

Übungsblatt 12

Ausgabe: 11.1.2018

Abgabe: Donnerstag, 18.01.2018 bis 14:00 Uhr vor dem Tutorium

Aufgabe 33: (5 Punkte, 2+3)

Betrachten Sie das lineare Ausgleichsproblem: Zu $A \in \mathbb{R}^{m \times n}$ und $b \in \mathbb{R}^m$ finde $x \in \mathbb{R}^n$, so dass $\|Ax - b\|_2$ minimal ist.

a) Seien

$$A = \begin{pmatrix} 1 & -1 \\ 1 & 1 \\ 2 & 1 \end{pmatrix} \quad \text{und} \quad b = \begin{pmatrix} 2 \\ 4 \\ 8 \end{pmatrix}.$$

Stellen Sie die zugehörige Normalgleichung explizit auf und bestimmen Sie die Lösung $x \in \mathbb{R}^2$. Berechnen Sie das Residuum $r = Ax - b \in \mathbb{R}^m$ und den mittleren quadratischen Fehler

$$e := \sqrt{\frac{1}{m} \sum_{i=1}^m r_i^2}, \quad r := (r_1, \dots, r_m)^T \in \mathbb{R}^m.$$

b) Das lineare Ausgleichsproblem aus a) werde um die Gleichungen

$$x_1 + 2x_2 = 4$$

$$2x_1 - x_2 = 5$$

$$x_1 - 2x_2 = 2$$

erweitert. Geben Sie die zugehörige neue Matrix \tilde{A} und den Vektor \tilde{b} an. Stellen Sie die Normalgleichung explizit auf und bestimmen Sie die Lösung $\tilde{x} \in \mathbb{R}^2$. Berechnen Sie das Residuum und den mittleren quadratischen Fehler wie in a).

Aufgabe 34: (3 Punkte)

Gegeben seien Daten $(t_i, y_i) \in \mathbb{R}^2$ für $i = 1, \dots, m$ mit $t_i \neq t_j$ bei $i \neq j$. Gesucht sind jeweils

(1) eine lineare Funktion $y_1(t) = \alpha_1 t + \alpha_2$,

(2) eine quadratische Funktion $y_2(t) = \beta_1 t^2 + \beta_2 t + \beta_3$

(3) und eine kubische Funktion $y_3(t) = \gamma_1 t^3 + \gamma_2 t^2 + \gamma_3 t + \gamma_4$,

welche die Daten möglichst gut approximiert.

Formulieren Sie die Aufgaben (1) - (3) als lineare Ausgleichsprobleme und leiten Sie die Normalgleichungen her.

Aufgabe 35: (12 Punkte)

Im Folgenden soll in Matlab mit Hilfe der Dax-Daten ein Trend bestimmt werden. Laden Sie dazu auf der Veranstaltungshomepage die Excel Datei Dax.csv herunter. Öffnen Sie die csv-Datei über `fopen` mit Leserechten und importieren Sie die Daten über den `textscan`-Befehl. Als `formatSpec` tragen Sie `'%s %*f %*f %*f %f %*f %*f'` und als `delimiter` ein Komma (','') ein.¹ Achten Sie darauf, die Überschrift nicht einzulesen. Schließen Sie anschließend die Datei wieder über `fclose`.

Haben wir die Datei erfolgreich eingelesen, so ist in unserem Workspace nun ein neues 1×2 -Cell-Array gespeichert, über das wir die Handelstage sowie die DAX-Schlusskurse in einem chronologisch aufsteigenden Format einsehen können. Leider liegen die Daten der Handelstage in einem String-Array

¹Durch diese Kombination wird bewirkt, dass beim Einlesen der Daten jedes gelesene Komma als eine Trennung gehandhabt wird, weshalb beim eingetragenen `formatSpec` auch insgesamt sieben Formate integriert sind (für jede Spalte ein Format). Durch die `*`-Sterne wird Matlab kenntlich gemacht, dass die entsprechende Spalte nicht eingelesen werden soll. Somit lesen wir nur die erste Spalte mit den Handelstagen sowie die fünfte Spalte mit den Schlusskursen ein.

und somit in einem für unsere Zwecke schlecht zu handhabenden Format vor. Eine Möglichkeit, solche Daten handhabbarer zu machen, bietet die Funktion `datenum`, die jedem Datum im String-Format eine eindeutige Nummer (beginnend beim 0. Januar 0000) zuordnet:

```
DateNumbers = datenum(DateStrings);
```

Hierbei werden in einem Vektor `DateNumbers` die Integerwerte für die Daten gespeichert. Erstellen Sie nun mit Hilfe des Vektors `DateNumbers` einen Vektor `NumbersNonUni`, der die Tage wie folgt durch nummeriert:

$$\text{DateNumbers} = \begin{pmatrix} '2015-01-02' \\ '2015-01-05' \\ '2015-01-06' \\ '2015-01-07' \\ \vdots \\ '2018-01-05' \end{pmatrix} \quad \text{NumbersNonUni} = \begin{pmatrix} 1 \\ 4 \\ 5 \\ 6 \\ \vdots \\ 1100 \end{pmatrix}.$$

Beide Vektoren sind dann 764×1 groß. Da an Samstagen und Sonntagen an der Börse nicht gehandelt wird, stehen zum Beispiel für den `'2015-01-03'` und `'2015-01-04'` keine Daten zur Verfügung. Um Daten auf einem uniformen Gitter zu erhalten, sollen die fehlenden Daten linear interpoliert werden. Erstellen sie hierzu einen Vektor `NumbersUni`, für den `NumbersUni(i+1)-NumbersUni(i)=1` gilt. Verwenden Sie dann die Funktion `[S] = LinearInterpol(Tk, Sk, T)` von der Veranstaltungshomepage, um die Dax-Daten für Samstag, Sonntag bzw. Feiertage linear zu interpolieren. Benennen Sie den resultierenden Vektor mit `DaxUni`. Zur Bestimmung eines Trends soll wie in Aufgabe 34 ein lineares Ausgleichsproblem zu den Daten `(NumbersUni(i), DaxUni(i))` formuliert werden. Implementieren Sie hierzu die zugehörigen Normalgleichungen, wenn eine lineare, quadratische und eine kubische Funktion, die die Dax-Daten möglichst gut approximieren sollen, verwendet wird. Lösen Sie die Normalgleichungen und plotten Sie die Ergebnisse zusammen mit den Dax-Daten in einen Plot. Achten Sie auf eine aussagekräftige Beschriftung.

Zur Abgabe: Geben Sie einen Ausdruck des Programmdurchlaufs und des vorbildlich kommentierten m-files im Tutorium (Zentralübung) ab. Schicken Sie Ihren Programmcode zusätzlich per Email an Ihren jeweiligen Übungsgruppenleiter und zwar mit Subject/Betreff à la:

Subject: Uebung2, Muster, Hans

Bitte erstellen Sie hierzu einen ZIP komprimierten Ordner mit den m-files. Bitte benennen Sie die m-files zum Beispiel mit `aufgabe35.m` und schreiben Sie in jedem m-file ihren Namen. Die Email-Adressen der jeweiligen Übungsgruppenleiter finden Sie auf der Veranstaltungshomepage.