ΠΛΗΡΟΦΟΡΙΚΗ Ι

Βασικά Στοιχεία Λογισμού Πινάκων και Θεμελιώδεις Αλγόριθμοι

Μέρος ΙΙ : Απαλοιφή *Gauss*

Μαριλένα Μητρούλη

Οκτώβριος 2023

1 Βασικές Πράξεις Γραμμικής Άλγεβρας

1.1 Εσωτερικό Γινόμενο (inner product: ip)

Εάν $x, y \in R^n$,το εσωτερικό τους γινόμενο (inner product) ορίζεται ως εξής $x^t y = \sum_{i=1}^{n} x_i y_i$

Υπολογισμός εσωτερικού γινομένου

Εστω $x, y \in \Re^n$. Να δοθεί αλγόριθμος που υπολογίζει το εσωτερικό τους γινόμενο.

$$z = x^t \cdot y = \sum_{i=1}^{n} x_i \cdot y_i$$

Αλγόριθμος ip:

$z = 0$
$for\ i = 1, 2, \ldots, n$
$\quad z = z + x_i \cdot y_i \rightarrow 1$ flop

Για κάθε επανάληψη i απαιτείται 1 $flop$, έτσι ο αλγόριθμος για την εκτέλεσή του χρειάζεται n $flops$ (γραμμική πολυπλοκότητα).

Αποτελεσματικότητα:	$O(n)$	$flops$
Μνήμη :	$2n+1$	θέσεις

Διανυσματική Εκτέλεση (vectorized)

$ip = x.'y$ (εκτέλεση πράξεων κατά συντεταγμένες)

1.2 Εξωτερικό Γινόμενο (Outer Product)

Το εξωτερικό γινόμενο δύο διανυσμάτων x, y ορίζεται ως εξής.

$$x, y \in R^{n \times n} \rightarrow xy^t = \begin{bmatrix} x_1y_1 & x_1y_2 & \cdots & x_1y_n \\ x_2y_1 & x_2y_2 & \cdots & x_2y_n \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ x_ny_1 & x_ny_2 & \cdots & x_ny_n \end{bmatrix} = \mathrm{G} = [G_{ij}] \in R^{n \times n}$$

Υπολογισμός εξωτερικού γινομένου

Εστω $x, y \in \Re^n$. Να δοθεί αλγόριθμος που υπολογίζει το εξωτερικό τους γινόμενο.

Αλγόριθμος *op*:

for $i = 1, 2, \ldots, n$
  for $j = 1, 2, \ldots, n$
    $G(_{ij}) = x(i) * y(i) \rightarrow 1$ flop

Για κάθε επανάληψη απαιτείται 1 $flop$, έτσι ο αλγόριθμος για την εκτέλεσή του χρειάζεται n^2 $flops$ (τετραγωνική πολυπλοκότητα).

Αποτελεσματικότητα:	$O(n^2)$	$flops$
Μνήμη :	$n^2 + 2n$	θέσεις

1.3 Γινόμενο Πίνακα-Διάνυσμα (Matrix Vector Product: mvp)

Εστω A ένας $m \times n$ και $x \in \Re^n$ Το γινόμενο του πίνακα A με το διάνυσμα x ορίζεατι ως εξής. $y = Ax \in R^{n\times 1} \Rightarrow y_i = \sum_{j=1}^{m} a_{i,j} x_j$

Υπολογισμός γινομένου διανύσματος με πίνακα (matrix vector product, mvp).
Εστω A ένας $m \times n$ και $x \in \Re^n$. Να δοθεί αλγόριθμος που υπολογίζει το γινόμενο $y = A \cdot x$.

Αλγόριθμος **mvp**:

for $i = 1, 2, \ldots, m$
  for $j = 1, 2, \ldots, p$
    $y_i = \sum_{k=1}^{n} a_{ik} x_k \rightarrow 1$ flop

Για τον υπολογισμό κάθε y_i απαιτούνται n flops. Αφού το i τρέχει από 1 έως m, ο συνολικός αριθμός πολλαπλασιασμών που απαιτούνται είναι mn. Εάν $m = n$ ο αλγόριθμος για την εκτέλεσή του χρειάζεται n^2 flops (τετραγωνική πολυπλοκότητα).

Αποτελεσματικότητα:	$O(mn)$	flops
Μνήμη :	$mn + 2n$	θέσεις

Παρατήρηση: Οι υπολογισμοί εσωτερικών γινομένων και γινομένων διανυσμάτων πινάκων αποτελούν τον πυρήνα σχεδόν κάθε αλγόριθμου Αριθμητικής

Γραμμικής Αλγεβρας. Πολλές φορές λοιπόν εκτιμάται η αποτελεσματικότητα του αλγορίθμου από το συνολικό αριθμό των απαιτούμενων ip's και mvp's.

Διανυσματική Εκτέλεση(vectorized) $c = A * x$

1.4 Γινόμενο πινάκων (Matrix Product)

Εστω $A = [a_{ij}] \in R^{m \times n}$, $B = [b_{ij}] \in R^{n \times p}$. Το γινόμενό τους ορίζεται ως εξής. $C = [c_{ij}] = AB \in R^{m \times p}$ με $c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj},\ i = 1, \ldots, m,\ j = 1, \ldots, p$

Υπολογισμός γινομένου πινάκων
(matrix product).
Εστω A ένας $m \times n$ πίνακας και B ένας $n \times p$ πίνακας. Ο ακόλουθος αλγόριθμος υπολογίζει το γινόμενο $C = A \cdot B$.

Αλγόριθμος **Matprod**:

$C = zeros(m, p)$ (αρχικοποίηση θέσεων μνήμης)

for $i = 1, 2, \ldots, m$
  for $j = 1, 2, \ldots, p$
    $c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj} \rightarrow$ n flops

Για τον υπολογισμό κάθε c_{ij} απαιτούνται n flops. Αφού το j τρέχει από 1 έως p και το i από 1 έως m, ο συνολικός αριθμός πολλαπλασιασμών που απαιτούνται είναι mnp.

Αποτελεσματικότητα:	$O(mnp)$	flops
Μνήμη :	$mn + np + mp$	θέσεις

Εάν έχουμε δύο τετραγωνικούς πίνακες τάξης n τότε απαιτούνται συνολικά n^3 flops (κυβική πολυπλοκότητα). Παρατηρούμε λοιπόν ότι το γινόμενο πινάκων απαιτεί τη μέγιστη επιτρεπτή πολυπλοκότητα, επομένως θεωρείται πολύ δαπανηρή πράξη και θα πρέπει να χρησιμοποιείται με φειδώ.

Διανυσματική Εκτέλεση(vectorized)
$C = A * B$

1.5 Υπολογισμός Ορίζουσας Πίνακα

Έστω πίνακας $A \in R^{25*25}$ του οποίου θέλουμε να υπολογίσουμε την ορίζουσα. Από το ανάπτυγμα Laplace ώς προς την 1η γραμμή προκύπτει:
$det(A) = (-1)^{1+1}det(A_{11}) + (-1)^{1+2}det(A_{12}) + \ldots .(-1)^{1+25}det(A_{125})(1)$
Ο τύπος 1 χρειάζεται 25 πολλαπλασιασμούς και 25 υπολογισμούς από ορίζουσες διάστασης 24 όπου καθεμία χρειάζεται 24 πολλαπλασιασμούς και 24 υπολογισμούς από ορίζουσες διάστασης 23 και ούτω καθεξής.
Επομένως ο υπολογισμός της det(A). απαιτεί περισσότερους από 25 * 24 * ...1=25! $= = 1.55 * 1025$ πράξεις.
Σε έναν υπολογιστή που εκτελεί 10^{12} πολλαπλασιασμούς ανά δευτερόλεπτο δηλαδή $3.15 * 1019$ το χρόνο, ο υπολογισμός της det(A) θα απαιτούσε 49.000 χρόνια!!!
Πρέπει λοιπόν να αναζητηθεί άλλη μέθοδος υπολογισμού της ορίζουσας. Έτσι χρησιμοποιούμε γραμμοπράξεις που οδηγούν σε ισοδύναμο πίνακα που έχει την ίδια ορίζουσα

1.6 Ορισμός: ισοδύναμοι πίνακες (equivalent matrices)

Δύο πίνακες $A, B \in R^{n\times n}$ λέγονται ισοδύναμοι (equivalent) αν υπάρχουν αντιστρέψιμοι πίνακες $P, Q \in R^{n\times n} : B = P * A * Q$

1.7 Ορισμός : Γραμμοπράξεις (row operations)

I. Εναλλαγή δύο γραμμών του πίνακα.

II. Πολλαπλασιασμός μιας γραμμής με βαθμωτό.

III. Πρόσθεση σε μία γραμμή του πολλαπλασιασμού μιας άλλης.

Εάν ένας πίνακας B προκύπτει από έναν άλλο A στον οποίο έχει εφαρμοστεί ένας πεπερασμένος αριθμός γραμμοπράξεων έχει την ίδια ορίζουσα με τον A. δλδ εάν $A \xrightarrow[\gamma\rho\alpha\mu\mu o\pi\rho\xi\epsilon\iota\varsigma]{} B \Longrightarrow det(A) = det(B)$

<u>Παράδειγμα 1</u>

$$A = \begin{array}{r} r1 \\ r2 \\ r3 \end{array} \begin{bmatrix} -1 & 4 & 1 \\ 3 & 1 & 1 \\ 2 & 6 & 1 \end{bmatrix} \xrightarrow[r_3+2r_1]{r_2+3r_1} \begin{bmatrix} -1 & 4 & 1 \\ 0 & 13 & 4 \\ 0 & 14 & 3 \end{bmatrix} \xrightarrow{r_2-r_3} \begin{bmatrix} -1 & 4 & 1 \\ 0 & -1 & 1 \\ 0 & 14 & 3 \end{bmatrix} \xrightarrow{r_3+14r_2} \begin{bmatrix} -1 & 4 & 1 \\ 0 & -1 & 1 \\ 0 & 0 & 17 \end{bmatrix}$$

$det(A) = det(B) = (-1)(-1)(17) = 17$

1.8 Εκτέλεση γραμμοπράξεων με πίνακες

Οι παραπάνω γραμμοπράξεις μπορούν να εκτελεσθούν με γινόμενα πινάκων. Πολλαπλασιάζουμε από αριστερά με τον μοναδιαίο πίνακα I_n κατάλληλα μετασχηματισμένο.

I. Αλλαγή γραμμών.

Πολλαπλασιασμός πχ με $\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ αντιμεταθέτει την 1η με την 2η γραμμή

II. Πολλαπλασιασμός γραμμής

Πολλαπλασιασμός πχ με $\begin{bmatrix} 1 & 0 & 0 \\ 0 & c & 0 \\ 0 & 0 & 1 \end{bmatrix}$ Πολλαπλασιάζει με c τα στοιχεία της 2ης γραμμής.

III. Πρόσθεση σε γραμμή άλλης πολλαπλασιασμένης με αριθμό

Πολλαπλασιασμός πχ με $\begin{bmatrix} 1 & c & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ προσθέτει c φορές στην 1η γραμμή τη 2η

Έστω A= $\begin{bmatrix} 1 & 2 & 3 \\ 2 & 1 & 5 \\ 3 & 2 & 4 \end{bmatrix}$

I. $\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \\ 2 & 1 & 5 \\ 3 & 1 & 4 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 5 \\ 1 & 2 & 3 \\ 3 & 1 & 4 \end{bmatrix}$

II. $\begin{bmatrix} 1 & 0 & 0 \\ 1 & 3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \\ 2 & 1 & 5 \\ 3 & 1 & 4 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 \\ 6 & 3 & 15 \\ 3 & 1 & 4 \end{bmatrix}$

III. $\begin{bmatrix}1 & 3 & 0\\0 & 1 & 0\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 2 & 3\\2 & 1 & 5\\3 & 1 & 4\end{bmatrix}=\begin{bmatrix}5 & 4 & 13\\2 & 1 & 5\\3 & 1 & 4\end{bmatrix}$

Στο παράδειγμα 1 μπορούμε να δούμε την εκτέλεση των γραμμοπράξεων με πολλαπλασιασμούς πινάκων.

A $=\begin{bmatrix}-1 & 4 & 1\\3 & 1 & 1\\2 & 6 & 1\end{bmatrix}$ να προσδιορισθεί ισοδύναμος B που να είναι άνω τριγωνικός

$$\underset{M_1}{\begin{bmatrix}1 & 0 & 0\\3 & 1 & 0\\2 & 0 & 1\end{bmatrix}} \times \underset{A^{(0)}}{\begin{bmatrix}-1 & 4 & 1\\3 & 1 & 0\\2 & 6 & 1\end{bmatrix}} = \underset{A^{(1)}}{\begin{bmatrix}-1 & 4 & 1\\0 & -1 & 1\\0 & 14 & 3\end{bmatrix}}$$

$$\underset{M_2}{\begin{bmatrix}1 & 0 & 0\\0 & 1 & -1\\0 & 0 & 1\end{bmatrix}} \times \underset{A^{(1)}}{\begin{bmatrix}-1 & 4 & 1\\0 & 13 & 4\\0 & 14 & 3\end{bmatrix}} = \underset{A^{(2)}}{\begin{bmatrix}-1 & 4 & 1\\0 & 13 & 4\\0 & 14 & 3\end{bmatrix}}$$

$$\underset{M_3}{\begin{bmatrix}1 & 0 & 0\\0 & 1 & 0\\0 & 14 & 1\end{bmatrix}} \times \underset{A^{(2)}}{\begin{bmatrix}-1 & 4 & 1\\0 & -1 & 1\\0 & 14 & 3\end{bmatrix}} = \underset{A^{(3)}}{\begin{bmatrix}-1 & 4 & 1\\0 & -1 & 1\\0 & 0 & 17\end{bmatrix}}$$

Τελικά

$B = A^{(3)} = M_3A^{(2))} = M_3M_2A^{(1)} = M_3M_2M_1A^{(0)} = M * A$, όπου M= $M_3M_2M_1$

$det(M) = det(M_3M_2M_1) = det(M_3) \cdot det(M_2) \cdot det(M_1) = 1 * 1 * 1 = 1$

$det(B) = det(MA) = det(M) \cdot det(A) = det(A) \Rightarrow det(A) = 17$

1.9 Τεχνική υπολογισμού ορίζουσας

Ξεκινώντας από έναν πυκνό πίνακα A προσπαθούμε με γραμμοπράξεις να τον μετατρέψουμε σε άνω τριγωνικό οπότε η ορίζουσα θα ισούται με το γινόμενο των διαγώνιων στοιχείων του. Στη συνέχεια θα περιγράψουμε την Αριθμητική μέθοδο με την οποία ένας πίνακας μπορεί να μετατραπεί σε άνω τριγωνικό και κατά συνέπεια μπορούμε να υπολογίσουμε την ορίζουσά του.

1.10 Απαλοιφή του Gauss (Gaussian Elimination)

Παράδειγμα 2 : Να μετατραπεί ο παρακάτω πίνακας A σε άνω τριγωνικό με τη μέθοδο απαλοιφής του Gauss

$$A = \begin{array}{c} r_1 \\ r_2 \\ r_3 \\ r_4 \end{array} \begin{bmatrix} 1 & 2 & 1 & 4 \\ 2 & 0 & 4 & 3 \\ 4 & 2 & 2 & 1 \\ -3 & 1 & 3 & 2 \end{bmatrix}$$

Ξεκινάμε το μηδενισμό κάθε στήλης κάτω από τη διαγώνιο ορίζοντας για καθεμία κατάλληλο πολλαπλασιαστή m_{ij}

Οδηγό στοιχείο 1ης στήλης

$$\begin{array}{l} m_{21} = 2/1 = 2 \\ m_{31} = 4/1 = 4 \\ m_{41} = -3/1 = -3 \end{array} \quad \begin{bmatrix} 1 & 2 & 1 & 4 \\ 2 & 0 & 4 & 3 \\ 4 & 2 & 2 & 1 \\ -3 & 1 & 3 & 2 \end{bmatrix} \xrightarrow{\substack{r_2-m_{21}r_1 \\ r_3-m_{31}r_1}\; r_4-m_41r_1}$$

Οδηγό στοιχείο 2ης στήλης

$$\begin{bmatrix} 1 & 2 & 1 & 4 \\ 0 & -4 & 2 & -5 \\ 0 & -6 & -2 & -15 \\ 0 & 7 & 6 & 14 \end{bmatrix}$$

Οδηγό στοιχείο 3ης στήλης

$$\begin{array}{l} m_{32} = -6/-4 \\ m_{31} = 7/-4 \end{array} \xrightarrow{\substack{r_3-m_{32}r_2 \\ r_3-m_{31}r_1}} \begin{bmatrix} 1 & 2 & 1 & 4 \\ 0 & -4 & 2 & -5 \\ 0 & 0 & -5 & -7.5 \\ 0 & 0 & 9.5 & 5.25 \end{bmatrix} \quad m_{43} = 9.5/-5 \quad \xrightarrow{r_4-m_{43}r_3}$$

$$\begin{bmatrix} 1 & 2 & 1 & 4 \\ 0 & -4 & 2 & -5 \\ 0 & 0 & -5 & -7.5 \\ 0 & 0 & 0 & -9 \end{bmatrix} \Rightarrow$$

Παρατηρούμε ότι η προηγούμενη διαδικασία χαρακτηρίζεται από την επανάληψη μιας συγκεκριμένης μεθοδολογίας. Για κάθε στήλη μηδενίζουμε τα στοιχεία κάτω από τη διαγώνιο ορίζοντας σαν πολλαπλασιαστές τα πηλίκα των τιμών των προς μηδενισμό στοιχείων δια του οδηγού. Παρατηρούμε ότι αυτή η διαδικασία μπορεί να συνεχιστεί για κάθε στήλε του πίνακα και επομένως οδηγεί σε αλγόριθμο. Στη συνέχεια παρουσιάζουμε τον αλγόριθμο της απαλοιφής του Gauss, τον αρχαιότερο και πιο αληθοφανή αλγόριθμο της Αριθμητικής Γραμμικής Αλγεβρας.

1.11 Αλγόριθμος Απαλοιφής του Gauss

INPUT : Διάβασε τον $A \in R^{n\times n}$
% Θα μετατραπεί σε άνω τριγωνική μορφή

For k=1,2,...,n
B1 : δημιουργία πολλαπλασιαστών
$a_{ik} = m_{ik} = \dfrac{a_{ik}}{a_{kk}}$, $i = k+1,..,n \to n-k flops$

B2 : ενημέρωση εισόδων
$a_{ij} = a_{ij} - m_{ik}a_{kj}, \begin{Bmatrix} i = k+1,..,n \\ j = k+1,..,n \end{Bmatrix} \to (n-k)^2 flops$

OUTPUT :Εκτύπωσε τον $A \in R^{n\times n}$

A= $\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ m_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ m_{n1} & m_{n2} & \cdots & a_{nn} \end{bmatrix}$ Εδώ αποθηκεύονται τα καινούργια στοιχεία

Οικονομική Αποθήκευση
Στο κάτω τριγωνικό μέρος, δεν αποθηκεύουμε τα μηδενικά αλλά κρατάμε τους πολλαπλασιαστές.

Πολυπλοκότητα
$\sum_{k=1}^{n-1}(n-k)^2 + \ \sum_{k=1}^{n-1}(n-k) \approx 0(\frac{n^3}{3})$

Αλγόριθμος

```
for k=1:n-1
  for i=k+ 1:n
    A(i,k)=A(i,k)/A(k,k)
  end
  for i=k+ 1:n
    for j=k+ 1:n
      A(i,j)=A(i,j)-A(i,k)*A(k,j)
  end
end
```

Διανυσματική Εκτέλεση(vectorized)

A(k+1:n,k+1:n)=A(k+1:n,k+1:n)-A(k+1:n,k)*A(k,k+1:n)

Σημείωση : Επειδή η άνω τριγωνοποίηση ενός πίνακα είναι από τις βασικότερες αλλά και πολυπλοκότερες διαδικασίες ο αριθμός των απαιτούμενων flops για την επίτευξή του, αποτελεί μέτρο σύγκρισης της πολυπλοκότητας και είναι η μέγιστη επιτρεπτή πολυπλοκότητα σε αλγόριθμους γραμμικής άλγεβρας

Παρατήρηση : Εάν στο παράδειγμα 1 η άνω τριγωνοποίηση του

$$A = \begin{bmatrix} -1 & 4 & 1 \\ 3 & 1 & 1 \\ 2 & 6 & 1 \end{bmatrix}$$

γίνει με την απαλοιφή του Gauss, η πολυπλοκότητα θα είναι $\boxed{0\ (\frac{n^3}{3})}$.

Ενώ εάν γίνει με τον προσδιορισμό των πινάκων μετασχηματισμών, θα χρειαστούν 3 πολλαπλασιασμοί πινάκων διάστασης $n \times n$, οπότε η πολυπλοκότητα θα είναι $\boxed{0\ (3n^3)}$.

Συμπέρασμα : Σε μία σχέση μετασχηματισμού πίνακα A σε ισοδύναμη μορφή B οι πίνακες μετασχηματισμού μας προσδιορίζουν τη σχέση που συνδέει τους A και B. Στην υλοποίηση όμως, δεν υπολογίζουμε τους πίνακες μετασχηματισμών και μετασχηματίζουμε τον A σε επίπεδο συντεταγμένων δηλαδή εφαρμόζουμε κατευθείαν της γραμμοπράξεις στις γραμμές του.