

סיבוכיות

© ארזים

14 במרץ 2017

1 מבוא

סיבוכיות - התחום החוקר את הקושי של בעיות חישוביות. נרצה להבין מה הם המשאבים הדרושים לשם פתרון בעיות חישוביות.

משאבים: זמן ריצה, זיכרון, מקביליות, אקראיות, תקשורת וכדומה. נרצה גם להבין את יחסי הגומלין בין המשאבים השונים.

דוגמאות לשאלות שננסה להבין:

1. כמה מהר ניתן לפתור מערכת משוואות לינאריות (n משוואות עם n נעלמים)?
 2. כפל לעומת חיבור - עד כמה חיבור מהיר יותר?
 3. האם אקראיות עוזרת להאיץ חישובים בצורה משמעותית?
 4. האם יש בעיות חישוביות שלצורך פתירתן דרוש הרבה זיכרון?
 5. האם מחשבים יכולים להחליף מתמטיקאים?
 6. איך מוודאים שלמישהו יש כח חישוב גדול משלנו?
 7. האם קושי חישובי יכול להיות מועיל?
- אנחנו נדבר בתחילה על בעיקר על שאלות 1,2,7.

1.1 פתרון משוואות

נתונה מטריצה ריבועית A ממימד n , ווקטור $b \in \mathbb{R}^n$. רוצים למצוא ווקטור $x \in \mathbb{R}^n$ המקיים

$$Ax = b$$

נניח לשם הפשטות כי A הפיכה. לכן הפתרון הוא כמובן

$$x = A^{-1}b$$

ולכן אנו רואים שדי לחשב את A^{-1} . האלגוריתם הבסיסי לחישוב הופכי למטריצה (על ידי דירוג) לוקחת זמן $\Omega(n^3)$. האם אפשר טוב יותר? בעיות קשורות מאלגברה לינארית:

- פתרון מערכת משוואות לינאריות.

- חישוב מטריצה הופכית.

- חישוב דטרמיננטה.

- כפל מטריצות.

כל הבעיות הללו שקולות חישובית. נראה את הרדוקציה מחישוב הופכי לכפל מטריצות. נתונה מטריצה ריבועית ממסית A ממימד n (נניח כי n חזקת 2). נניח כי A מטריצה סימטרית מוגדרת חיובית. מטריצה סימטרית היא מטריצה בעלת המבנה:

$$A = \begin{pmatrix} X & Y^t \\ Y & Z \end{pmatrix}$$

כאשר X, Y, Z מטריצות ריבועיות ממימד $\frac{n}{2}$ (X, Z סימטריות בעצמן).

הגדרה 1.1 A נקראת מוגדרת חיובית אם היא סימטרית ולכל וקטור $v \neq 0$,

$$v^t A v > 0$$

עובדה עבור A כזו, המטריצה X בתיאור מעלה גם כן מוגדרת חיובית.

עובדה אם B מטריצה הפיכה, אזי BB^t מוגדרת חיובית, וכך גם $B^t B$.

אם כן, נניח כי

$$A = \begin{pmatrix} X & Y^t \\ Y & Z \end{pmatrix}$$

מטריצה מוגדרת חיובית. נגדיר מטריצה חדשה:

$$S = Z - YX^{-1}Y^t$$

תרגיל S הפיכה.

טוענים כי

$$\begin{pmatrix} X & Y^t \\ Y & Z \end{pmatrix}^{-1} = \begin{pmatrix} X^{-1} + X^{-1}Y^tS^{-1}YX^{-1} & -X^{-1}Y^tS^{-1} \\ -S^{-1}YX^{-1} & S^{-1} \end{pmatrix}$$

נבדוק:

$$\begin{pmatrix} X & Y^t \\ Y & Z \end{pmatrix} \begin{pmatrix} X^{-1} + X^{-1}Y^tS^{-1}YX^{-1} & -X^{-1}Y^tS^{-1} \\ -S^{-1}YX^{-1} & S^{-1} \end{pmatrix} = \begin{pmatrix} I_{\frac{n}{2}} & 0 \\ 0 & I_{\frac{n}{2}} \end{pmatrix}$$

קואורדינטה שמאלית תחתונה:

$$YX^{-1} + YX^{-1}Y^tS^{-1}YX^{-1} - ZS^{-1}YX^{-1} = YX^{-1} - (Z - YX^{-1}Y^t)S^{-1}YX^{-1} = \\ = YX^{-1} - SS^{-1}YX^{-1} = 0$$

שאר החישובים דומים (או קלים). מכאן נקבל נוסחת רקורסיה - כדי לחשב את A^{-1} צריך לחשב את S^{-1}, X^{-1} (ממימד $\frac{n}{2}$) ועוד כל מיני כפלי מטריצות:

$$S^{-1}YX^{-1}, X^{-1}Y^tS^{-1}YX^{-1}$$

נסמן את הזמן הדרוש לכפול מטריצות מסדר n בתור $M(n)$, ובתור $I(n)$ את הזמן הדרוש לחישוב הופכי של מטריצה ממימד n . אזי

$$I(n) \leq 2I\left(\frac{n}{2}\right) + 4M\left(\frac{n}{2}\right) + O(n^2)$$

ארבעת הכפלים הם $(YX^{-1})^t(S^{-1}YX^{-1}), S^{-1}(YX^{-1}), (YX^{-1})Y^t, YX^{-1}$. אנחנו מכניסים את $O(n^2)$ בשביל כל החיבורים והפעולות שנוותרו, שכולן לוקחות לכל היותר n^2 זמן, שכן אנחנו לא יודעים לכפול מטריצות בזמן n^2 .

תרגיל $I(n) = O(M(n))$ (כי $M(n) \gg n^2$).

הערה 1.2 נניח שנתון רק כי A הפיכה. כדי לחשב את A^{-1} נוכל לחשב את הדבר הבא:

$$A^{-1} = (A^t A)^{-1} \cdot A^t$$

והמטריצה $A^t A$ היא מוגדרת חיובית.

נעבור לדבר על כפל מטריצות. יהיו A, B מטריצות ריבועיות מסדר n . נסמן את עמודות B בתור $v_1, \dots, v_n$. אזי

$$AB = A(v_1, \dots, v_n) = (Av_1, \dots, Av_n)$$

למעשה, כפל מטריצות שקול לחישוב n הפעלות של הפונקציה $f(x) = Ax$. ברור כי זמן הריצה הוא לכל היותר n כפול זמן הריצה של חישוב f .

שאלה האם לחשב n עותקים של פונקציה "קשה פי n " מאשר לחשב עותק בודד?

מסתבר שיש אלגוריתמים "מהירים" לחישוב כפל מטריצות.

1.1.1 אלגוריתם שטראסן (Strassen) לכפל מטריצות

שוב, נבצע רדוקציה לכפל מטריצות ממימד קטן פי 2.

$$AB = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}$$

האלגוריתם הנאיבי דורש 8 פעולות כפל ועוד 4 פעולות חיבור. שטראסן מצא דרך לעשות אותו דבר עם 7 פעולות כפל והרבה (מספר קבוע עדיין) פעולות חיבור.

$$\begin{aligned}c_1 &= (A_{11} + A_{22})(B_{11} + B_{22}) \\c_2 &= (A_{21} + A_{22})B_{11} \\c_3 &= A_{11}(B_{12} - B_{22}) \\c_4 &= A_{22}(B_{21} - B_{11}) \\c_5 &= (A_{11} + A_{12})B_{22} \\c_6 &= (A_{21} - A_{11})(B_{11} + B_{12}) \\c_7 &= (A_{12} - A_{22})(B_{21} + B_{22})\end{aligned}$$

כעת, הפלט הוא

$$\begin{pmatrix} c_1 + c_4 - c_5 + c_7 & c_3 + c_5 \\ c_2 + c_4 & c_1 + c_3 - c_2 + c_6 \end{pmatrix}$$

יש כאן 7 פעולות כפל ועוד 18 פעולות חיבור.

$$\begin{aligned}M(n) &\leq 7M\left(\frac{n}{2}\right) + 18n^2 \\M(n) &= O(n^{\log_2 7}) \approx O(n^{2.81\dots})\end{aligned}$$

האלגוריתם הזה הפתיע מאוד את הקהילה המתמטית, שהאמינה שלא ניתן לבצע כפל מטריצות מהר יותר מאשר n^3 . האלגוריתם הטוב ביותר שידוע היום הוא בזמן

$$M(n) = O(n^{2.373\dots})$$

ומאמינים שניתן בזמן $n^{2+o(1)}$.

1.2 כפלים אחרים - מספרים, פולינומים

פולינום במשתנה אחד מעל שדה $\mathbb{F}$ (אצלנו למשל $\mathbb{R}, \mathbb{C}, \mathbb{F}_2$ וכדומה) הוא ביטוי מהצורה

$$f(x) = \sum_{i=0}^n a_i x^i$$

כאשר $a_0, \dots, a_n \in \mathbb{F}$ מסמנים

$$\deg(f) = \max\{i \mid a_i \neq 0\}$$

בעיית כפל פולינומים: בהינתן $f(x), g(x)$ עם $\deg(f), \deg(g) < n$ רוצים לחשב את $h(x) = f(x)g(x)$. נכתוב

$$f(x) = \sum_{i=0}^{n-1} a_i x^i$$

$$g(x) = \sum_{i=0}^{n-1} b_i x^i$$

ואז נקבל

$$h(x) = \sum_{i=0}^{2n-2} \left(\sum_{j=0}^i a_j b_{i-j} \right) x^i$$

האלגוריתם הטבעי דורש n^2 פעולות כפל. לעומת זאת, חישוב $f + g$ דורש $O(n)$ זמן. כפל מספרים: נתון מספר בייצוג בינארי/עשרוני/כל בסיס אחר:

$$a = a_{n-1} \dots a_2 a_1 a_0$$

$$b = b_{n-1} \dots b_2 b_1 b_0$$

ורוצים לחשב את $a \cdot b$. גם כאן, כפל ייקח $O(n^2)$ בשיטה הנאיבית, בעוד חיבור דורש $O(n)$ זמן. שתי הבעיות דומות וקשורות, והאלגוריתמים שנראה קשורים גם כן.

1.2.1 אלגוריתם קרצובה (Karatsuba)

נכתוב

$$f = f_0 + x^{\frac{n}{2}} f_1$$

$$f_0 = \sum_{i=0}^{\frac{n}{2}-1} a_i x^i$$

$$f_1 = \sum_{i=\frac{n}{2}}^{n-1} a_i x^{i-\frac{n}{2}}$$

באותו אופן נכתוב גם

$$g = g_0 + x^{\frac{n}{2}} g_1$$

כעת נחשב את

$$a = (f_0 + f_1)(g_0 + g_1)$$

$$b = f_0 g_0$$

$$c = f_1 g_1$$

נשים לב כי

$$\begin{aligned} h = fg &= \left(f_0 + x^{\frac{n}{2}} f_1\right) \left(g_0 + x^{\frac{n}{2}} g_1\right) = f_0 g_0 + x^{\frac{n}{2}} (f_1 g_0 + f_0 g_1) + x^n f_1 g_1 = \\ &= b + x^{\frac{n}{2}} (a - b - c) + x^n c \end{aligned}$$

ביצענו כאן 3 פעולות כפל ועוד 6 פעולות חיבור. לכן

$$T(n) \leq 3T\left(\frac{n}{2}\right) + O(n)$$

$$T(n) = O\left(n^{\log_2 3}\right) \approx O\left(n^{1.58\dots}\right)$$