מבוא לתורת הקודים לתיקון שגיאות

ארזים ©

1 אלגוריתמים לתיקון שגיאות

נראה היום:

- תיקון מחיקות
- אלגוריתם יעיל לתיקון שגיאות עד חצי המרחק בקודי RS

1.1 תיקון מחיקות

בהינתן מילת קוד $v \in \{0,1,?\}$, מתי יש מילת קוד שמסכימה עם v על הקואורדינטות שלא נמחקו?
במקרה שבה כמות המחיקות גדולה מהמרחק בקוד, אם יש 2 או יותר מילות קוד כאלה, גם המרחק בין 2 מילות קוד הוא מילת קוד ולכן נקבל שיש מילה בה כל הקואורדינטות שאינן מתאפסות מוכלת בקואורדינטות שלא נמחקו.

טענה 1.1 תבנית מחיקות היא "רעה" (שתי מילות קוד נמחקות לאותו מילה) אם ורק אם יש מילת קוד $(0 \neq)$ שכל הקואורדינטות ההשונות מאפס מוכלת בקואורדינטות שנמחקו.

משפט 1.2 יהי Cקוד לינארי ו H ה־ PCM של C. אזי ניתן לתקן מחיקות בתת קבוצה $S \subset [n]$ של הקואורדינטות אם ורק אם העמודות ב H המתאימות לקואורדינטות ב S הם בת"ל.

הוכחה: העמודות של H הן בלתי תלויות אמ"ם יש צירוף לא טריוויאלי שלהן ששווה לאפס. זה קורה אמ"ם יש מילה v מאורך n השונה מאפס ושכל הקואורדינטות שלו שאינן 0 מוכלות ב־S, כך ש $Hv = 0$ וזה קורה אמ"ם יש $0 \neq v \in C$ הנתמך ב־S ($\{i : v_i \neq 0\} \subset S$). ■

טענה 1.3 אם ניתן לתקן מחיקות בקבוצה Sאז ניתן זאת ביעילות (פולינומיאלי) .

הוכחה: יהי S קבוצת המחיקות של מילת קוד $v \in S$ ונניח בה"כ כי המחיקות בסוף $v = (v_1, \ldots, v_m, ? \ldots, ?)^t$.

$$0 = H \cdot v$$

$$0 = \begin{pmatrix} | & | & & & | \\ c_1 & c_2 & \cdots & \cdots & c_n \\ | & | & & & | \end{pmatrix} \begin{pmatrix} v_1 \\ \vdots \\ v_m \\ ? \\ \vdots \\ ? \end{pmatrix} \begin{matrix} x_1 \\ \vdots \\ x_{|S|} \end{matrix}$$

ולכן אנו מחפשים $x_1, \ldots x_{|S|}$ כך ש־

$$0 = v_1 \cdot c_1 + \cdots + v_{n-|S|} c_{n-|S|} + x_1 c_{n-|S|+1} + \cdots + x_{|S|} c_n$$

אם נסתכל על הצמצום של H לעמודות S אזי נקבל מערכת משוואות:

$$H|_S \cdot x = -H|_{[n]\setminus S} \cdot \begin{pmatrix} v_1 \\ \vdots \\ v_{[n]-S} \end{pmatrix}$$

קיבלנו מערכת לינארית ומובטח לנו שיש פיתרון מהטיעונים מקודם על המרחק. ■

מסקנה 1.4 ניתן לתקן כל תבנית מחיקות ממשקל $\mathrm{dist(C)} >$, מספר העמודות התלויות ב־Hשווה ל distance של הקוד!

1.2 תיקון טעויות בקוד RS

קוד RS הוא קוד $[n,k,n-k+1]$, $q \geq n$ עבור $\grave{deg}(f) < k \Rightarrow (f(\alpha_1),\ldots,f(\alpha_n))$. בהינתן $f(\alpha_1),\ldots,f(\alpha_n)$, ניתן למצוא את מקדמי f ע"י אינטרפולציה.

שאלה: איך עושים אינטרפולציה כשיש לנו קצת רעש?

משפט 1.5 יש אלגוריתם יעי לתיקון t טעויות ו־m מחיקות בקוד RS כל עוד $2t+m < n-k+1$.

רעיון: אם אנחנו יודעים את המיקום של הטעויות אז אנחנו במקרה של מחיקה.

הגדרה 1.6 (polynomial error locator) נניח כי מילת הקוד הוא $\bar{v} = (f(\alpha_1),\ldots,f(\alpha_n))$, והמילה עם הטעויות היא $\bar{y} = (m_1,\ldots,m_n)$ (ואין מחיקות) אז $E(x)$ הוא ELP עבור $\bar{v},\bar{y}$ אם:

1. $\deg E(x) \leq t$ $(\mathrm{dist}(\bar{y},\bar{v}) \leq t)$.
2. אם $y_p \neq f(\alpha_p)$ אז $E(\alpha_p) = 0$.

למשל: $E(x) = \prod_{s=1}^{t'}(x-\alpha_{p_s})$, $t' \leq t$ עבור קואורדינטות בהן נפלק טעויות $p_1,\ldots,p_{t'}$. נניח שנתון Eשהוא ELP עבור v,y.

גישה 1: נניח שנתון $\mathrm{Enc}(f) = \bar{v},\bar{y}$ ונתון ELP, $E(x)$ (לדברי המרצה, $\bar{v},E$ לא באמת נתונים לנו). נסתכל על הפולינום $N(x) = f(x)\cdot E(x)$ ונשים לב שלכל α_i מתקיים ש $N(\alpha_i) = f(\alpha_i)\cdot E(\alpha_i) = y_p \cdot E(\alpha_i)$. כלומר שמידיעת $\bar{y},E$ ניתן לחשב את $N(x)$ בn נקודות. $\deg(N) < k+t < n$ ולכן ע"י אינטרפולציה נוכל לחשב את $N(x)$. כעת, בהינתן $N(x) = f(x)\cdot E(x)$ ו$E(x)$ נחשב את f ע"י $\frac{N(x)}{E(x)}$.
מה הבעיה באלגוריתם? אין לנו את E ובשביל האלגוריתם צריך את E ואת y... ובשביל למצוא את f צריך את Nו E... ננסה למצוא גרסאות קרובות של N,E שיביאו אותנו ל f.

אלגוריתם:

1. מצא פולינומים $E'(x),N'(x)$ כך ש־
   (א) $\deg(E') \leq t$ (כמו E)
   (ב) $\deg(N') \leq k+t$ (כמו N)
   (ג) לכל $1 \leq i \leq n$, מתקיים $N(\alpha_i) = y_i \cdot E(\alpha_i)$ (כמו הדרישה בגרסה השניה רק בלי לדבר על ההתאפסויות של E).
2. חשב את המנה $N'(x)/E'(x)$

טענה 1.7 ניתן לממש את $1,2$ ביעילות והתוצאה (המפתיעה) היא $f(x)$ אם מתקיים ש $\mathrm{dist}(\bar{y},\mathrm{Enc(f)}) < \frac{n-k+1}{2}$.

הוכחה: נכתוב $E'(x)=\sum_{i=0}^{t} e_i x^i$, $N'(x)=\sum_{i=0}^{k+t-1} c_i x^i$. נחשוב על $e_0,\ldots,e_t$, $c_0,\ldots,c_{k+t-i}$ כעל נעלמים אותם אנחנו רוצים לחשב. לפי 1ג, צקיד להתקיים עבור

$$\sum_{i=0}^{k+t-1} c_i \left(\alpha_j\right)^i = \left(\sum_{i=0}^{t} e_i \left(\alpha_j\right)^i\right) y_j$$

זו משוואה לינארית בנעלמים של N', E', כלומר $\{n_i\} \cup \{e_i\}$. ולכן כדי לממש את שלב 1, נפתור את המשוואות הנ"ל, יש n כאלה (α_i־ים) ב$t+(t+k+1)=2t+k+1$ נעלמים ומקודם הראנו שמערכת זו פתירה, כלומר שקיים לה פתרון לא טריוויאלי. ולכן ניתן לממש בזמן $\mathcal{O}(n^3)$ את שלב 1 ולמצוא את E', N'. ■

טענה 1.8 יהי E ה־ELP, $N = f \cdot E$. אזי לכל פתרון לא טריוויאלי מתקיין ש:

$$f(x) = \frac{N(x)}{E(x)} = \frac{N'(x)}{E'(x)}$$

הוכחה: נראה ש

$$\overbrace{N(x) \cdot E'(x)}^{\deg<k+2t} = \overbrace{N'(x) \cdot E(x)}^{\deg<k+2t}$$

נראה כי לכל i מתקיים $N(\alpha_i) \cdot E'(\alpha_i) = N'(\alpha_i) \cdot E(\alpha_i)$.

- אם $f(\alpha_i) = y_i$ (אין טעות) ואז

$$N(\alpha_i) = f(\alpha_i) \cdot E(\alpha_i) = y_i \cdot E(\alpha_i) \qquad N'(\alpha_i) = y_i \cdot E'(\alpha_i) \Rightarrow$$

$$N(\alpha_i)E'(\alpha_i) = N'(\alpha_i)E(\alpha_i)$$

- אם $f(\alpha_i) \neq y_i$ אז $E(\alpha_i) = 0 \Rightarrow N(\alpha_i) = 0$ ואז כל הצדדים 0 וכולם שמחים, כלומר ש

$$0 = N(\alpha_i)E'(\alpha_i) = N'(\alpha_i)E(\alpha_i) = 0$$

ובפרט, לפחות במקרה של רק טעויות, בלי מחיקות מתקיים

$$f(x) = \frac{N(x)}{E(x)} = \frac{N'(x)}{E'(x)}$$

כנדרש ■

כעת (אחרי שמצאנו את f) נניח שיש m מחיקות וt טעויות. $2t+m < n-k+1$. נצמצם את הקוד ל$n-m$ קואורדינטות ללא מחיקות. נקבל קוד RS (על פחות נקודות) ־ $[n-m, k, n-m-k+1]$ ובו ניתן לתקן t' טעויות כל עוד $2t' < n-m-k+1$.

1.3 אבסטרקציה של האלגוריתם לתיקון טעויות בקוד RS

יהי C קוד מעל $\mathbb{F}_2^n$. זוג A, B נקרא Error locating pair עבור C אם מתקיימים התכונות הבאות:

סימון: בהינתן $v, u \in \mathbb{F}_q^n$, נגדיר $v * u \in \mathbb{F}_q^n$ ע"י $(v*u)_i = v_i \cdot u_i$

1. $A * C \subseteq B$ כלומר $\forall v \in A, u \in C. v * u \in B$.
2. $dist(A) > t$.
3. $dist(B) > t$

4. $dist(C) > n - dist$ (כי $n - k + 1 < t$)

דוגמה:

$$C = \mathrm{RS}\left[n, k, n-k+1\right] \; A = \mathrm{RS}\left[n, t+1, n-t\right] \; B = \mathrm{RS}\left[n, k+t, n-k-t+1\right]$$

$$\mathrm{Enc}\left(E(x)\right) \in A \qquad \mathrm{Enc}\left(N(x)\right) \in B \qquad \mathrm{Enc}\left(f(x)\right) \in C$$

אלגוריתם לתיקון t טעויות ב־ C: נתון: קוד C, זוג A, B שמקיימים את התכונות ומילה "מורעשת" $\bar{y}$.

1. מצא $a \in A$, $b \in B$, $(a,b) \neq 0$ כך ש $a * y = b$
2. נחשב $z \in (\mathbb{F}_q \cup \{?\})^n$ (שאומרת איפה יש טעויות) באופן הבא:

$$z_i = \begin{cases} ? & a_i = 0 \\ y_i & else \end{cases}$$

3. תקן את z בקוד C (תיקון ממחיקות).

טענה 1.9 אם יש מילת קוד $c \in C$ כך ש $dist\left(\bar{y}, c\right) < t$ אזי האלגוריתם מוצא את c.

טענה 1.10 יש פתרון a, b למערכת.

הוכחה: יהי $a \in A \neq 0$ המקיים $a_i \neq 0$ אם $y_i \neq c_i$. אלה tאילוצים על a. ומכיוון ש $\dim(A) > t$ יש a כנ"ל. נגדיר $b = a * c$ונשים לב ש $b = a * y$. מכיוון שהאילוץ $a * y = b$ היא מערכת משוואות לינארית הומוגנית ויש פתרון לא טריוויאלי

$$\begin{pmatrix} y_1 & & \\ & \ddots & \\ & & y_n \end{pmatrix} \overbrace{G_A \begin{pmatrix} x_1 \\ \vdots \\ x \end{pmatrix}}^{\in A} = \overbrace{G_B \begin{pmatrix} \gamma_1 \\ \vdots \\ \end{pmatrix}}^{\in B}$$

ולכן שלב 1 באלגוריתם טובה, נסתכל על שלב ג. ■

טענה 1.11 אם (a', b') פתרון אזי $a' * c = b'$.

הוכחה: אנחנו יודעים ש $a' * y = b'$. נביט ב $a' * c = b'' \in B$

$$\mathrm{dist}\left(b', b''\right) \leq \mathrm{dist}\left(y, c\right) \leq t \overset{\mathrm{dist}(B) > t}{\Rightarrow} b' = b''$$

■

טענה 1.12 האלגוריתם עובד.

הוכחה: מהטענה האחרונה נובע שאם $z_i \neq ?$ אזי $z_i = c_i$. כל מה שנותר לעשות הוא לחסום את כמות המחיקות. אם כמות המחיקות $> \mathrm{dist}(C)$אז סיימנו. בכמה קואורדינטות יכול להתקיים $a' = 0$?
$a' = 0$ בלכל היותר $n - \mathrm{dist}(A)$ קואודינטות־ אחרת הוא יהיה קרוב מדי למילת ה־0 ואז נקבל שהוא שווה ל 0 בסתירה.
התנאי הרביעי $\mathrm{dist}(C) > n - \mathrm{dist}(A)$. ■