SVD-Algorithmen

1 Householder Reflexionen (Alston Householder 1958)

Ziel: Zerlegung einer Matrix $A \in \mathbb{C}^{m \times n}$ in eine unitäre Matrix Q $\in \mathbb{C}^{m \times n}$ und eine obere Dreiecksmatrix $R \in Gl(n; \mathbb{C})$, so dass $A = Q * R$.
Verfahren:[1] Wir iterieren auf A spaltenweise mit

$$Q_k = \begin{pmatrix} I & 0 \\ 0 & F \end{pmatrix}$$

so dass I die $(k-1) \times (k-1)$ Einheitsmatrix und F der unitäre $(m-k+1) \times (m-k+1)$ Householder Reflektor ist.Sei $x \in \mathbb{C}^{(m-k+1)}$ der Vektor, der im Schritt k die Einträge k bis m der Spalte k von A beinhaltet. Für F angewandt auf x soll gelten:

$$F * x = \|x\| * e_1$$

Mit $v = \|x\| * e_1 - x$ und $F = (I - 2 * \frac{v * v'}{v' * v})$ wird x durch Q_k auf $\|x\| * e_1$ reflektiert. Durch die Einheitsmatrix werden außerdem alle bereits eliminierten Einträge beibehalten.

Verbesserung: Falls $x + v \approx x$ kann es durch die Subtraktion zur Auslöschung kommen. Da es zu jedem Vektor x mehr als einen orthogonalen Vektor $\hat{x}$ gibt, wählt man denjenigen aus, der den größten Abstand zu x besitzt. Man ist an der Richtung von $\hat{x}$ interessiert, deswegen wird die Signumsfunktion auf die erste Komponenete von x angewendet: $v = -sign(x_1) * \|x\| - x$, wobei gelte $sign(x_1) = 1$, wenn $x_1 = 0$. Multiplikation mit -1 liefert: $v = +sign(x_1) * \|x\| + x$.

Das Householder Verfahren benötigt insgesamt $\begin{cases} 2 * n^2 * m \text{ falls } m >> n \\ \frac{4}{3} * n^3 \text{ falls } m \approx n \end{cases}$.

Da Q oft nicht benötigt wird, ist eine explizite Berechnung meist nicht notwendig.
Damit lautet der Algorithmus (in MATLAB):

```
function [R] = householderQR(A)
[m,n]=size(A);
I=eye(m,n);

for k=1:n
   x=A(k:m,k);
   v=sign(x(1,1))*norm(x,2)*I(k:end,k)+x;
   v=v/norm(v,2);
   A(k:m,k:n)=A(k:m,k:n)-2*v*(v'*A(k:m,k:n));
end
R=triu(A);
end
```

2 SVD - Algorithmen

Es sei in diesem Abschnitt[2] $A \in \mathbb{C}^{m \times n}$ mit $m >= n$.

[1]Quelle: Trefethen-Bau Lecture Lecture 10
[2]Quelle: Trefethen-Bau Lecture 31 und Deuflhard-Hohmann Numerische Mathematik 1 Kapitel 5.4

2.1 Algorithmus 1

Wiederholung:Aus Satz 3.5 des letzten Vortrags ist bekannt: Die Singulärwerte von A sind die positiven Quadratwurzeln der absoluten Eigenwerte von $A' * A$.

Aus diesem Satz kann ein erster Algorithmus abgeleitet werden. Algorithmus 1:

1. Berechnung von $A' * A$
2. Eigenwertberechnung von $A' * A = Q * \Lambda * Q'$
3. Da Λ Diagonalmatrix mit $\Lambda = \Sigma' * \Sigma$, bezeichne fortan Σ die nicht-negative Quadratwurzel der Diagonalelemente von Λ
4. Lösung des Gleichungssystems $U * \Sigma = A * Q$ nach U (unitär) mit QR-Zerlegung

Dieser Algortihmus ist für Singulärwerte σ_k mit $\sigma_k << \|A\|$ instabil.

2.2 Algorithmus 2

Für quadratische Matrizen A gibt es eine stabile Variante, die A und A' verwendet. Sei dazu

$$H := \begin{bmatrix} 0 & A' \\ A & 0 \end{bmatrix}$$

Da die SVD $A = U * \Sigma * V'$ äuqivalent zu $A * V = U * \Sigma$ ist und $A' * U = V * \Sigma' = V * \Sigma$, folgt die Blockgleichheit:

$$\begin{bmatrix} 0 & A' \\ A & 0 \end{bmatrix} * \begin{bmatrix} V & V \\ U & -U \end{bmatrix} = \begin{bmatrix} V & V \\ U & -U \end{bmatrix} * \begin{bmatrix} \Sigma & 0 \\ 0 & -\Sigma \end{bmatrix}$$

Es wird nun also direkt mit A gearbeitet und nicht über den Umweg $A' * A$ und das Singulärwertproblem auf die Eigenwertzerlegung von H geführt. Die Beträge der Eigenwerte von H sind die Singulärwerte von A und die Singulärvektoren können aus den Eigenvektoren von H extrahiert werden (Siehe Blockgleichheit). In der Praxis werden keine Matrizen der Größe m+n generiert.

2.3 Algorithmus 3

Der dritte Algorithmus ist in zwei Schritte unterteilt und gilt als Standard.

Schritt 1: Golub-Kahan Bidiagonalisierung

Im ersten Schritt wird die Matrix A auf Bidiagonalgestalt gebracht. Dafür werden abwechselnd **n** Householdertransformationen von links und **n-2** von rechts auf A angewendet.
Schritt 1 benötigt dabei $O(m * n)$ flops.
Eine schnellere Variante ist durch das sog. LHC-Verfahren[3] gegeben.

[3] Siehe Terefethen-Bau S.237 ff

Schritt 2

Im Schritt 2 muss A auf Diagonalgestalt gebracht werden. Für diesen Schritt gibt es viele unterschiedliche Verfahren[4]. Hier wird ein abgeänderter QR-Algorithmus vorgestellt, das sog. „chasing“-Verfahren.
Das chasing Verfahren verwendet abwechselnd Givenstransformationen von rechts und links. Dabei gilt im Schritt k: Die Transformation von rechts eliminiert das Element (k,k+1) und erzeugt das Element (k+1,k), während die Transformation von links das erzeugte Element (k+1,k) eliminiert und dafür das Element (k,k+2) erzeugt. Diese Transformationen „jagen sich die Diagonale herunter“daher die Namensgebung des Verfahrens.

Satz 2.1 *Das chasing Verfahren konvergiert angewendet auf eine Bidiagonalmatrix A gegen eine Diagonalmatrix mit den Singulärwerten von A auf der Diagonalen.*

Programmcode in MATLAB für Schritt 2 (Skizze):

```
function [B] = gk_step2(B)
[m,n]=size(B);
y=zeros(n-1,1);
j=1;

%Abbruchbedingung, falls alle Superdiagonalelemente klein genug.
while sum(y)<(n-1)
    for i=1:(n-1)
        % Berechne Givens von rechts
        omega=givens(B(i,i),B(i,i+1));
        Q=eye(n);
        Q(i:i+1,i:i+1)=omega;
        B=B*Q';

        % Berechne Givens von links
        omega = givens(B(i,i),B(i+1,i));
        Q=eye(m);
        Q(i:i+1,i:i+1)=omega;
        B=Q*B;

        if abs(B(i,i+1))<=eps*(abs(B(i,i))+abs(B(i+1,i+1)))
            y(i)=1;
        end
    end
end
B=tril(triu(abs(B)));
end
```

[4]Siehe „Handbook of Linear Algebra“, Kapitel 45, Leslie Hogben, ISBN: 1-58488-510-6