Technische Universität Chemnitz
Prof. Dr. R. Herzog, M. Winkler

Chemnitz, 27. Oktober 2017
Abgabe am 1. November 2017

Grundlagen der Optimierung

Übung 3

Aufgabe 11: Gradientenverfahren mit exakter Liniensuche am Beispiel

Wir betrachten das Problem

$$\text{Minimiere} \quad x^2 + (y-3)^2.$$

Bestimmen Sie die ersten zwei Iterierten des Gradientenverfahrens mit Startpunkt $(x_0, y_0) = (0, 0)$ und exakter Liniensuche.

Aufgabe 12: Lineare Regression

Gegeben seien die Messwerte $(x_i, y_i), i = 1, \ldots, m$. Ein zugrunde liegendes physikalisches Modell sagt den linearen Zusammenhang

$$y(x) = ax + b$$

voraus. Anhand der Messwerte sollen die Parameter a und b bestimmt werden, sodass die Summe der Fehlerquadrate minimiert wird, d.h. es ist das unbeschränkte Optimierungsproblem

$$\min_{(a,b)\in\mathbb{R}^2} \quad f(a,b) = \sum_{i=1}^{m} (y(x_i) - y_i)^2 = \sum_{i=1}^{m} (ax_i + b - y_i)^2$$

zu lösen. Wie lauten die notwendigen Optimalitätsbedingungen erster Ordnung?

Aufgabe 13: Ein Häufungspunkt des Gradientenverfahrens kann kein Maximum sein

Zeige: Bricht das Gradientenverfahren (Algorithmus 4.4) nicht nach endlich vielen Schritten ab (mit $\nabla f(x_k) = 0$) und ist x^* ein Häufungspunkt der durch dieses Verfahren konstruierten Folge $\{x_k\}$, so ist x^* kein lokales Maximum der stetigen Funktion f. Gilt diese Aussage auch, wenn der Algorithmus nach endlich vielen Schritten in einem Punkt x^* abbricht?

Aufgabe 14: Exakte Liniensuche für eine quadratische Funktion

Gegeben sei eine symmetrischen, positiv definite Matrix $Q \in \mathbb{R}^{n\times n}$, ein Vektor $c \in \mathbb{R}^n$ und ein $\gamma \in \mathbb{R}$. Wir wollen die Optimierungsaufgabe

$$f(x) = \frac{1}{2} x^\top Q\, x + c^\top x + \gamma \to \min!$$

mit dem Gradientenverfahren lösen.

(a) Berechne den Gradienten und die Hessematrix der Funktion f.

(b) Es sei $d \in \mathbb{R}^n$ eine Abstiegsrichtung. Die Vorschrift der exakten Liniensuche lautet:

$$\text{„Bestimme } t_{\min} \text{ so, dass } f(x + t_{\min} d) = \min_{t \geq 0} f(x + t\, d) \text{ gilt.“}$$

Berechne $t_{\min}$ für die Funktion f.

Hausaufgabe 7: Implementierung der Armijo-Schrittweitenstrategie

Implementiere die Armijo-Schrittweitenstrategie aus der Vorlesung in Matlab. Erstelle dazu eine Datei `ls_armijo.m` und verwende

```
function t = ls_armijo(fhandle, x, d, s, sigma, beta)
```

als erste Zeile.
Dabei bezeichnet `fhandle` das Handle auf eine Funktion (`help function_handle` und `doc function_handle`; Beispiel: `@rosenbrock`), die den Funktionswert und ggf. den Gradienten zurückgibt, `x` den Ausgangspunkt, `d` eine Abstiegsrichtung, `s` die Startschrittweite und `sigma` und `beta` die in der Vorlesung eingeführten Parameter des Algorithmus. Beachte, dass `d` und `x` als Spaltenvektoren übergeben werden.

Zurückgegeben werden soll ein Zeilenvektor `t`$= (t_1, \ldots, t_m)$, der alle Iterierten[1] der Backtracking-Suche zur Bestimmung der Armijo-Schrittweite t_m beinhaltet.

Ein fertiges Testprogramm ist im Skript `solution_ha_7.m` bereits implementiert. Dort ist lediglich der Parameter `Exercise` zu setzen. Das Skript erfordert nach jeder Iteration eine Tastatureingabe.

Teste die Funktion an folgenden Beispielen und Eingabewerten:

(a) An der Rosenbrock-Funktion

$$f(x, y) = (1 - x)^2 + 100(y - x^2)^2$$

mit $x = (1.7;\ 1.5)$, $d = (-1;\ 0)$, `s`$= 4$, `sigma`$= 0.1$ und `beta`$= 0.5$.

(b) An der Rosenbrock-Funktion mit `x`$= [0; 0]$, `d`$= [1; 0]$, `s`$= 1$, `sigma`$=0.1$ und `beta`$= 0.5$.

[1] wird vom Skript `solution_ha_7.m` zum Plotten verwendet

(c) An

$$\varphi(x) := \omega(c_1)\sqrt{(1-x)^2 + c_2^2} + \omega(c_2)\sqrt{x^2 + c_1^2}$$

mit $\omega(c) = \sqrt{1+c^2} - c$, $c_1 = 0.01$, $c_2 = 0.001$, `x`$= 0$, `d`$= 1$, `s`$= 1$, `beta`$= 0.5$. Variiere `sigma` zwischen 0.01 und 0.5. Was stellt man fest?

Hinweis 1: Verwende die Dateien `rosenbrock.m` und `ls_testfun.m` von der Webseite der Lehrveranstaltung. Dort sind die Funktionen aus Aufgabe (a) und (c) bereits implementiert.
Hinweis 2: Der Quelltext ist angemessen zu kommentieren.

Hausaufgabe 8: Implementierung des Gradientenverfahrens

Implementiere das Gradientenverfahren mit allgemeinen Skalarprodukt und Armijo-Schrittweitenstrategie in Matlab (Algorithmus 4.4 aus der Vorlesung). Verwende das Abbruchkriterium aus Bemerkung 4.7 (a) und (b).

Erstelle dazu eine Datei `steepest_descent_armijo.m` und verwende

```
function X = steepest_descent_armijo(fh, M, x0, tol, s, sigma, beta)
```

als erste Zeile. Dabei bezeichnet `fh` das Handle auf die Zielfunktion, `M` die Skalarprodukt induzierende Matrix, `x0` den Startpunkt, `tol` eine Struktur, welche die vier Parameter `ATOL_f`, `RTOL_f`, `ATOL_x` und `RTOL_x` für das Abbruchkriterium aus Bemerkung 4.7 (a) und (b) enthält, und `s`, `sigma` und `beta` die Parameter für die Armijo-Schrittweitensuche. Zurückgegeben werden soll eine Matrix `X=[x0, x1, x2, ...]`, welche den gesamten Iterationsverlauf enthält.

Teste das implementierte Verfahren an folgenden Funktionen und Eingabewerten und visualisiere den Iterationsverlauf im eindimensionalen Fall bzw. zweidimensionalen Fall in einem `plot` bzw. `contour-plot` mit Hilfe der zurückgegebenen Matrix X. Sofern nichts anderes angegeben, verwende dabei `M`$= I$, `s`$= 1$, `sigma`$= 1e-2$, `beta`$= 0.5$, sowie `1e-2` für die vier Toleranzen. Interpretiere dabei auch die Anzahl der benötigten Iterationen.

(a) An der Cosinus-Funktion mit `x0`$= 1.1656$

(b) An der Himmelblau-Funktion

$$f(x_1, x_2) = \left(x_1^2 + x_2 - 11\right)^2 + \left(x_1 + x_2^2 - 7\right)^2$$

mit den Startwerten `x0`$= [-0.27; -0.91]$, `x0`$= [-0.271; -0.91]$, `x0`$= [-0.25; -1.1]$ und `x0`$= [-0.25; -1]$

(c) An der Funktion $f\colon \mathbb{R}^2 \to \mathbb{R}$,

$$f(x) := \frac{1}{2}x^\top Qx + b^\top + d$$

mit

$$Q = \begin{pmatrix} 5 & -3 \\ -3 & 4 \end{pmatrix}, \qquad b = (3 \quad 1)^\top, \qquad d = 2,$$

und `x0`$= [4; 1]$

(d) An der Rosenbrock-Funktion

$$f(x, y) = (1 - x)^2 + 100(y - x^2)^2$$

mit `x0`$= [1; -0.5]$ und `1e-6` für die vier Toleranzen.

Als Grundlage dient das Skript `solution_ha_8.m`, welches auf der Homepage der Lehrveranstaltung zu finden ist. Ferner sind dort bereits die Beispielfunktionen in `cosine.m`, `himmelblau.m`, `rosenbrock.m` und `general_quadratic_function.m` implementiert.

Experimentiere in Teil (d) mit der Matrix M. Finde eine Matrix M, sodass in Teil (d) weniger Iterationen benötigt werden.

Schicke die erzeugten Dateien `steepest_descent_armijo.m` und `ls_armijo.m` an max.winkler@mathematik.tu-chemnitz.de (Betreff: HA Grundlagen der Optimierung Übung 3)!

Hausaufgabe 9: Häufungspunkte des Gradientenverfahrens haben den gleichen Funktionswert

Zeige: Seien $f : \mathbb{R}^n \to \mathbb{R}$ stetig und $\{x_k\} \subset \mathbb{R}^n$ eine durch das Gradientenverfahren (Algorithmus 4.4) erzeugte Folge. Sind x^* und x^{**} zwei Häufungspunkte der Folge $\{x_k\}$, so gilt $f(x^*) = f(x^{**})$.