

Układy równań liniowych „są wszędzie”

Wiele problemów matematycznych, fizycznych, chemicznych, inżynierskich czy ekonomicznych i statystycznych daje się sprowadzić do modeli wykorzystujących układy równań liniowych

Układ 3 równań z trzema niewiadomymi

$$2x + y - z = 8$$

$$-3x - y + 2z = -11$$

$$-2x + y + 2z = -3$$

Sprostowanie:

Nie Nobel z ekonomii a nagroda ;-)

Na konferencji „Large scale analysis and modeling” sponsorowanej przez IBM, nagrodzono pracę o oddzielaniu informacji genetycznej od środowiskowej dla krów mlecznych.

Uwzględniono informację od wszystkich spokrewnionych krów - wymagało to nie tylko ogromnej bazy danych, ale rozwiązania **układu równań o wymiarze 10 milionów!**

Ze względu na ogólność zapisu i wygodę wykorzystujemy zapis macierzowy

$$2x + y - z = 8$$

$$-3x - y + 2z = -11$$

$$-2x + y + 2z = -3$$

$$\mathbf{A} = \begin{pmatrix} 2 & 1 & -1 \\ -3 & -1 & 2 \\ -2 & 1 & 2 \end{pmatrix} \quad \mathbf{x} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad \mathbf{b} = \begin{pmatrix} 8 \\ -11 \\ -3 \end{pmatrix}$$

$$\mathbf{Ax} = \mathbf{b}$$

W ogólności mamy (n równań i n niewidomych):

$$\mathbf{Ax} = \mathbf{b}$$

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \cdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \cdots \\ b_n \end{bmatrix}$$

W ogólności mamy (n równań i n niewiadomych):

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}$$

- macierz
współczynników
układu
równań

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

- szukany
wektor
rozwiązań

$$b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

- wektor
wyrazów
wolnych

Parę słów o macierzach

Macierz $m \times n$: tablica m na n (m wierszy n kolumn) liczb

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} = [\mathbf{a}_1 \quad \mathbf{a}_2 \quad \dots \quad \mathbf{a}_n] \quad \mathbf{a}_i \in \mathfrak{R}^m$$

Macierz kwadratowa: $m=n$

Macierz trójkątna górna (dolna) – macierz kwadratowa, w której

$$\forall i, j \quad i > j \Rightarrow a_{ij} = 0 \quad (i < j \Rightarrow a_{ij} = 0)$$

Macierz transponowana \mathbf{A}^T : $(\mathbf{A}^T)_{ij}=a_{ji}$ $\mathbf{A}^T = [a'_{ij}] \Rightarrow a'_{ij} = a_{ji}$

Macierz symetryczna (zawsze kwadratowa): $a_{ij}=a_{ji}$

$$\mathbf{A} = \mathbf{A}^T$$

Macierz nieosobliwa: macierz o niezerowym wyznaczniku.

Macierz osobliwa – $\det(\mathbf{A}) = 0$

Macierz dodatnio określona:

$\mathbf{x}^T \mathbf{A} \mathbf{x} > 0$ dla każdego niezerowego wektora \mathbf{x} .

Norma euklidesowa macierzy: $\|\mathbf{A}\| = \sqrt{\sum_{i=1}^n \sum_{j=1}^n a_{ij}^2}$

Wskaźnik uwarunkowania macierzy $\text{cond } \mathbf{A} = \|\mathbf{A}\| \|\mathbf{A}^{-1}\|$

Macierz odwrotna – \mathbf{A}^{-1} : $\mathbf{A} \mathbf{A}^{-1} = \mathbf{A}^{-1} \mathbf{A} = \mathbf{E}$

Metody skończone/dokładne:

- WYZNACZNIKI CRAMERA
- Metoda Gaussa
- Metoda Gaussa-Jordana
- Metody Choleskiego

Metody iteracyjne dla dużych układów równań:

- Metoda Jacobiego
- Metoda Gaussa-Seidla

Singular Value Decomposition – SVD

Rozwiązanie ze szkoły podstawowej (SIŁOWE)

posługując się wzorami wyznacznikowymi Cramera

trzeba policzyć $n+1$ wyznaczników z macierzy

$n \times n$ i wykonać dodatkowo n dzieleni $x_i = \frac{|A_i|}{|A|}$

oszacujmy ile mnożeń trzeba wykonać,

obliczając wyznacznik macierzy z definicji

Układy równań liniowych – wyznaczniki Kramera

rozwijając wzgl. i -tej kolumny:

n razy mnożymy przez wyznaczniki stopnia $n-1$

każdy wyznacznik stopnia $n-1$:

$n-1$ razy mnożymy przez

...

$$n + n(n-1 + (n-1)(n-2 + (n-2)(\dots+3(2)\dots))$$

$$n + n(n-1) + n(n-1)(n-2) + \dots + n! < en!$$

ale

$$n! \approx \left(\frac{n}{e}\right)^n \sqrt{2\pi n}$$

Układy równań liniowych – wyznaczniki Kramera

dla $n=30$

$n! \sim 10^{31}$ mnożeń

SUN 5 - $\sim 10^7$ operacji zmiennoprzecinkowych / s

1 rok $\sim 3 * 10^7$ s, \Rightarrow w 1 roku $\sim 3 * 10^{14}$

mnożeń

ile to lat dla macierzy $n=30$?

$\sim 10^{15}$ lat.

konieczność użycia metod numerycznych

120 teraFlops= $1,2 * 10^{14}$ $\sim 10^7$ lat

Układ Równań o Macierzy Trójkątnej

$$\left\{ \begin{array}{l} u_{11}x_1 + u_{12}x_2 + \dots + u_{1,n-1}x_{n-1} + u_{1n}x_n = b_1 \\ \quad u_{22}x_2 + \dots + u_{2,n-1}x_{n-1} + u_{2n}x_n = b_2 \\ \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \dots \\ \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad u_{n-1,n-1}x_{n-1} + u_{n-1,n}x_n = b_{n-1} \\ \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad u_{nn}x_n = b_n \end{array} \right.$$

$$\left\{ \begin{array}{l} l_{11}x_1 = b_1 \\ l_{21}x_1 + l_{22}x_2 = b_2 \\ \dots \\ l_{n-1,1}x_1 + l_{n-1,2}x_2 + \dots + l_{n-1,n-1}x_{n-1} = b_{n-1} \\ l_{n1}x_1 + l_{n2}x_2 + \dots + l_{n,n-1}x_{n-1} + l_{nn}x_n = b_n \end{array} \right.$$

Układ Równań o Macierzy Trójkątnej

Proste do rozwiązania (podstawianie wstecz lub w przód)

$$x_n = \frac{b_n}{u_{nn}}$$
$$x_i = \frac{1}{u_{ii}} \left(b_i - \sum_{j=i+1}^n u_{ij} x_j \right) \quad i = n-1, n-2, \dots, 1$$

$$x_1 = \frac{b_1}{l_{11}}$$
$$x_i = \frac{1}{l_{ii}} \left(b_i - \sum_{j=1}^{i-1} l_{ij} x_j \right) \quad i = 2, 3, \dots, n$$

Układy równań liniowych - Metoda eliminacji Gaussa

Metoda eliminacji Gaussa polega na sprowadzeniu układu równań $AX=B$ do układu o postaci $A^{(n)}X=B^{(n)}$ gdzie $A^{(n)}$ jest macierzą trójkątną górną, a następnie rozwiązaniu tego trójkątnego układu równań.

$$\begin{array}{cccccccc} a_{11}^{(1)} x_1 & + & a_{12}^{(1)} x_2 & + & a_{13}^{(1)} x_3 & + & \cdots & + & a_{1n}^{(1)} x_n & = & b_1^{(1)} \\ a_{21}^{(1)} x_1 & + & a_{22}^{(1)} x_2 & + & a_{23}^{(1)} x_3 & + & \cdots & + & a_{2n}^{(1)} x_n & = & b_2^{(1)} \\ a_{31}^{(1)} x_1 & + & a_{32}^{(1)} x_2 & + & a_{33}^{(1)} x_3 & + & \cdots & + & a_{3n}^{(1)} x_n & = & b_3^{(1)} \\ \vdots & & \vdots & & \vdots & & \ddots & & \vdots & & \vdots \\ a_{n1}^{(1)} x_1 & + & a_{n2}^{(1)} x_2 & + & a_{n3}^{(1)} x_3 & + & \cdots & + & a_{nn}^{(1)} x_n & = & b_n^{(1)} \end{array}$$

zakładając że $a_{11}^{(1)} \neq 0$ pomnożmy pierwsze równanie przez $a_{i1}^{(1)} / a_{11}^{(1)}$ i odejmijmy od i -tego równania ($i = 2, 3, \dots, n$), otrzymamy wtedy

Układy równań liniowych - Metoda eliminacji Gaussa

otrzymujemy

$$\begin{array}{cccccccc} a_{11}^{(1)} x_1 & + & a_{12}^{(1)} x_2 & + & a_{13}^{(1)} x_3 & + & \cdots & + & a_{1n}^{(1)} x_n & = & b_1^{(1)} \\ & & a_{22}^{(2)} x_2 & + & a_{23}^{(2)} x_3 & + & \cdots & + & a_{2n}^{(2)} x_n & = & b_2^{(2)} \\ & & a_{32}^{(2)} x_2 & + & a_{33}^{(2)} x_3 & + & \cdots & + & a_{3n}^{(2)} x_n & = & b_3^{(2)} \\ & & \vdots & & \vdots & & \ddots & & \vdots & & \vdots \\ & & a_{n2}^{(2)} x_2 & + & a_{n3}^{(2)} x_3 & + & \cdots & + & a_{nn}^{(2)} x_n & = & b_n^{(2)} \end{array}$$

gdzie $a_{ij}^{(2)} = a_{ij}^{(1)} - \frac{a_{i1}^{(1)} a_{1j}^{(1)}}{a_{11}^{(1)}}, \quad b_i^{(2)} = b_i^{(1)} - \frac{a_{i1}^{(1)} b_1^{(1)}}{a_{11}^{(1)}} \quad (i, j = 2, 3, \dots, n).$

w 1. kroku eliminujemy 1. kolumnę

Układy równań liniowych - Metoda eliminacji Gaussa

Następnie pomnożmy drugie równanie przez $a_{i2}^{(2)} / a_{22}^{(2)}$ ($a_{22}^{(2)} \neq 0$) i odejmijmy od i-tego równania ($i = 3, 4, \dots, n$), otrzymamy więc

$$\begin{array}{rcccccccc} a_{11}^{(1)} x_1 & + & a_{12}^{(1)} x_2 & + & a_{13}^{(1)} x_3 & + & \cdots & + & a_{1n}^{(1)} x_n & = & b_1^{(1)} \\ & & a_{22}^{(2)} x_2 & + & a_{23}^{(2)} x_3 & + & \cdots & + & a_{2n}^{(2)} x_n & = & b_2^{(2)} \\ & & & & a_{33}^{(3)} x_3 & + & \cdots & + & a_{3n}^{(3)} x_n & = & b_3^{(3)} \\ & & & & \vdots & & \ddots & & \vdots & & \vdots \\ & & & & a_{n3}^{(3)} x_3 & + & \cdots & + & a_{nn}^{(3)} x_n & = & b_n^{(3)} \end{array}$$

gdzie $a_{ij}^{(3)} = a_{ij}^{(2)} - \frac{a_{i2}^{(2)} a_{2j}^{(2)}}{a_{22}^{(2)}}$, $b_i^{(3)} = b_i^{(2)} - \frac{a_{i2}^{(2)} b_2^{(2)}}{a_{22}^{(2)}}$ ($i, j = 3, 4, \dots, n$).

Układy równań liniowych - Metoda eliminacji Gaussa

Kontynuując takie postępowanie otrzymamy układ trójkątny

$$\begin{array}{ccccccccccc} a_{11}^{(1)} x_1 & + & a_{12}^{(1)} x_2 & + & a_{13}^{(1)} x_3 & + & \cdots & + & a_{1n}^{(1)} x_n & = & b_1^{(1)} \\ & & a_{22}^{(2)} x_2 & + & a_{23}^{(2)} x_3 & + & \cdots & + & a_{2n}^{(2)} x_n & = & b_2^{(2)} \\ & & & & a_{33}^{(3)} x_3 & + & \cdots & + & a_{3n}^{(3)} x_n & = & b_3^{(3)} \\ & & & & & & \ddots & & \vdots & & \vdots \\ & & & & & & & & a_{nn}^{(n)} x_n & = & b_n^{(n)} \end{array}$$

gdzie
$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - \frac{a_{ik}^{(k)} a_{kj}^{(k)}}{a_{kk}^{(k)}}, \quad b_i^{(k+1)} = b_i^{(k)} - \frac{a_{ik}^{(k)} b_k^{(k)}}{a_{kk}^{(k)}}, \quad a_{kk}^{(k)} \neq 0$$

$$(k = 1, 2, \dots, n-1 \quad i, j = k+1, k+2, \dots, n).$$

Układy równań liniowych - Metoda eliminacji Gaussa

Rozwiązanie tego układu równań jest proste. Rozwiązujemy go od „tyłu” tzn. obliczamy kolejno $x_n, x_{n-1}, x_{n-2}, \dots, x_1$ wg wzorów:

$$x_n = \frac{b_n^{(n)}}{a_{nn}^{(n)}}, \quad x_i = \frac{b_i^{(i)} - \sum_{j=i+1}^n a_{ij}^{(i)} x_j}{a_{ii}^{(i)}} \quad (i = n-1, n-2, \dots, 2, 1).$$

Dla „oszczędności” warto macierz **A** rozszerzyć o kolumnę będącą wektorem **b**

ogólnie (k – monitoruje, kolejny krok – nie jest realnym indeksem)

$$a_{ij}^{(k)} = a_{ij}^{(k-1)} - \frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}} a_{kj}^{(k-1)}$$

k = 1, 2, ..., n-1

k-ty krok, eliminacja k-tej kolumny

j = k+1, ..., n

Operujemy na i=k+1 wierszu (do n)

j = k+1, ..., n+1

Operujemy na j=k+1 kolumnie (do n+1) (**b**)

musimy zapewnić, żeby $a_{kk}^{(k-1)} \neq 0$

- x_n - znajdujemy z ostatniego równania
- x_{n-1} z równania (n-1) i znajomości x_n
- znając $x_n, x_{n-1}, \dots, x_{i+1}$

x_i znajdziemy z i-tego równania jako:

$$x_i = \frac{1}{a_{ii}^{(i-1)}} \left(a_{i,n+1} - \sum_{j=i+1}^n a_{ij}^{(i-1)} \cdot x_j \right)$$

i = n, n-1, ..., 1

Układy równań liniowych - Metoda eliminacji Gaussa

KOSZT METODY:

ilość mnożeń w obydwu etapach:

w k-tym kroku triangularyzacji wykonuje się

n-k dzieleni oraz

(n-k)(n-k+1) mnożeń

w sumie po n-1 krokach etapu I

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

liczba mnożeń w etapie II jest $\sim 1/2 n^2$

w sumie dla dużych n, liczba mnożeń $\sim n^3 / 3$

dla n=30

czas rozwiązania tego zagadnienia na SUN-ie

\sim ułamek sekundy

Układy równań liniowych - Metoda eliminacji Gaussa

```
{calculate the output matrix}
for i:=1 to MAXEQNS-1 do
  for j:=i+1 to MAXEQNS do
    begin
      temp:=a[j,i];
      for k:=1 to MAXX+1 do
        a[j,k]:=a[j,k] - temp*a[i,k]/a[i,i];
      end;
    end;
{calc the values of x1 x2 etc}
x[MAXEQNS] := a[MAXEQNS,MAXX+1] / a[MAXEQNS,MAXX];
for i:=MAXEQNS downto 1 do
  begin
    x[i]:=a[i,MAXX+1];
    for j:=i+1 to MAXX do
      begin
        x[i]:=x[i] - a[i,j]*x[j];
      end;
    x[i]:=x[i]/a[i,i];
  end;
end;
```

„PIVOTING”

(wybór elementu głównego)

Bezpośrednia „naiwna” metoda eliminacji nie działa dla

$$a_{kk} = 0$$

np.:

$$\begin{array}{rcl} 0x_1 & + & x_2 = 1 \\ x_1 & + & x_2 = 2 \end{array}$$

$$a_{11}=0$$

sprawdźmy dla $a_{11} = \varepsilon$ - bliskie zeru;
podstawianie wstecz prowadzi do:

$$x_2 = \frac{2 - 1/\varepsilon}{1 - 1/\varepsilon} \qquad x_1 = \frac{1 - x_2}{\varepsilon}$$

jeśli $\varepsilon \approx 0$ to $1/\varepsilon$ zdominuje 1 i 2

w rezultacie $x_2 \approx 1$, $x_1 \approx 0$ lub źle
określone

a ścisłe rozwiązanie: $x_2 = 1$, $x_1 = 1$

Aby uniknąć dzielenia przez zero lub przez małe liczby stosuje się metodę wyboru elementu podstawowego. Algorytm jest następujący:

1. W pierwszym kroku wyszukujemy element o maksymalnym module wśród wszystkich współczynników $a_{ij}^{(1)}$

$$\max_{i,j=1\dots n} |a_{ij}^{(1)}| = |a_{rs}^{(1)}|.$$

Niech nim będzie element leżący w r-tym wierszu i s-tej kolumnie. Zamieniamy r-ty wiersz z pierwszym a następnie s-tą kolumnę z pierwszą. Dalej dokonujemy redukcji macierzy wg wzoru [7].

2. W drugim kroku wyszukujemy elementu o maksymalnym module wśród wszystkich współczynników $a_{ij}^{(2)}$

$$\max_{i,j=2\dots n} |a_{ij}^{(2)}| = |a_{pq}^{(2)}|.$$

Zamieniamy odpowiednio p-ty wiersz z drugim wierszem i q-tą kolumnę z drugą itd.

Jeśli w którymś kroku znaleziony element o maksymalnym module jest równy zero to obliczenia przerywamy. Oznacza to bowiem, że $\det A = 0$, czyli układ jest albo sprzeczny albo nieoznaczony.

Należy również zapamiętywać wszystkie kolejne zamiany kolumn, ponieważ powodują one zmianę kolejności niewiadomych x_i w wektorze \mathbf{X} .

Metoda Gaussa -Jordana

Metoda ta stanowi pewną modyfikację metody Gaussa. Otóż przekształca się układ równań [6] do układu w którym macierz współczynników jest macierzą jednostkową

$$AX=B \rightarrow EX=B^{(n)}$$

wg następującego algorytmu. Dzielimy obustronnie pierwszy wiersz układu równań [6] przez współczynnik $a_{11}^{(1)}$ a następnie mnożymy przekształcony pierwszy wiersz przez współczynnik $a_{i1}^{(1)}$ i odejmujemy od i-tego wiersza ($i = 2, 3, \dots, n$). Otrzymamy

$$\begin{array}{rcccccc} x_1 & + & a_{12}^{(2)} x_2 & + & a_{13}^{(2)} x_3 & + & \dots & + & a_{1n}^{(2)} x_n & = & b_1^{(2)} \\ & & a_{22}^{(2)} x_2 & + & a_{23}^{(2)} x_3 & + & \dots & + & a_{2n}^{(2)} x_n & = & b_2^{(2)} \\ & & a_{32}^{(2)} x_2 & + & a_{33}^{(2)} x_3 & + & \dots & + & a_{3n}^{(2)} x_n & = & b_3^{(2)} \\ & & \vdots & & \vdots & & \ddots & & \vdots & & \vdots \\ & & a_{n2}^{(2)} x_2 & + & a_{n3}^{(2)} x_3 & + & \dots & + & a_{nn}^{(2)} x_n & = & b_n^{(2)} \end{array}$$

Układy równań liniowych Metoda Gaussa -Jordana

gdzie
$$a_{1j}^{(2)} = \frac{a_{1j}^{(1)}}{a_{11}^{(1)}} \quad (j = 2, 3, \dots, n) \quad b_1^{(2)} = \frac{b_1^{(1)}}{a_{11}^{(1)}} \quad a_{11}^{(1)} \neq 0$$
$$a_{ij}^{(2)} = a_{ij}^{(1)} - a_{i1}^{(1)} a_{1j}^{(2)}, \quad b_i^{(2)} = b_i^{(1)} - a_{i1}^{(1)} b_1^{(2)} \quad (i, j = 2, 3, \dots, n).$$

W otrzymanym układzie równań dzielimy drugi wiersz przez współczynnik $a_{22}^{(2)}$ a następnie mnożymy przekształcony drugi wiersz przez współczynnik $a_{i2}^{(2)}$ i odejmujemy od i -tego wiersza ($i = 1, 3, \dots, n$). Otrzymamy

$$\begin{array}{rcccccc} x_1 & + & & a_{13}^{(3)} x_3 & + & \cdots & + & a_{1n}^{(3)} x_n & = & b_1^{(3)} \\ & & x_2 & + & a_{23}^{(3)} x_3 & + & \cdots & + & a_{2n}^{(3)} x_n & = & b_2^{(3)} \\ & & & a_{33}^{(3)} x_3 & + & \cdots & + & a_{3n}^{(3)} x_n & = & b_3^{(3)} \\ & & & \vdots & & \ddots & & \vdots & & \vdots \\ & & & a_{n3}^{(3)} x_3 & + & \cdots & + & a_{nn}^{(3)} x_n & = & b_n^{(3)} \end{array}$$

Układy równań liniowych Metoda Gaussa -Jordana

Kontynuując obliczenia po „k” krokach otrzymamy

$$\begin{array}{rcccccc}
 x_1 & + & & a_{1,k+1}^{(k+1)} x_{k+1} & + \dots + & a_{1n}^{(k+1)} x_n & = & b_1^{(k+1)} \\
 & & x_2 & + & & a_{2,k+1}^{(k+1)} x_{k+1} & + \dots + & a_{2n}^{(k+1)} x_n & = & b_2^{(k+1)} \\
 & & & & \ddots & \vdots & & \vdots & = & \vdots \\
 & & & & & & x_k & + & a_{k,k+1}^{(k+1)} x_{k+1} & + \dots + & a_{kn}^{(k+1)} x_n & = & b_k^{(k+1)} \\
 & & & & & & & & \vdots & & \vdots & = & \vdots \\
 & & & & & & & & a_{n,k+1}^{(k+1)} x_{k+1} & + \dots + & a_{nn}^{(k+1)} x_n & = & b_n^{(k+1)}
 \end{array}$$

gdzie $a_{kj}^{(k+1)} = \frac{a_{kj}^{(k)}}{a_{kk}^{(k)}} \quad (j = k + 1, k + 2, \dots, n) \quad b_k^{(k+1)} = \frac{b_k^{(k)}}{a_{kk}^{(k)}} \quad a_{kk}^{(k)} \neq 0$

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - a_{ik}^{(k)} a_{kj}^{(k+1)}, \quad b_i^{(k+1)} = b_i^{(k)} - a_{ik}^{(k)} b_k^{(k+1)} \quad (i = 1, 2, \dots, n, i \neq k \quad j = k + 1, \dots, n)$$

Rozkład LU

Lower – Upper decomposition

$$\mathbf{A} \mathbf{x} = (\mathbf{L} \mathbf{U}) \mathbf{x} = \mathbf{L} (\mathbf{U} \mathbf{x}) = \mathbf{b}$$

1. $\mathbf{L} \mathbf{y} = \mathbf{b}$

2. $\mathbf{U} \mathbf{x} = \mathbf{y}$

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ L_{21} & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ L_{n1} & L_{n2} & \dots & 1 \end{bmatrix} \quad \mathbf{U} = \begin{bmatrix} U_{11} & U_{12} & \dots & U_{1n} \\ 0 & U_{22} & \dots & U_{2n} \\ \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & U_{nn} \end{bmatrix}$$

Tadeusz Banachiewicz (1938) – polski astronom, matematyk,

Z pierwszego równania wyznaczamy wektor Y a z drugiego wektor X . Zaletą takiego rozbicia jest fakt, że obydwa wektory otrzymujemy natychmiast, bowiem rozwiązujemy dwa układy równań z macierzami trójkątnymi:

$$\begin{cases} y_1 = b_1 \\ y_i = b_i - \sum_{k=1}^{i-1} l_{ik} y_k \quad i = 2, 3, \dots, n \end{cases}$$

$$\begin{cases} x_n = \frac{y_n}{u_{nn}} \\ x_i = \frac{y_i - \sum_{k=i+1}^n u_{ik} x_k}{u_{ii}} \quad i = n-1, n-2, \dots, 1 \end{cases}$$

Elementy macierzy L i U można wyznaczyć dwoma sposobami.

1. Istnieje ścisły związek między rozkładem macierzy A na macierze L i U a metodą eliminacji Gaussa. Można wykazać, że elementy kolejnych kolumn macierzy L są równe współczynnikom przez które mnożone są w kolejnych krokach wiersze układu równań celem dokonania eliminacji niewiadomych w odpowiednich kolumnach

$$l_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} \quad k = 1, 2, \dots, n \quad i = k + 1, \dots, n.$$

Natomiast macierz U jest równa macierzy trójkątnej uzyskanej w eliminacji Gaussa, czyli

$$u_{ki} = a_{ki}^{(k)} \quad k = 1, 2, \dots, n \quad i = k, k + 1, \dots, n.$$

- metoda Doolittle'a

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} = \begin{bmatrix} 1 & & & \\ l_{21} & 1 & & \\ \vdots & \vdots & \ddots & \\ l_{n1} & l_{n2} & \dots & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{21} & \dots & u_{1n} \\ & u_{22} & \dots & u_{2n} \\ & & \ddots & \vdots \\ & & & u_{nn} \end{bmatrix}$$

$$i = 1, \dots, n$$

$$u_{ij} = a_{ij} - \sum_{k=1}^{i-1} l_{ik} u_{kj}, j = i, i+1, \dots, n$$

$$l_{ji} = \left(a_{ji} - \sum_{k=1}^{i-1} l_{jk} u_{ki} \right) / u_{ii}, j = i+1, i+2, \dots, n$$

Metoda Doolittle'a

$$[A] = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ l_{21} & 1 & 0 & 0 \\ l_{31} & l_{32} & 1 & 0 \\ l_{41} & l_{42} & l_{43} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{bmatrix}$$

$$[A] = \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ l_{21}u_{11} & l_{21}u_{12} + u_{22} & l_{21}u_{13} + u_{23} & l_{21}u_{14} + u_{24} \\ l_{31}u_{11} & l_{31}u_{12} + l_{32}u_{22} & l_{31}u_{13} + l_{32}u_{23} + u_{33} & l_{31}u_{14} + l_{32}u_{24} + u_{34} \\ l_{41}u_{11} & l_{41}u_{12} + l_{42}u_{22} & l_{41}u_{13} + l_{42}u_{23} + l_{43}u_{33} & l_{41}u_{14} + l_{42}u_{24} + l_{43}u_{34} + u_{44} \end{bmatrix}$$

wiersz 1: $u_{11} = a_{11}; u_{12} = a_{12}; u_{13} = a_{13}; u_{14} = a_{14}$

Kolumna 1: $l_{21} = a_{21} / u_{11}; l_{31} = a_{31} / u_{11}; l_{41} = a_{41} / u_{11}$

Metoda Doolittle'a

$$[A] = \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ l_{21}u_{11} & l_{21}u_{12} + u_{22} & l_{21}u_{13} + u_{23} & l_{21}u_{14} + u_{24} \\ l_{31}u_{11} & l_{31}u_{12} + l_{32}u_{22} & l_{31}u_{13} + l_{32}u_{23} + u_{33} & l_{31}u_{14} + l_{32}u_{24} + u_{34} \\ l_{41}u_{11} & l_{41}u_{12} + l_{42}u_{22} & l_{41}u_{13} + l_{42}u_{23} + l_{43}u_{33} & l_{41}u_{14} + l_{42}u_{24} + l_{43}u_{34} + u_{44} \end{bmatrix}$$

Wiersz 2: $l_{21}u_{12} + u_{22} = a_{22}$; $l_{21}u_{13} + u_{23} = a_{23}$; $l_{21}u_{14} + u_{24} = a_{24}$

$$\begin{cases} u_{22} = a_{22} - l_{21}u_{12} \\ u_{23} = a_{23} - l_{21}u_{13} \\ u_{24} = a_{24} - l_{21}u_{14} \end{cases}$$

Kolumna 2: $l_{31}u_{12} + l_{32}u_{22} = a_{32}$; $l_{41}u_{12} + l_{42}u_{22} = a_{42}$

$$\begin{cases} l_{32} = (a_{32} - l_{31}u_{12}) / u_{22} \\ l_{42} = (a_{42} - l_{41}u_{12}) / u_{22} \end{cases}$$

Metoda Doolittle'a

$$[A] = \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ l_{21}u_{11} & l_{21}u_{12} + u_{22} & l_{21}u_{13} + u_{23} & l_{21}u_{14} + u_{24} \\ l_{31}u_{11} & l_{31}u_{12} + l_{32}u_{22} & l_{31}u_{13} + l_{32}u_{23} + u_{33} & l_{31}u_{14} + l_{32}u_{24} + u_{34} \\ l_{41}u_{11} & l_{41}u_{12} + l_{42}u_{22} & l_{41}u_{13} + l_{42}u_{23} + l_{43}u_{33} & l_{41}u_{14} + l_{42}u_{24} + l_{43}u_{34} + u_{44} \end{bmatrix}$$

Wiersz 3: $l_{31}u_{13} + l_{32}u_{23} + u_{33} = a_{33}$; $l_{31}u_{14} + l_{32}u_{24} + u_{34} = a_{34}$

$$\begin{cases} u_{33} = a_{33} - l_{31}u_{13} - l_{32}u_{23} \\ u_{34} = a_{34} - l_{31}u_{14} - l_{32}u_{24} \end{cases}$$

Kolumna 3: $l_{43} = (a_{43} - l_{41}u_{13} - l_{42}u_{23}) / u_{33}$

Wiersz 4: $u_{44} = a_{44} - l_{41}u_{14} - l_{42}u_{24} - l_{43}u_{34}$

Metoda Doolittle'a – algorytm:

```
for i = 1 ... n
  {
  Lii = 1
  for j = 1 ... i
    
$$U_{ji} = a_{ji} - \sum_{k=1}^{j-1} L_{jk} U_{ki}$$

  for j = i+1 ... n
    
$$L_{ji} = \frac{a_{ji} - \sum_{k=1}^{i-1} L_{jk} U_{ki}}{U_{ii}}$$

  }
```

```
for i = 1 ... n
  {
  for j = 1 ... i
    
$$a_{ji} = a_{ji} - \sum_{k=1}^{j-1} a_{jk} a_{ki}$$

  for j = i+1 ... n
    
$$a_{ji} = \frac{a_{ji} - \sum_{k=1}^{i-1} a_{jk} a_{ki}}{a_{ii}}$$

  }
```

Każdy element a_{ij} potrzebny jest tylko raz, zatem możemy wykorzystać jego miejsce w macierzy (tablicy) na przechowanie wartości L_{ij} lub U_{ij} .

- metoda Crouta

- macierz U posiada jedynki na przekątnej głównej

$$l_{i1} = a_{i1}, \quad i = 1, 2, \dots, n$$

$$u_{1j} = a_{1j} / l_{jj}, \quad j = 2, 3, \dots, n$$

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} = \begin{bmatrix} l_{11} & & & \\ l_{21} & l_{22} & & \\ \vdots & \vdots & \ddots & \\ l_{n1} & l_{n2} & \dots & l_{nn} \end{bmatrix} \begin{bmatrix} 1 & u_{21} & \dots & u_{1n} \\ & 1 & \dots & u_{2n} \\ & & \ddots & \vdots \\ & & & 1 \end{bmatrix}$$

$$j = 2, 3, \dots, n-1$$

$$l_{ij} = a_{ij} - \sum_{k=1}^{j-1} l_{ik} u_{kj}, \quad i = j, j+1, \dots, n$$

$$u_{jk} = \left(a_{jk} - \sum_{i=1}^{j-1} l_{ji} u_{ik} \right) / l_{jj}, \quad k = j+1, j+2, \dots, n$$

- zapamiętując rozkład LU możemy szybko rozwiązać wiele układów różniących się wektorem \mathbf{b}

Inne zastosowania rozkładu trójkątnego macierzy:

- obliczanie wyznacznika macierzy \mathbf{A}
 - $\det(\mathbf{A}) = \det(\mathbf{LU}) = \det(\mathbf{L})\det(\mathbf{U}) = \det(\mathbf{U})$

Układy równań liniowych – wykorzystanie do odwracania macierzy

odwracanie macierzy A^{-1}

$$A A^{-1} = I$$

$$\begin{bmatrix} a_{11} & \dots & a_{1n} \\ \dots & \dots & \dots \\ a_{n1} & \dots & a_{nm} \end{bmatrix} \cdot \begin{bmatrix} x_{11} & \dots & x_{1n} \\ \dots & \dots & \dots \\ x_{n1} & \dots & x_{nm} \end{bmatrix} = \begin{bmatrix} 1 & \dots & 0 \\ \dots & \dots & \dots \\ 0 & \dots & 1 \end{bmatrix}$$

n razy rozwiązujemy

$$\begin{bmatrix} a_{11} & \dots & a_{1n} \\ \dots & \dots & \dots \\ a_{n1} & \dots & a_{nm} \end{bmatrix} \cdot \begin{bmatrix} x_{1k} \\ \dots \\ x_{nk} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \leftarrow na \text{ } k\text{-tym}$$

Zatem należy rozwiązać układ:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \\ x_{41} & x_{42} & x_{43} & x_{44} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Dostajemy cztery układy równań różniące się tylko kolumną wyrazów wolnych

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} x_{11} \\ x_{21} \\ x_{31} \\ x_{41} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} x_{12} \\ x_{22} \\ x_{32} \\ x_{42} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} x_{13} \\ x_{23} \\ x_{33} \\ x_{43} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \qquad \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} x_{14} \\ x_{24} \\ x_{34} \\ x_{44} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Doskonale nadaje się to tego celu rozkład LU.

Zmieniają się tylko prawe strony !

obliczanie wyznacznika macierzy \mathbf{A}

1. operacja dodania wiersza (kolumny), pomnożonego przez liczbę, do innego wiersza (kolumny) nie zmienia wyznacznika macierzy

2. wyznacznik macierzy trójkątnej

$$\det \mathbf{T} = \prod_{i=1}^n T_{ii}$$

to etap triangularyzacji \mathbf{A} prowadzi do $\det \mathbf{A}$

Obliczenie wyznacznika macierzy trójkątnej sprowadza się do wymnożenia elementów leżących na głównej przekątnej:

$$\mathbf{L} = \begin{bmatrix} l_{1,1} & 0 & 0 & 0 & 0 \\ l_{2,1} & l_{2,2} & 0 & 0 & 0 \\ l_{3,1} & l_{3,2} & \ddots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & 0 \\ l_{n,1} & l_{n,2} & \dots & l_{n,n-1} & l_{n,n} \end{bmatrix}$$

$$\det(L) = \prod_{i=1}^n l_{i,i} = l_{1,1} \cdot l_{2,2} \cdot \dots \cdot l_{n,n}$$

$$\mathbf{U} = \begin{bmatrix} u_{1,1} & u_{1,2} & u_{1,3} & \dots & u_{1,n} \\ 0 & u_{2,2} & u_{2,3} & \dots & u_{2,n} \\ 0 & 0 & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \ddots & u_{n-1,n} \\ 0 & 0 & 0 & 0 & u_{n,n} \end{bmatrix}$$

$$\det(U) = \prod_{i=1}^n u_{i,i} = u_{1,1} \cdot u_{2,2} \cdot \dots \cdot u_{n,n}$$

Rozkład typu Choleskiego dla macierzy symetrycznych ($A = A^T$) dodatnio określonych ($X^T \cdot A \cdot X > 0$ dla każdego X)

$$\mathbf{A} = \mathbf{L}\mathbf{L}^T$$

$$Ax = b$$

$$LL^T x = b$$

$$l_{11} = \sqrt{a_{11}}$$

$$l_{j1} = \frac{a_{j1}}{l_{11}}$$

$$Ly = b$$

$$L^T x = y$$

$$l_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} l_{jk}^2}$$

$$l_{ji} = \frac{1}{l_{ii}} \left[a_{ij} - \sum_{k=1}^{i-1} l_{ik} l_{jk} \right]$$

Metoda wymaga ok. $n^3/6$ operacji (2 razy mniej niż metoda eliminacji Gaussa) i n pierwiastkowań.

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} l_{11} & & & \\ l_{21} & l_{22} & & \\ \vdots & \vdots & \ddots & \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{bmatrix} \begin{bmatrix} l_{11} & l_{21} & \cdots & l_{1n} \\ & l_{22} & \cdots & l_{2n} \\ & & \ddots & \vdots \\ & & & l_{nn} \end{bmatrix}$$

Elementy macierzy L wyznaczmy kolumnami:

I kolumna macierzy L :

$$l_{11} \cdot l_{11} = a_{11} \rightarrow l_{11} = \sqrt{a_{11}}$$

$$l_{21} \cdot l_{11} = a_{21} \rightarrow l_{21} = a_{21} / l_{11}$$

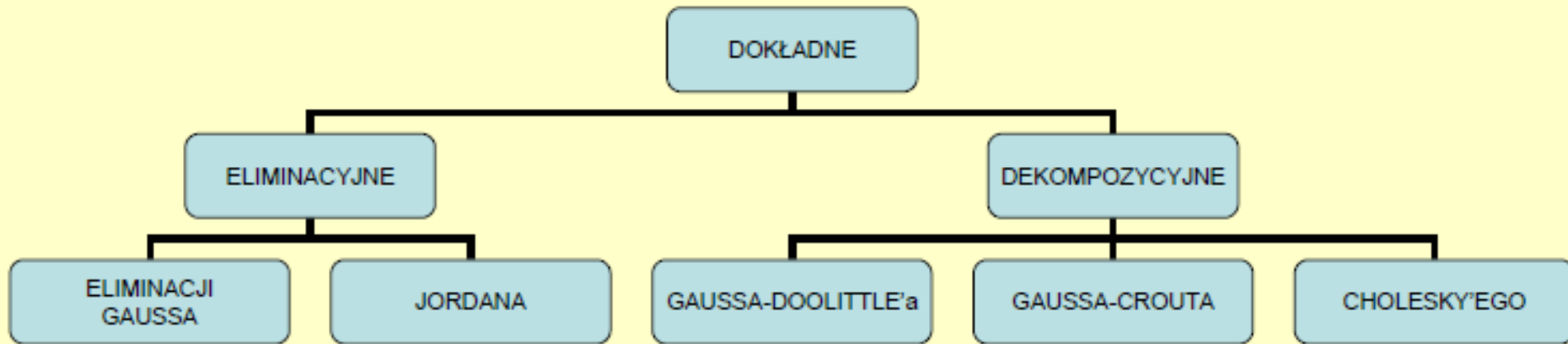
ogólnie:

$$l_{i1} \cdot l_{11} = a_{i1}, \quad i=2,3,\dots,n$$

zatem

$$l_{i1} = \frac{a_{i1}}{l_{11}}, \quad i=2,3,\dots,n$$

Rozwiązywanie układów równań liniowych – metody dokładne (nieiteracyjne)



- Uwaga na stabilność rozwiązań
- Macierze rzadkie - lepiej metody iteracyjne
- Obciążenie pamięci operacyjnej (ograniczenie co do wielkości układów równań)
- Niewielkie różnice pomiędzy metodami

Metody iteracyjne: iteracji Jacobiego oraz iteracji Gaussa-Seidla

Metody te stosuje się dla dużych układów równań, pojawiających się np. przy rozwiązywaniu równań różniczkowych cząstkowych lub równań całkowych.

Iteracja Jacobiego

1. Wyznaczamy przybliżenie początkowe $\mathbf{x}^{(0)}$
2. Kolejne przybliżenia $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots$, obliczamy wyznaczając x_i z i -tego równania dla $i=1, 2, \dots, n$ a następnie podstawiając po prawej stronie poprzednie przybliżenie \mathbf{x} .

$$x_i^{(p+1)} = - \sum_{\substack{k=1 \\ k \neq i}}^n \frac{a_{ik}}{a_{ii}} x_k^{(p)} + \frac{b_i}{a_{ii}}$$

3. Proces kończymy jeżeli $\|\mathbf{x}^{(p+1)} - \mathbf{x}^{(p)}\| < \delta$.

METODA ITERACJI PROSTEJ (JACOBIEGO)

$$\mathbf{Ax} = \mathbf{b}$$

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\ \dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = b_n \end{cases}$$

$$\begin{cases} x_1 = \frac{1}{a_{11}}(b_1 - a_{12}x_2 - a_{13}x_3 \dots - a_{1n}x_n) \\ x_2 = \frac{1}{a_{22}}(b_2 - a_{21}x_1 - a_{23}x_3 \dots - a_{2n}x_n) \\ \dots \\ x_n = \frac{1}{a_{nn}}(b_n - a_{n1}x_1 - a_{n2}x_2 \dots - a_{n,n-1}x_{n-1}) \end{cases}$$

$$\mathbf{x} = \mathbf{Cx} + \mathbf{g}$$

$$c_{ij} = \begin{cases} -\frac{a_{ij}}{a_{ii}} & i \neq j \\ 0 & i = j \end{cases}, \quad g_i = \frac{b_i}{a_{ii}}$$

DANE (WEKTOR STARTOWY)

$$\mathbf{x}^{(0)}$$

WYZNACZAMY

$$\mathbf{x}^{(k+1)} = \mathbf{C}\mathbf{x}^{(k)} + \mathbf{g}$$

$$\left\{ \begin{array}{l} X_1^{(k+1)} = \frac{1}{a_{11}} \left(b_1 - a_{12}X_2^{(k)} - a_{13}X_3^{(k)} \dots - a_{1n}X_n^{(k)} \right) \\ X_2^{(k+1)} = \frac{1}{a_{22}} \left(b_2 - a_{21}X_1^{(k)} - a_{23}X_3^{(k)} \dots - a_{2n}X_n^{(k)} \right) \\ \dots \\ X_n^{(k+1)} = \frac{1}{a_{nn}} \left(b_n - a_{n1}X_1^{(k)} - a_{n2}X_2^{(k)} \dots - a_{nn-1}X_{n-1}^{(k)} \right) \end{array} \right.$$

DOPÓKI

$$\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\| \leq \varepsilon$$

$$\mathbf{A}=\mathbf{L}+\mathbf{D}+\mathbf{U}$$

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 4 & 0 & 0 \\ 7 & 8 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 9 \end{bmatrix} + \begin{bmatrix} 0 & 2 & 3 \\ 0 & 0 & 6 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{Ax} = \mathbf{b} \Rightarrow (\mathbf{L} + \mathbf{D} + \mathbf{U})\mathbf{x} = \mathbf{b} \Rightarrow$$

$$\Rightarrow \mathbf{Dx} = -(\mathbf{L} + \mathbf{U})\mathbf{x} + \mathbf{b} \Rightarrow \mathbf{x} = -\mathbf{D}^{-1}(\mathbf{L} + \mathbf{U})\mathbf{x} + \mathbf{b}$$

$$\mathbf{Dx}^{(i+1)} = -(\mathbf{L} + \mathbf{U})\mathbf{x}^{(i)} + \mathbf{b},$$

Metoda iteracji Gaussa-Seidla.

Metoda ta różni się od metody iteracji Jacobiego tym, że kolejne przybliżenie i -tej współrzędnej wektora x obliczamy wykorzystując przybliżenia x_1, \dots, x_{i-1} wyliczone w aktualnej ($p+1$) iteracji oraz przybliżenia x_{i+1}, \dots, x_n z poprzedniej (p -tej) iteracji.

$$x_i^{(p+1)} = - \sum_{k=1}^{i-1} \frac{a_{ik}}{a_{ii}} x_k^{(p+1)} - \sum_{k=i+1}^n \frac{a_{ik}}{a_{ii}} x_k^{(p)} + \frac{b_i}{a_{ii}}$$

$$= \sum_{k=1}^{i-1} b_{ik}^{(l)} x_k^{(p+1)} + \sum_{k=i+1}^n b_{ik}^{(r)} x_k^{(p)} + c_i$$

$$b_{ik}^{(l)} = \begin{cases} -\frac{a_{ik}}{a_{ii}} & k < i \\ 0 & k \geq i \end{cases} \quad b_{ik}^{(r)} = \begin{cases} 0 & k \leq i \\ -\frac{a_{ik}}{a_{ii}} & k > i \end{cases} \quad c_i = \frac{b_i}{a_{ii}}$$

Metoda iteracji Gaussa-Seidla jest zawsze zbieżna dla macierzy dodatnio określonych.

Układy równań liniowych – met. iteracyjne

DANE

$$\mathbf{x}^{(0)}$$

WYZNACZAMY

$$\mathbf{x}^{(k+1)} = -\mathbf{D}^{-1}\mathbf{L}\mathbf{x}^{(k+1)} - \mathbf{D}^{-1}\mathbf{U}\mathbf{x}^{(k)} + \mathbf{b}$$

$$\left\{ \begin{array}{l} X_1^{(k+1)} = \frac{1}{a_{11}} \left(b_1 - a_{12}X_2^{(k)} - a_{13}X_3^{(k)} - a_{14}X_4^{(k)} \dots - a_{1n}X_n^{(k)} \right) \\ X_2^{(k+1)} = \frac{1}{a_{22}} \left(b_2 - a_{21}X_1^{(k+1)} - a_{23}X_3^{(k)} - a_{24}X_4^{(k)} \dots - a_{2n}X_n^{(k)} \right) \\ X_3^{(k+1)} = \frac{1}{a_{33}} \left(b_3 - a_{31}X_1^{(k+1)} - a_{32}X_2^{(k+1)} - a_{34}X_4^{(k)} \dots - a_{3n}X_n^{(k)} \right) \\ \dots \\ X_n^{(k+1)} = \frac{1}{a_{nn}} \left(b_n - a_{n1}X_1^{(k+1)} - a_{n2}X_2^{(k+1)} - a_{n3}X_3^{(k+1)} \dots - a_{nn-1}X_{n-1}^{(k+1)} \right) \end{array} \right.$$

DOPÓKI

$$\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\| \leq \varepsilon$$

Nie musimy trzymać 2 wektorów rozwiązań

METODA GAUSSA-SEIDLA

$$\mathbf{Ax} = \mathbf{b}$$

$$(\mathbf{L} + \mathbf{D})\mathbf{x}^{(i+1)} + \mathbf{U}\mathbf{x}^{(i)} = \mathbf{b}$$

$$\mathbf{x}^{(i+1)} = \mathbf{B}_{\text{GS}}\mathbf{x}^{(i)} + \mathbf{c}$$

$$\mathbf{B}_{\text{GS}} = -(\mathbf{L} + \mathbf{D})^{-1}\mathbf{U},$$

$$\mathbf{c} = (\mathbf{L} + \mathbf{D})^{-1}\mathbf{b}.$$

$$x_k^{(i+1)} = \frac{-\sum_{j=1}^{k-1} a_{kj}x_j^{(i+1)} - \sum_{j=k+1}^n a_{kj}x_j^{(i)} + b_k}{a_{kk}}, k = 1, 2, \dots, n.$$

Warunki wystarczające zbieżności procesu iteracyjnego w metodzie Seidela są identyczne jak w metodzie Jacobiego (wzór [28]). Zazwyczaj proces Seidela jest szybciej zbieżny niż proces iteracji prostej (Jacobiego). Często jest zbieżny podczas gdy proces Jacobiego jest rozbieżny.

Metoda SOR (nadrelaksacji)

Modyfikacja metody Gaussa-Seidela, przyspieszająca jej zbieżność

$$\omega(\mathbf{L} + \mathbf{D} + \mathbf{U})\mathbf{x} = \omega\mathbf{b} \Rightarrow$$

$$\mathbf{D}\mathbf{x} = \mathbf{D}\mathbf{x} - \omega[(\mathbf{L} + \mathbf{D} + \mathbf{U})\mathbf{x} - \mathbf{b}] \Rightarrow$$

$$(\mathbf{D} + \omega\mathbf{L})\mathbf{x}^{(i+1)} = (1 - \omega)\mathbf{D}\mathbf{x}^{(i)} - \omega\mathbf{U}\mathbf{x}^{(i)} + \omega\mathbf{b}$$

lub równoważnie

$$\mathbf{D}\mathbf{x}^{(i+1)} = (1 - \omega)\mathbf{D}\mathbf{x}^{(i)} - \omega(\mathbf{L}\mathbf{x}^{(i+1)} + \mathbf{U}\mathbf{x}^{(i)} - \mathbf{b})$$

$$x_k^{(i+1)} = x_k^{(i)} + \omega \frac{-\sum_{j=1}^{k-1} a_{kj}x_j^{(i+1)} - \sum_{j=k}^n a_{kj}x_j^{(i)} + b_k}{a_{kk}}, k = 1, 2, \dots, n,$$

Dla $\omega=1$ jest to metoda SOR (ang. *successive over relaxation*). Zwiększając współczynnik ω , można próbować przyspieszać jej zbieżność. Parametr ω może przyjmować wartości co najwyżej z przedziału $(0, 2)$.

Macierzą rzadką nazywamy macierz zawierającą dużo zer. Przykładami macierzy rzadkich są macierze wstęgowe, diagonalne, trójdzielne, trójkątne.

Wiele zagadnień (np. metody numeryczne służące do rozwiązywania równań różniczkowych cząstkowych) prowadzi do układów liniowych rzadkich, w których

Technika macierzy rzadkich pozwala na:

- oszczędne gospodarowanie pamięcią przez zapamiętanie tylko niezerowych współczynników układu, ich pozycji w macierzy oraz minimum danych umożliwiających efektywne docieranie do tych elementów. Wówczas zajętość pamięci jest proporcjonalna tylko do liczby elementów niezerowych;
- wykonywanie operacji tylko na elementach niezerowych macierzy i wektora prawych stron, prowadzące do zmniejszenia liczby operacji w stosunku równym mierze rzadkości;
- skuteczny i szybki wybór elementów podstawowych przy zachowaniu rzadkości macierzy.

W przypadku macierzy pełnych, liczba obliczeń potrzebna do uzyskania rozwiązania metodą iteracyjną jest zwykle znacznie większa niż przy stosowaniu metod dokładnych. Jednak w przypadku macierzy rzadkich metody iteracyjne mogą być lepsze niż metody dokładne.

Nie nakład obliczeń jednak decyduje o tym (choć zwykle jest mniejszy), lecz fakt, że podczas obliczeń nie zmieniamy położenia elementów macierzy \mathbf{A} układu równań $\mathbf{Ax}=\mathbf{b}$, a zatem zachowujemy jej rzadką strukturę.

Singular Value Decomposition

Gaussian-elimination and LU decomposition do not work for singular matrices

- singular matrix:
- matrix without an inverse
 - will have determinant that is zero:
 - at least one row or column with all elements = 0
 - two or more linearly dependent rows or columns
 - if we can't invert a matrix, we can't solve $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$
 - another way to look at it, is that a singular matrix will have at least one pivot that = 0

$$\langle i \rangle = \langle i \rangle - \frac{u_{ij}}{u_{jj}} \langle j \rangle \rightarrow \infty$$

- one way to deal with singular matrices (or matrices that are nearly singular) is to use singular value decomposition

- singular value decomposition is based on a theorem in linear algebra which states that an $M \times N$ matrix can be decomposed as: $\mathbf{A} = \mathbf{U}\mathbf{W}\mathbf{V}^T$

- \mathbf{A} = the $M \times N$ matrix to be decomposed
- \mathbf{U} = an $M \times M$ column-orthonormal matrix
- \mathbf{W} = an $M \times N$ diagonal matrix
- \mathbf{V} = an $N \times N$ column-orthonormal matrix

some definitions use different dimensions for these matrices, but this is consistent with LAPACK libraries

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{M1} & a_{M2} & \cdots & a_{MN} \end{pmatrix} = \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1M} \\ u_{21} & u_{22} & \cdots & u_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ u_{M1} & u_{M2} & \cdots & u_{MM} \end{pmatrix} \begin{pmatrix} w_1 & 0 & \cdots & 0 & 0 \\ 0 & w_2 & \cdots & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & w_M & 0 \end{pmatrix} \begin{pmatrix} v_{11} & v_{21} & \cdots & v_{N1} \\ v_{12} & v_{22} & \cdots & v_{N2} \\ \vdots & \vdots & \ddots & \vdots \\ v_{1N} & v_{2N} & \cdots & v_{NN} \end{pmatrix}$$

- a column-orthonormal matrix satisfies: $\sum_{i=1}^M U_{ik} U_{in} = \delta_{kn}$ $w_1 > w_2 > \dots > w_M > 0$

we won't look at how to do this decomposition right now, but will use SVD for linear equations

Wielkości w_i nazywamy wartościami szczególnymi (osobliwymi) macierzy $\mathbf{A}^T\mathbf{A}$,

rozkład $\mathbf{A}=\mathbf{U}\mathbf{W}\mathbf{V}^T$ nazywamy rozkładem według wartości szczególnych (SVD)

Można pokazać, że

w_i są pierwiastkami wartości własnych macierzy $\mathbf{A}^T\mathbf{A}$

kolumny macierzy \mathbf{V} odpowiadającymi im ortonormalnymi wektorami własnymi tej macierzy.

kolumny \mathbf{U} są wektorami własnymi macierzy $\mathbf{A}\mathbf{A}^T$

Algorytm znajdowania SVD macierzy $\mathbf{A} \in \mathbf{R}^{m \times n}$

- 1) Znajdujemy wartości własne λ_i macierzy $\mathbf{A}^T \mathbf{A}$ albo $\mathbf{A} \mathbf{A}^T$ (w zależności od tego, która macierz ma niższy stopień).
- 2) Określamy liczbę r niezerowych wartości własnych macierzy $\mathbf{A}^T \mathbf{A}$ (lub $\mathbf{A} \mathbf{A}^T$).
- 3) Znajdujemy ortonormalne wektory własne macierzy $\mathbf{A}^T \mathbf{A}$ (lub $\mathbf{A} \mathbf{A}^T$) odpowiadające znalezionym wartościom własnym. Tworzymy macierz ortogonalną (unitarną) $\mathbf{V} \in \mathbf{R}^{n \times n}$, której kolejne kolumny tworzą wektory własne macierzy $\mathbf{A}^T \mathbf{A}$ uporządkowane w malejącym porządku odpowiadających im wartości własnych.
- 4) Tworzymy pseudodiagonalną macierz \mathbf{W} umieszczając na diagonalnej pierwiastki kwadratowe $w_i = \sqrt{\lambda_i}$ z wartości własnych macierzy $\mathbf{A}^T \mathbf{A}$ (lub $\mathbf{A} \mathbf{A}^T$) w porządku malejącym.

- let's focus on cases where \mathbf{A} is an $N \times N$ matrix
- we need to get \mathbf{A}^{-1} so we can solve $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$

$$\mathbf{A} = \mathbf{U}\mathbf{W}\mathbf{V}^T \rightarrow \left((\mathbf{V}^T)^{-1} \mathbf{W}^{-1} \mathbf{U}^{-1} \right) (\mathbf{U}\mathbf{W}\mathbf{V}^T) = \mathbf{I} \rightarrow \mathbf{A}^{-1} = (\mathbf{V}^T)^{-1} \mathbf{W}^{-1} \mathbf{U}^{-1}$$

- \mathbf{U} and \mathbf{V} are square orthogonal matrices, so their inverses equal their transposes
- \mathbf{W} is a diagonal matrix, so its inverse is another diagonal matrix whose diagonal elements are $1/w_i$

$$\mathbf{A}^{-1} = \mathbf{V} \left[\text{diag}(1/w_i) \right] \mathbf{U}^T$$

- this definition poses problems if at least one w_i is zero \rightarrow singular
- formally we assess whether a matrix is singular by using the condition number:

$$k(\mathbf{A}) = \frac{\max(|w_i|)}{\min(|w_i|)} \left. \begin{array}{l} k(\mathbf{A}) = 0 \rightarrow \text{singular} \\ k(\mathbf{A}) < \varepsilon \rightarrow \text{ill-conditioned} \end{array} \right\}$$

- to see how SVD can be used for singular or ill-conditioned matrices, consider solving a set of linear equations with an ill-conditioned matrix **A**
- the equation $\mathbf{Ax} = \mathbf{b}$ maps the vector space **x** onto the vector space **b**
- for a singular **A**, there will be some subspace of **x** that is mapped to zero – this is called the nullspace and its dimension is called the nullity
- the subspace of **A** that can be mapped onto **b** is called the range, and its dimension is the rank
- nullity + rank = N

- SVD constructs orthonormal bases for the range and nullspace of a matrix
- the bases for the range correspond to the columns of U with indices that match non-zero w_j elements of W
- the bases for the nullspace correspond to the columns of V with indices that match zero w_j elements of W
- using these two different basis sets, one can solve for \mathbf{x} provided that \mathbf{b} is in the range of \mathbf{A}
- there will be an infinite set of solutions because the columns of \mathbf{V} can be added to \mathbf{x} in any linear combination
- it is common practice to solve for \mathbf{x} with the lowest length $|\mathbf{x}|^2$
- to do this, for all elements of \mathbf{W} where $w_j = 0$, set $1/w_j = 0$
- then solve: $\mathbf{x} = \mathbf{V} \times \left[\text{diag}(1/w_j) \right] \times (\mathbf{U}^T \times \mathbf{b})$

$$\tilde{\mathbf{A}}^{-1} = \mathbf{V} [\text{diag}(\tilde{w}_i^{-1})] \mathbf{U}^T. \quad (10a)$$

$$\tilde{w}_i^{-1} = \begin{cases} w_i^{-1} & \text{gdy } w_i \neq 0, \\ 0 & \text{gdy } w_i = 0. \end{cases} \quad (10b)$$

Jeśli macierz jest źle uwarunkowana, ale *formalnie* odwracalna, numeryczne rozwiązanie równania $\mathbf{Ax} = \mathbf{b}$ może być zdominowane przez wzmacniony błąd zaokrąglenia. Aby tego uniknąć, często zamiast (bezużytecznego!) rozwiązania dokładnego (9), używa się *przybliżonego* (i użytecznego!) rozwiązania w postaci (10) z następującą modyfikacją

$$\tilde{w}_i^{-1} = \begin{cases} w_i^{-1} & \text{gdy } |w_i| > \tau, \\ 0 & \text{gdy } |w_i| \leq \tau, \end{cases} \quad (14)$$

gdzie τ jest pewną zadaną tolerancją.

Mamy rozwiązać następujące dwa układy równań:

$$\begin{bmatrix} 0.66666667 & -0.16666666 & -0.33333333 \\ -0.16666666 & 0.16666667 & -0.16666666 \\ -0.33333333 & -0.16666666 & 0.66666667 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1.284457048 \\ -0.577350273 \\ -0.129756514 \end{bmatrix} \quad (15a)$$

$$\begin{bmatrix} 0.66666667 & -0.16666666 & -0.33333333 \\ -0.16666666 & 0.16666667 & -0.16666666 \\ -0.33333333 & -0.16666666 & 0.66666667 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1.284457052 \\ -0.577350265 \\ -0.129756510 \end{bmatrix} \quad (15b)$$

Równania te różnią się tylko prawymi stronami, przy czym norma różnicy prawych stron jest rzędu 10^{-8} . Błąd takich rozmiarów łatwo może pojawić się w wyniku jakichś poprzednich obliczeń lub na skutek niepewności danych “zewnętrznych”, z którymi pracujemy.

Macierz w układach równań (15) jest symetryczna i dodatnio określona,

$$\begin{bmatrix} 0.816496581 \\ -0.204124145 & 0.353553392 \\ -0.408248290 & -0.707106777 & 0.000077460 \end{bmatrix} \quad (16)$$

Rozwiązania równań (15) za pomocą faktoryzacji Cholesky'ego mają postać

$$x_a = \begin{bmatrix} -0.179434106 \\ -5.237183389 \\ -1.593647668 \end{bmatrix}, \quad x_b = \begin{bmatrix} 3.903048668 \\ 2.927782158 \\ 2.488835105 \end{bmatrix}. \quad (17)$$

Różnica rozwiązań jest, co do normy, rzędu 10, czyli jest rzędu 10^9 razy większa, niż różnica prawych stron.

Faktoryzacja *SVD* macierzy z układów (15) pokazuje, że wartości szczególne tej macierzy są w przybliżeniu równe $1, \frac{1}{2}, 10^{-9}$. Jeśli do rozwiązania układów równań (15) zastosować pseudoodwrotność (14) ($\tau = 10^{-8}$), w obu wypadkach otrzymamy

$$x = \begin{bmatrix} 1.861807320 \\ -1.154700538 \\ 0.447593757 \end{bmatrix}. \quad (18)$$

(18) jest jedynie *przybliżonym* rozwiązaniem równań (15). Jest ono jednak bardziej użyteczne, niż “ściśle” rozwiązania (17). Te dwa ostatnie najwyraźniej są zdominowane przez błąd, jaki wystąpił wzdłuż kierunku odpowiadającego najmniejszej wartości szczególnej macierzy. Nie wiemy — i nie mamy sposobu, aby to stwierdzić — które z dwu rozwiązań (17) jest “poprawne”. Przybliżone rozwiązanie (18) po prostu ignoruje wpływ tego kierunku, a więc i zaburzeń wzdłuż niego występujących.

- Przybliżone rozkłady SVD znajdują zastosowanie do analizy, kompresji i przesyłu obrazów w telekomunikacji i informatyce.
- Mogą być również z powodzeniem stosowane do analizy ekspresji genów w biologii czy też do analizy sygnałów EEG albo obrazów NMR mózgu w medycynie.



Rys. 10.3. Zastosowanie rozkładu SVD do analizy obrazów. Obraz można interpretować jako (dużą) macierz (lub kilka macierzy), której elementami są liczby odpowiadające jasności danego piksela albo współrzędnymi RGB. Pierwszy rysunek pokazuje oryginalny obraz (macierz RGB) a następne w kolejności zredukowany rozkład SVD tego obrazu obcięty do 1, 2, 4, 8, 16, 32, 64, 128 największych wartości szczególnych. Ostatni rysunek otrzymano z pełnego rozkładu SVD.

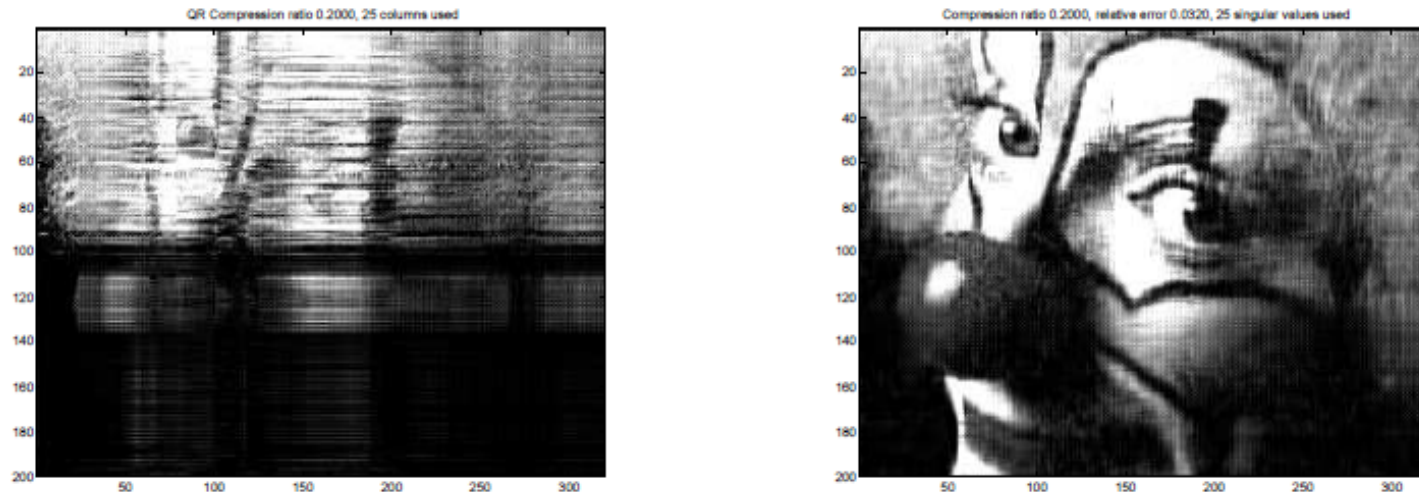


Figure 3: Image compressed using QR factorization (left) and SVD (right).

Besides applications to inconsistent and underdetermined linear systems and least squares problems, the SVD has important applications in image and data compression (see our discussion of low-rank approximation below). Figure 3 shows the difference between using the SVD and the QR factorization (to be introduced later) for compression of the same image. In both cases the same amount (20%) of information was retained. Clearly, the SVD does a much better job in picking out what information is “important”. We will also see below that a number of theoretical facts about the matrix A can be obtained via the SVD.

Singular value decomposition (SVD) complements association analysis by providing another method to identify items that have an affinity for each other. Singular value decomposition of the transaction item matrix reduces the matrix to a manageable number of dimensions, thereby enabling you to group similar transactions and similar items.

http://www.jmp.com/support/help/13/Additional_Example_3a_SVD_Analysis.shtml

SINGULAR VALUE DECOMPOSITION(SVD)

- × Any $N \times d$ matrix X can be uniquely expressed as:

$$\mathbf{X} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T$$

The diagram shows the equation $\mathbf{X} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T$ with dimensions indicated above each matrix:

- \mathbf{X} is $N \times d$ (represented by a tall rectangle).
- \mathbf{U} is $N \times r$ (represented by a tall rectangle).
- $\mathbf{\Sigma}$ is $r \times r$ (represented by a small square).
- \mathbf{V}^T is $r \times d$ (represented by a wide rectangle).

The matrices are connected by an equals sign and multiplication signs: $\mathbf{X} = \mathbf{U} \times \mathbf{\Sigma} \times \mathbf{V}^T$.

- × r is the **rank** of the matrix X (# of linearly independent columns/rows).
 - + \mathbf{U} is a column-orthonormal $N \times r$ matrix.
 - + $\mathbf{\Sigma}$ is a **diagonal** $r \times r$ matrix where the **singular values** σ_i are sorted in descending order.
 - + \mathbf{V} is a column-orthonormal $d \times r$ matrix.