Computerphysik WS 2013/2014

Prof. Dr. Roland Netz, FU Berlin

Übungsblatt 6: Ausgleichsrechnung

Suliman Adam, Klaus Rinne

25. November 2013

Allgemeine Hinweise

Abgabetermin für die Lösungen ist

- **Sonntag, 01.12., 24:00 Uhr.**

6.1 Lineare Ausgleichsrechnung (6 Punkte)

In den vergangenen 40 Jahren sind die Dönerpreise stetig gestiegen. Diverse Verbraucherschutzorganisationen haben in unregelmäßigen Abständen die Berliner Durchschnitspreise festgehalten.

Lösen Sie für den gegebenen Datensatz das lineare Ausgleichsproblem, indem Sie ein Python-Programm schreiben, welches das Normalengleichungssystem

$$\mathbf{A}^T\mathbf{A}\mathbf{a} = \mathbf{A}^T\mathbf{y} \tag{1}$$

löst.

Nutzen Sie die Ansatzfunktionen

(i) $f(x) = a_2(\frac{x-1970}{100})^2 + a_1(\frac{x-1970}{100}) + a_0$

(ii) $f(x) = a_3(\frac{x-1970}{100})^3 + a_1(\frac{x-1970}{100}) + a_0 cos(\frac{x-1970}{100})$

und lesen Sie die Preisentwicklung aus der Datei *data.txt* ein.

a) Wie lauten die optimalen Parameter für die Ausgleichsfunktionen?

b) Geben Sie für die jeweilige Ausgleichsfunktion die Summe der Fehlerquadrate an. An welcher Stelle tritt jeweils der größte Fehler auf?

c) Plotten Sie die Ausgangsdaten gemeinsam mit den optimalen Ausgleichsfunktionen in einem Graphen. Welche Preisvorhersage treffen Ihre Ausgleichsfunktionen für das Jahr 2020?

Aufgabe 6.2 Nichtlineare Ausgleichsrechnung (14 Punkte)

Für Ansatzfunktionen $f_i(a_1, ..., a_m, x_i)$, die nicht-linear in den Parametern a_i sind, lässt sich eine Ausgleichsrechnung von N Wertepaaren (x_i, y_i) nach dem Gauss-Newton-Verfahren durchführen. Hierbei werden die a_i so bestimmt, dass das Fehlerfunktional $E(\mathbf{a}) = \|g(\mathbf{a})\|_2^2$ mit

$$g_i = y_i - f_i(a_1, ..., a_m, x_i) \tag{2}$$

minimal wird, das heißt

$$\sum_i^N (f_i(a_1, ..., a_m, x_i) - y_i) \frac{\partial f_i(a_1, ..., a_m, x_i)}{\partial a_i} = 0 \tag{3}$$

für alle i. Durch Linearisierung der Fehlergleichung (2) führt man das nichtlineare Ausgleichsproblem auf eine Folge von linearen Ausgleichsproblemen zurück:

$$min \|g(\mathbf{a}^n) - J(\mathbf{a}^n)\delta^n\|_2^2$$

mit der Jacobi-Matrix

$$J(x) = \begin{pmatrix} \frac{\partial f_1}{\partial a_1}(x) & \dots & \frac{\partial f_1}{\partial a_m}(x) \\ . & \dots & . \\ \frac{\partial f_N}{\partial a_1}(x) & \dots & \frac{\partial f_N}{\partial a_m}(x) \end{pmatrix}$$

Der Vektor $\mathbf{a}^{n+1}$ in der nächsten Iteration ergibt sich nach $\mathbf{a}^{n+1} = \mathbf{a}^n + \delta^n$.

6.2.1 (6 Punkte)

Schreiben Sie ein Python-Programm, welches das Gauss-Newton-Verfahren implementiert.

6.2.2 (3 Punkte)

Erweitern Sie Ihr Programm so, dass die Anpassung der Parameter $\mathbf{a}$ gedämpft erfolgen kann. Dazu bestimmen Sie das kleinste k mit $k = 0, 1, 2, ...$ für das mit $\mathbf{a}^{n+1} = \mathbf{a}^n + 2^{-k}\delta^n$ der Fehler $\|g(a^{n+1})\|_2^2$ im nächsten Iterationsschritt $n+1$ kleiner ist als der Fehler $\|g(a^n)\|_2^2$ im aktuellen Schritt n. Das Abbruchkriterium für die Iteration mit Dämpfung soll dann $2^{-k} \cdot \|\delta\| < 10^{-6}$ sein.

Lassen Sie 2^{-k} nicht kleiner als 10^{-6} werden.

6.2.3 (3 Punkte)

Lesen Sie die Messdaten $x(t)$ des gedämpften harmonischen Oszillators aus der Datei *data2.txt* ein und ermitteln Sie die Ausgleichsfunktion mit den Ansatzfunktionen

a) $f(t) = \exp(-a_0 t)[a_2 \sin(a_1 t) + a_3 \cos(a_1 t)]$

b) $g(t) = \exp(-a_0 t) a_2 \sin(a_1 t)$

und den Startparametern

a1) $\vec{a} = [0.8, 6.4, 4.2, -0.3]$,

a2) $\vec{a} = [0.3, 5.4, 7.2, -1.3]$,

a3) $\vec{a} = [1.0, 7.0, -6.0, 3.0]$,

b1) $\vec{a} = [0.8, 6.4, 4.2]$,

b2) $\vec{a} = [0.3, 5.4, 7.2]$,

b3) $\vec{a} = [1.0, 7.0, -6.0]$.

Verwenden Sie dabei sowohl den gedämpften als auch den ungedämpften Algorithmus. Für welche Startwerte erhalten Sie eine sinnvolle Lösung?

6.2.4 (2 Punkte)

Verwenden Sie nun nur einen reduzierten Datensatz um für $f(t)$ und $\vec{a} = [0.8, 6.4, 4.2, -0.3]$ die Ausgleichsfunktion mit dem ungedämpften Algorithmus zu finden:

1) das erste Viertel der Messdaten

2) das letzte Viertel der Messdaten

3) jeden fünften Datenpunkt

4) jeden zwanzigsten Datenpunkt

5) jeden vierzigsten Datenpunkt

Plotten Sie die Ausgleichsfunktionen zusammen mit den Messdaten.