מסוגלים יותר לייצר פרקים מעובדים.

משפט 2: לכל $1 \leq n \leq N$ קיים *FPC* עם פרמטרים n, N, ℓ, ε כאשר $\varepsilon = \frac{1}{N}$ וכאשר
$$\ell = O(n^2 \cdot \log N)$$

הוכחה חלקית של משפט 2:
אנחנו נראה הוכחה חלקית של המשפט הנ"ל, עם פרמטר $\ell = \text{poly}(n)$ במקום $\ell \approx n^2$.

המטרה הנוכחית שלנו היא לבנות אלגוריתם $\mathcal{F}$ עבור משחק ה $IFPC$ הנ"ל. זכרו כי, באופן אינטואיטיבי, לאורך המשחק אלגוריתם $\mathcal{F}$ מנסה "להאשים" יותר ויותר שחקנים "רעים", עד אשר (בתקווה) הוא יאשים את כל השחקנים הרעים. אנחנו נתחיל מלתכנן אלגוריתם (נקרא לו $\mathcal{F}_1$) אשר מטרתו תהייה להאשים משתמש רע אחד מתוך קבוצת המשתמשים הרעים. לאחר מכן נשתמש באלגוריתם זה בצורה איטרטיבית על מנת לזהות משתמשים רעים נוספים.

אלגוריתם $\mathcal{F}_1$

(1) תהי Z המטריצה הבאה:

$$\begin{pmatrix} \textbf{0 block} & \textbf{1st block} & \textbf{2dn block} & & \textbf{nth block} \\ 000\ldots000 & 111\ldots111 & 111\ldots111 & & 111\ldots111 \\ 000\ldots000 & 000\ldots000 & 111\ldots111 & & 111\ldots111 \\ 000\ldots000 & 000\ldots000 & 000\ldots000 & \cdots & 111\ldots111 \\ \vdots & \vdots & \vdots & & 111\ldots111 \\ 000\ldots000 & 000\ldots000 & 000\ldots000 & & 111\ldots111 \end{pmatrix}$$

במטריצה הזאת יש $(n+1)$ סוגי עמודות, כאשר מכל סוג עמודה ישנם $\tilde{O}(n^2)$ העתקים. כלומר בסה"כ ישנם $\tilde{O}(n^3)$ עמודות במטריצה הזאת.

(2) תהי Y מטריצה המתקבלת לאחר פרמוטציה אקראית של העמודות במטריצה Z.

(3) במשך $\ell = \tilde{O}(n^3)$ סיבובים, אלגוריתם $\mathcal{F}_1$ פולט (בזה אחר זה) את העמודות של המטריצה Y בתור הוקטורים של ה *watermarks* שהוא אמור לפלוט במשחק ה *IFPC*.

(4) נסמן ב $\vec{a} \in \{0,1\}^{\ell}$ את וקטור התשובות שאלגוריתם $\mathcal{F}_1$ קיבל מהיריב במהלך שלב (3).

(5) עבור כל סוג עמודה/בלוק i, סמן:
$\text{count}_i =$ מספר העמודות מסוג i שעליהן היריב החזיר תשובה 1.
שימו לב כי זהו מספר בין 0 ל- $\tilde{O}(n^2)$.

(6) החזר אינדקס $i \in [n]$ (כלומר האשם משתמש) שעבורו מתקיים
$$\text{count}_i > \text{count}_{i-1} + \tilde{O}(n)$$

תחילה נראה ששלב (6) מוגדר הייטב, כלומר שאינדקס i כנ"ל קיים. נזכור שאנחנו ניחים שהיריב צריך להיות עקבי (אחרת אנחנו מנצחים ולא אכפת לנו). לכן, על כל העמודות מסוג 0 היריב חייב להחזיר תשובה 0 ועל כל העמודות מסוג n היריב חייב להחזיר תשובה 1. כלומר,
$$\text{count}_0 = 0$$
$$\text{count}_n = \tilde{O}(n^2)$$
לכן, מכיוון שישנם $(n+1)$ סוגי עמודות, חייב להיות אינדקס i העונה לתנאי משלב (6) באלגוריתם.

עכשיו נראה שבה"ג מתקיים שאינדקס *i* המקיים את התנאי הזה הוא אינדקס של משתמש "רע". לצורך כך נקבע אינדקס *i* של משתמש "טוב" ונראה שההסתברות ש *i* יקיים את התנאי הנ"ל היא זניחה. כדי לראות את זה, נשים לב שמכיוון שמשתמש *i* הוא "טוב" אז היריב לא יכול להבחין בין עמודות מבלוק *(i-1)* לבין עמודות מבלוק *i*, כי השורה היחידה שמבדילה ביניהם זאת שורה *i* והיריב אף פעם לא רואה את הכניסות שלה. לכן, אינטואיטיבית, התשובות של היריב חייבות להתנהג אותו דבר על פני שני הבלוקים האלה, ולא יכולה להיות "קפיצה" של *1/n* בספירות.

בצורה יותר פורמלית: נקבע את התשובות של היריב ואת כל הפרמוטציה, פרט לחלוקה בין העמודות (אחרי הפרמוטציה) ששייכות לבלוק *(i-1)* והעמודות ששייכות לבלוק *i*. מכיוון שהיריב לא רואה את השורה ה *i*, החלוקה הזאת בין העמודות של *(i-1)* ושל *i* היא עדיין אקראית לחלוטין גם בהינתן כל התשובות של היריב. נסמן ב- $\vec{a}_i$ את אוסף התשובות של היריב שמתאימות לעמודות מבלוקים *i-1, i*. כעת, כל אחד מבין count_i ו- count_{i-1} זה סכום של חצי אקראי מהביטים ב $\vec{a}_i$. לכן, לשני המשתנים האלה יש בדיוק את אותה התוחלת, ולפי חסם הופדינג בהסתברות עצומה הסכום לא יסטה ביותר מ $\tilde{O}(n)$ מהתוחלת שלו (זה סכום של $\tilde{O}(n^2)$ ביטים). לכן, ההסתברות שתהייה קפיצה גדולה עבור איזשהו משתמש *i* "טוב" היא זניחה.

איפה אנחנו עומדים בהוכחה של משפט 2?

ראינו כי בעזרת $\tilde{O}(n^3)$ סיבובים נוכל להאשים משתמש "רע" אחד. נריץ את האלגוריתם הזה במשך n פעמים. כל הרצה מתבצעת רק על הקבוצה של המשתמשים שעדיין "בחיים", ובכל פעם נמחוק משתמש אחד מהקבוצה של המשתמשים "החיים". סך הכל לאחר $\tilde{O}(n^4)$ סיבובים נזהה את כל המשתמשים הרעים (או שהיריב "יפסיד" מתישהו בדרך, כלומר לא ישמור על דרישת העקביות, ואז גם ננצח...)

מ.ש.ל. (משפט 2)

עכשיו נראה איך אפשר להשתמש ב *FPC* כדי להראות תוצאות שליליות עבור מכניזמים שעונים על שאילתות. ספציפית, אנחנו נראה שמכניזמים יעילים לא יכולים לעונות על יותר מ n^2 שאילתות בעזרת מדגם בגודל n.

הנחה מפשטת: המכניזם שעונה על שאילתות מקיים את התכונה הבאה:

<u>הגדרה:</u> מכניזם שעונה על שאילתות הוא "טבעי" אם כשהוא מחזיק מדגם $S \in X^n$ ומקבל שאילתא $q: X \to \{\pm 1\}$ אז הפלט שלו לא תלוי בערכים של השאילתא מחוץ למדגם. כלומר, לכל 2 שאילתות q, q' אשר מסכימות בדיוק על איברי המדגם S, התשובה המחוזרת ע"י האלגוריתם היא זהה (או מתפלגת אותו דבר).

<u>משפט:</u> יהי M מכניזם טבעי שעונה על k שאילתות אדפטיביות מעל דומיין בגודל $|X| \geq 2000n$ בעזרת מדגם בגודל n. אזי $n = \Omega(\sqrt{k})$.

<u>סקיצת הוכחה:</u>
נסתכל על המתקפה הבאה. התוקף שלנו מגדיר התפלגות מטרה $\mathfrak{D}$ ואז המכניזם מקבל מדגם מההתפלגות הזאת (התוקף לא רואה את המדגם). לאחר מכן התוקף שואל (בערך) n^2 שאילתות אדפטיביות, כאשר לפחות עבור אחת מהן המכניזם לא יצליח לענות בצורה מדוייקת.

Attack against a natural mechanism M
Setup: • *Given parameter n, let $N = 2000n$ and let $\mathfrak{D}$ be the uniform distribution over $\{1, \dots, N\}$* • *Let S be a sample containing n iid elements from $\mathfrak{D}$. The sample S is given to the mechanism M (but not to the adversary)* *The Attack:* *(1) Initialize an $(n, N, \ell, \varepsilon)$-FPC algorithm $\mathcal{F}$, where $\ell = \tilde{O}(n^2)$ and $\varepsilon \leq \frac{1}{N}$* *(2) Let $T^0 = \emptyset$* *(3) For $j = 1$ to ℓ do* *a. Let $c^j \in \{\pm 1\}^N$ be the vector chosen by $\mathcal{F}$* *b. Define the query $\hat{q}^j$ such that $\hat{q}^j(i) = \begin{cases} c_i^j & , \text{ if } i \notin T^{j-1} \\ 0 & , \text{ else} \end{cases}$* *c. Ask $\hat{q}^j$ to the mechanism M and obtain answer a^j. Round a^j to $\overline{a}^j \in \{\pm 1\}$* *d. Give $\overline{a}^j$ to $\mathcal{F}$ and let $I^j \subseteq [N]$ be the set of accused users. Let $T^j \leftarrow T^{j-1} \cup I^j$*

תחילה נשים לב כי לפי ההבטחות של *FPC*, בהסתברות לפחות $1 - \frac{1}{N}$, בכל סיבוב j מתקיים $I^j \subseteq S$, מכיוון שאנחנו מניחים שהאשמות שגויות קורות בהסתברות $\frac{1}{N}$ לכל היותר). במקרה כזה, לכל j מתקיים $\left|T^j\right| \leq n$. נניח שזה אכן המצב.

בנוסף, לפי ההבטחות של *FPC*, בהסתברות לפחות $1 - \frac{1}{N}$, קיים סיבוב j שבו התשובה $\overline{a}^j$ היא <u>לא עקבית</u>.
בפרט, ז"א שבזמן j הווקטור c^j חייב להיות או זהותית *1-* או זהותית 1 (כי אם הוקטור הזה מכיל גם אפסים וגם אחדים, אז פי הגדרה, $\overline{a}^j$ לא יכול להיות לא עקבי...).

זה אומר שבזמן j מתקיים אחד משני דברים:
מקרה 1: או ש $\overline{a}^j = -1$ אבל $c^j \equiv 1$
מקרה 2: או ש $\overline{a}^j = 1$ אבל $c^j \equiv -1$

בכל מקרה, זה אומר שהתשובה המוחזרת ע"י המכניזם רחוקה מאוד מהתוחלת של השאילתא באותו סיבוב.
ספציפית, במקרה 1 מתקיים

$$\hat{q}^j(\mathfrak{D}) \geq 1 - \frac{\left|T^j\right|}{N} \geq 1 - \frac{n}{N} \geq 1 - \frac{1}{2000}$$

ובמקרה 2 מתקיים

$$\hat{q}^j(\mathfrak{D}) \leq -1 + \frac{\left|T^j\right|}{N} \leq -1 + \frac{n}{N} \leq -1 + \frac{1}{2000}$$

נניח בשלילה שהתשובה של המכניזם בסיבוב j מדוייקת עד כדי טעות $\alpha = \frac{1}{2}$.
אזי במקרה 1 נקבל $a^j > 0$ ובמקרה 2 נקבל $a^j < 0$ ובכל מקרה ז"א ש- $\overline{a}^j$ צריך להיות עקבי. סתירה.

<u>שאלה:</u> איפה השתמשנו בהנחה המפשטת שהמכניזם הוא "טבעי"?

<u>תשובה:</u> ההבטחות של *FPC* הם רק תחת כללי המשחק של *IFPC*. נקודה חשובה היא שבמשחק של *IFPC* היריב $\mathcal{P}$ רואה רק את הערכים $c_{S^j}^j$, כלומר רואה רק את הקואו' של הוקטור c^j שמתאימות לאיברים ב- S שעדיין לא הואשמו כרעים. לעומת זאת, במתקפה האחרונה שראינו, אנחנו נותנים למכניזם M את <u>כל</u> הוקטור c^j

(הווקטור הזה מגדיר את השאילתא הסטטיסטית שאנו נותנים למכניזם). באופן כללי, זה מחוץ לכללי המשחק של *IFPC* וזה שובר את ההבטחות שלו. אבל, תחת ההנחה המפשטת שלנו – המכניזם הוא "טבעי" – אנחנו יודעים שהוא "לא מסתכל" בערכים של השאילתא הזאת מחוץ למדגם, כלומר "לא מסתכל" על קואו' של c^j שהוא לא אמור לראות, ולכן המתקפה עובדת.

<u>הערה:</u> לא הייתה לנו כאן שום הנחה על כח החישוב של המכניזם. כלומר, בעצם מה שהראינו זה שמכניזם שעונה על שאילתות, אם הוא "טבעי", אז אפילו אם הוא לא מוגבל חישובית הוא לא יכול לענות על יותר מ n^2 שאילתות.

<u>הערה:</u> כדי להיפטר מההנחה המפשטת הזאת, אפשר להשתמש בהצפנות על מנת להסתיר מהמכניזם את התוכן של השאילתא בקואו' שהוא לא אמור לראות. זה אפקטיבית מכריח את המכניזם להתנהג כמו מכניזם "טבעי" ואז המתקפה עדיין עובדת. (אבל עכשיו אנחנו צריכים להניח שהמכניזם מוגבל חישובית, אחרת הוא יכול לשבור הצפנות ולא נוכל להסתיר ממנו את הקואו' האלה...)