**Methodische und Praktische Grundlagen der Informatik 4**

# Übungsblatt 4

mpgi4@cg.tu-berlin.de

WiSe 2013/2014

## Aufgabe 1: Cholesky Zerlegung

Nachdem wir die LR-Zerlegung kennen gelernt haben, kommen wir nun zur Cholesky Zerlegung. Genau wie die LR-Zerlegung wird dabei eine Matrix $A$ in das Produkt zweier Matrizen zerlegt, einer unteren und einer oberen Dreiecksmatrix. Die Matrizen der Cholesky Zerlegung sind allerdings Transponierte von einander, d.h. die obere Dreiecksmatrix geht aus der unteren Dreiecksmatrix durch das vertauschen von Zeilen und Spalten hervor und umgekehrt. Wir haben also

$$A = LL^T = \begin{pmatrix} l_{1,1} & 0 & 0 & \cdots & 0 \\ l_{2,1} & l_{2,2} & l_{3,2} & \cdots & 0 \\ l_{3,1} & l_{3,2} & l_{3,3} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ l_{n,1} & l_{n,2} & l_{n,3} & \cdots & l_{n,n} \end{pmatrix} \begin{pmatrix} l_{1,1} & l_{2,1} & l_{3,1} & \cdots & l_{n,1} \\ 0 & l_{2,2} & l_{3,2} & \cdots & l_{n,2} \\ 0 & 0 & l_{3,3} & \cdots & l_{n,3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & l_{n,n} \end{pmatrix} \tag{1}$$

Diese Zerlegung ist immer möglich wenn die Matrix $A \in \mathbb{R}^{n \times n}$

a) symmetrisch ist, d.h. $A = A^T$.

b) positiv semidefinit ist, d.h. $x^T Ax \geq 0$ für alle $x \in \mathbb{R}^n$.

In der Vorlesung wurde besprochen, dass die Elemente in der unteren Dreicksmatrix $L$ gegeben sind durch

$$l_{i,i} = \sqrt{a_{i,i} - \sum_{k=1}^{i-1} l_{i,k}^2} \tag{2}$$

$$l_{i,j} = \frac{1}{l_{j,j}} \left( a_{i,j} - \sum_{k=1}^{j-1} l_{i,k} l_{j,k} \right) \tag{3}$$

Schaut man sich diese Formeln an, stellt man fest, dass ein Element $l_{i,j}$ der Matrix nur von Elementen „über“ oder „links“ von dem Element abhängt. Mit anderen Worten: um $l_{i,j}$ zu berechnen, reicht es aus die Elemente $l_{m,n}$ mit $m \leq i$ und $n \leq j$ zu kennen. Wir können also von oben beginnend Zeile für Zeile von links nach rechts die Elemente $l_{i,j}$ berechnen.

Folgender Pythoncode implementiert die Cholesky Zerlegung:

Erarbeitet mit den Studenten den python Code am Beamer. Geht insbesondere auf die slicing Operationen ein und dass sie nicht nur *syntaktischer Zucker* sind sondern numpy erlauben optimierte Operationen durchzuführen. Testet die Zerlegung indem ihr eine zufällige Matrix `m = np.random.rand(4,4)` berechnet und $m^T m$ zerlegt.
An welcher Stelle kann der Algortihmus scheitern, sofern die Bedingungen nicht numerisch erfüllt sind?
Antwort: Wurzelziehen bei Diagonalelementen kann scheitern.

```python
import numpy as np
from math import sqrt

def cholesky(m):
    N = m.shape[0];
    assert(N == m.shape[1])
    L = np.zeros((N,N))

    for i in range(N):
        for j in range(0, i):
            L[i, j] = (m[i, j] - np.dot(L[i, :j], L[j, :j])) / L[j, j]
        L[i, i] = sqrt(m[i, i] - np.sum(np.square(L[i, :i])))

    return L
```

Die Cholesky-Faktorisierung bietet den Vorteil, dass sie nur ca. halb soviele Berechungen benötigt, wie die LR-Zerlegung. Darüber hinaus ist sie stabil, auch ohne Pivoting.

## Aufgabe 2: Lineare Regression

Angenommen wir haben eine Menge von Paaren gemessener Daten $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$ und wir vermuten, dass es einen linearen Zusammenhang zwischen $x_i$ und $y_i$ gibt. D.h. wir suchen eine Funktion $f(x) = y = a + bx$, die für alle Datenpaare möglichst „gut“ erfüllt ist, es soll also gelten $f(x_i) \approx y_i$.

Problem an die Tafel malen: Messwerte, welche ungefähr linear sind. Diskussion warum wir nie eine perfekte Gerade messen werden, insbesondere Messfehler und idealisierte Modelle.

Die Terme $a + bx_i - y_i$ lassen sich in Matrixschreibweise ausdrücken. Mit

$$Y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \text{ und } X = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \\ 1 & x_n \end{pmatrix} \text{ gilt:} \tag{4}$$

$$X \begin{pmatrix} a \\ b \end{pmatrix} - Y. \tag{5}$$

Dies ist ein überbestimmtes Gleichungssystem und wir werden keine Zahlen $a, b$ finden, sodass das System exakt gelöst wird. Wir können aber eine Lösung $\tilde{a}, \tilde{b}$ finden, sodass der Abstand des Vektors $X \begin{pmatrix} \tilde{a} \\ \tilde{b} \end{pmatrix}$ zu $Y$ minimal ist. In der Vorlesung haben wir gesehen, dass diese Bedingung genau dann erfüllt ist, wenn

$$X^T X \begin{pmatrix} a \\ b \end{pmatrix} = X^T b \tag{6}$$

gilt.

Zeigt am Computer wie man so ein Problem praktisch löst. Erklärt, dass man das Problem tatsächlich mit Cholesky lösen kann (Bedingungen sind erfüllt). Macht nochmal deutlich, wie Vorwärts- und Rückwärtseinsetzen funktioniert ohne den Code dafür zu zeigen (ist Hausaufgabe). Verwendet die Datei `Tut.py` als Vorlage.

Die verwendeten Datenpunkte entsprechen dem Ergebnis von Hausaufgabenschein und Endklausur aller Studenten einer nicht näher benannten TU-Veranstaltung. Welchen Schluss kann man aus den Daten ziehen?