Prof. Dr. Reinhard Klein
Alexander Schier

Universität Bonn
Institut für Informatik II
Wintersemester 2023/2024 01.11.2023

Übungen zu Angewandte Mathematik: Numerik - Blatt 4

Die Lösungen für die praktischen Aufgaben müssen bis Mittwoch, den 08.11.2023, um 12:00 im eCampus hochgeladen werden. Die Lösungen zu Theorieaufgaben müssen bis 12:00 in die Postfächer im Raum 0.004 im Hörsaalgebäude eingeworfen oder digital im eCampus abgegeben werden. Bei digitaler Abgabe werden werden keine Scans, Fotos, etc. gewertet.

Aufgabe 1 (Sparversion der SVD, *3+1=4 Punkte*)

In dieser Aufgabe untersuchen wir, wie man die Singulärwertzerlegung einer $\mathbb{C}^{m\times n}$ Matrix mit $m \geq n$ schneller durchführen kann und warum. Gegeben sei die Matrix A:

$$\mathbf{A} := \begin{pmatrix} 1 & 0 \\ 2 & 1 \\ 0 & 1 \end{pmatrix}$$

a) Berechne die Singulärwertzerlegung $U\Sigma V^T = A$ von Hand. Gib alle wichtigen Zwischenschritte an. Nutze dazu den Algorithmus im Skript (Kapitel 3.2) mit folgenden Änderungen: Die Vektoren u_{n+1} bis u_m aus dem 5. Schritt werden nicht berechnet. Stattdessen werden in die U Matrix $*$ als Platzhalter eingesetzt.

b) Rekonstruiere aus der Zerlegung wieder die Matrix $A = U\Sigma V^T$. Was fällt auf?

Hinweis: Wurzeln müssen *nicht* ausgerechnet werden.

Aufgabe 2 (Lineare Gleichungssysteme mit SVD lösen, *4 Punkte*)

Implementiere eine Funktion `LinearSolve(A,b)`, die zu einem gegebenen linearen Gleichungssystem $A \cdot x = b$ mit $A \in \mathbb{C}^{m\times n}$ und $b \in \mathbb{C}^m$ die Lösung $x \in \mathbb{C}^n$ (wir nehmen an, dass diese existiert, d.h. $m \geq n$) berechnet und zurückgibt.

Dazu müssen die folgenden Teilaufgaben gelöst werden:

a) Berechne die SVD von A. Verwende dazu die Eigenwertzerlegung von $A^T A$ und die Ideen und Folgerung aus der Aufgabe 1.

b) Schreibe eine Funktion `PseudoInverse(A)`, die die Pseudo-Inverse A^+ von A mit Hilfe ihrer SVD berechnet.

c) Nutze die Pseudo-Inverse und schreibe damit die Funktion `LinearSolve(A,b)`. Dazu kann das lineare Gleichungssystem als lineares Ausgleichsproblem $\|A \cdot x - b\|_2 \to \min$ angesehen werden.

Hinweis: Um eine Eigenwertzerlegung durchzuführen, darf die Funktion `numpy.linalg.eig` verwendet werden.

Aufgabe 3 (Gewichtetes Fitten von Funktionen, *4 Punkte*)

Wenn man sich nur für das Verhalten der Messungen in einem kleinen Bereich interessiert, bietet es sich an nur Messungen aus diesem Bereich stark zu gewichten. Gegeben sei eine beliebige Menge D, Funktionen $f_1, \ldots, f_n : D \to \mathbb{R}$, Messpunkte $u_1, \ldots, u_m \in D$ und zugehörige Messwerte $v_1, \ldots, v_m \in \mathbb{R}$. Ferner seien Gewichte für die Messpunkte $w_1, \ldots, w_m \in \mathbb{R}$ gegeben, d.h. $w_i \geq 0$ gibt an wie stark die Messung (u_i, w_i) berücksichtigt werden soll.

Wir suchen $(x_1, \ldots, x_n)^T = x \in \mathbb{R}^n$ so, dass die Funktion $f := \sum_{j=1}^{n} x_j \cdot f_j$ eine optimale Approximation der Messungen im Sinne der kleinsten gewichteten Quadrate ist, d.h.

$$E(x) := \sum_{i=1}^{m} w_i \cdot \left| v_i - \sum_{j=1}^{n} x_j \cdot f_j(u_i) \right|^2$$

soll minimiert werden. Zeige, dass für

$$A := \begin{pmatrix} \sqrt{w_1} \cdot f_1(u_1) & \cdots & \sqrt{w_1} \cdot f_n(u_1) \\ \vdots & \ddots & \vdots \\ \sqrt{w_m} \cdot f_1(u_m) & \cdots & \sqrt{w_m} \cdot f_n(u_m) \end{pmatrix} \in \mathbb{R}^{m \times n} \qquad \text{und} \qquad b := \begin{pmatrix} \sqrt{w_1} \cdot v_1 \\ \vdots \\ \sqrt{w_m} \cdot v_m \end{pmatrix} \in \mathbb{R}^m$$

gilt: $E(x) = \|b - A \cdot x\|_2^2$.

Wie kann man nun f berechnen?

Aufgabe 4 (Bildkompression mit SVD, *3+1=4 Punkte*)

In dieser Aufgabe soll die Singulärwertzerlegung verwendet werden, um Bilder zu komprimieren. Wir fassen ein quadratisches Schwarzweißbild mit $m \cdot m$ Pixeln als Matrix $A \in \mathbb{R}^{m \times m}$ auf.

a) Implementiere die Funktionen `Compress` und `Decompress` im gegebenen Framework. Dabei bekommt `Compress` ein Bild A und die Anzahl p der zu behaltenden Singulärwerte übergeben und `Decompress` bekommt die Matrizen der SVD übergeben.
Die Kompression erfolgt dadurch, dass die Singulärwertzerlegung von A berechnet wird und, in Abhängigkeit von p, Singulärwerte und Spalten von U und V verworfen werden. Als komprimierte Darstellung werden dann die übriggebliebenen Teile von U und V sowie die übriggebliebenen Singulärwerte zurückgegeben. Außerdem soll `Compress` das Kompressionsverhältnis zurückgeben, also den Quotient des Speicherbedarfs vor und nach der Kompression.

   Hinweis: Zur Berechnung der Singulärwertzerlegung darf hier die Funktion `numpy.linalg.svd` benutzt werden.

b) Das Framework testet die Methode anhand dreier Bilder jeweils mit $1, 4, 8, 32$ und 64 beibehaltenen Singulärwerten. Wie kommen die unterschiedlichen Ergebnisse zustande? Erläutere insbesondere weshalb "Stoff" besser komprimiert wird als "Stoff2".

   Hinweis: Überlege dir zunächst, wie das Bild für $p = 1$ in `Decompress` zustandekommt und dann was sich bei größerem p ändert.

Bonusaufgabe 5 (Export und Import von komprimierten Bildern, *2 Punkte*)

Schreiben Sie Funktionen um Bilder, die mit der SVD komprimiert wurden, im NPZ-Format zu laden und zu speichern. Neben dem Testcode für die vorherige Aufgabe können die folgenden Funktionen hilfreich sein:

- `numpy.savez`
- `numpy.load`
- `matplotlib.image.imsave`

Laden Sie die Datei `Mystery.npz`, welche die Arrays `U, S, V` enthält, und benutzen Sie ihren Code um herauszufinden, was für ein Bild darin enthalten ist.

Hinweis: Verwenden Sie `cmap=cm.gray` damit matplotlib keine Colormap dir zur Datenvisualisierung gedacht ist verwendet.