

11 Nichtlineare Optimierung

11.1 Mathematische Grundlagen

Gegeben sei eine *multivariate reellwertige Funktion*

$$f : \mathbb{R}^n \rightarrow \mathbb{R}, x \mapsto f(x) = f(x_1, \dots, x_n) . \quad (1)$$

Der *Gradient* einer multivariaten reellwertigen Funktion ist definiert als

$$\nabla f(x) := \begin{pmatrix} \frac{\partial}{\partial x_1} f(x) \\ \vdots \\ \frac{\partial}{\partial x_n} f(x) \end{pmatrix} \in \mathbb{R}^n \quad (2)$$

und die *Hesse-Matrix* einer multivariaten reellwertigen Funktion ist definiert als

$$\nabla^2 f(x) := \begin{pmatrix} \frac{\partial^2}{\partial x_1 \partial x_1} f(x) & \cdots & \frac{\partial^2}{\partial x_1 \partial x_n} f(x) \\ \vdots & \ddots & \vdots \\ \frac{\partial^2}{\partial x_n \partial x_1} f(x) & \cdots & \frac{\partial^2}{\partial x_n \partial x_n} f(x) \end{pmatrix} \in \mathbb{R}^{n \times n} , \quad (3)$$

wobei $\frac{\partial}{\partial x_i} f(x)$ die (*erste*) *partielle Ableitung* von $f(x)$ nach x_i und $\frac{\partial^2}{\partial x_i \partial x_j} f(x)$ die *zweite partielle Ableitung* von $f(x)$ nach x_i und x_j ist.

Die multivariate Differentialrechnung bietet mit dem *Gradientenverfahren* eine Methode, die Minimalstellen (oder, durch Umkehrung des Vorzeichens, Maximalstellen) einer multivariaten reellwertigen Funktion zu bestimmen. In der Praxis einer faktisch zu minimierenden Funktion wird das Gradientenverfahren wie folgt durchgeführt:

- Man legt einen *Startpunkt* $x_0 \in \mathbb{R}^n$ fest, von dem man glaubt, dass er sich in der Nähe einer lokalen Minimalstelle oder einer globalen Minimalstelle von f befindet.
- Mit steigendem Index $k = 1, 2, 3, \dots$ berechnet man nun iterativ Funktionargumente auf Grundlage der Gleichung

$$x_{k+1} = x_k - \alpha \nabla f(x_k) , \quad (4)$$

wobei α die *Lernrate* des Gradientenverfahrens ist.

- Dieser iterative Schritt wird solange wiederholt, bis sich der Gradient der Funktion nicht mehr nennenswert ändert

$$\|\nabla f(x_{k+1})\| < \delta , \quad (5)$$

wobei δ das *Konvergenzkriterium* des Gradientenverfahrens ist.

Unter bestimmten Bedingungen konvergieren die mittels der obigen Gleichung berechneten Werte x_k dann gegen eine lokale Minimalstelle oder eine globale Minimalstelle von f .

11.2 Analyse in R

Wir beginnen zunächst mit der Definition einer bivariaten reellwertigen Funktion. Erklären Sie die Funktion und Details des folgenden R-Codes:

```
# Funktion
f = function(x) {
  f_x = x[1]^2 + x[2]^2          # f(x) = x_1^2 + x_2^2
  return(f_x)
}

# Gradient
nabla_f = function(x) {
  nabla_f_x = matrix(c(2*x[1],    # \nabla f(x) = (2 x_1, 2 x_2)^T
                      2*x[2]), nrow = 2)
  return(nabla_f_x)
}

# Hesse-Matrix
H_f = function(x) {
  H_f_x = matrix(c(2, 0,          # H_f(x) = (2 0)
                  0, 2), nrow = 2) #           (0 2)
  return(H_f_x)
}
```

11.3 Erste Programmieraufgabe

Berechnen Sie den Gradienten der Funktion f an denjenigen Stellen in ihrer Definitionsmenge \mathbb{R}^2 , die in der Vorlesung illustriert wurden. Gehen Sie dazu wie folgt vor:

- Definieren Sie eine 2×4 Matrix `x12`, bei der jede Spalte einem der folgenden zweidimensionalen Vektoren entspricht:

$$x = \begin{pmatrix} 0.7 \\ 0.7 \end{pmatrix} \quad x = \begin{pmatrix} -0.3 \\ 0.1 \end{pmatrix} \quad x = \begin{pmatrix} -0.5 \\ -0.4 \end{pmatrix} \quad x = \begin{pmatrix} 0.1 \\ -1.0 \end{pmatrix}$$

- Initialisieren Sie eine 2×4 Nullmatrix `nabla_f_x12`, in die im Folgenden die Gradientenwerte eingefüllt werden.
- Berechnen Sie mithilfe der oben definierten Funktion und einer `for`-Schleife über die obigen Vektoren die i -te Spalte von `nabla_f_x12` als den Gradienten der i -ten Spalte von `x12`.
- Definieren Sie mit `c()` einen Vektor, der die Werte “black”, “red”, “blue” und “lightgreen” enthält. Speichern Sie diesen Vektor als die Variable `cols`.

- Geben Sie die Werte des Gradienten aus. Sie sollten folgende Ergebnisse erhalten:

```
[,1] [,2] [,3] [,4]
[1,] 1.4 -0.6 -1.0 0.2
[2,] 1.4 0.2 -0.8 -2.0
```

11.4 Abbildung in R

Die soeben berechneten Werte des Gradienten sollen nun durch Visualisierung überprüft werden. Erklären Sie dazu den folgenden R-Code und die Abbildung, die er erzeugt:

```
# Definitionsmengenraum
x_min = -2                                # x_i Minimum
x_max = 2                                  # x_i Maximum
x_res = 1e2                                # x_i Auflösung
x_1 = seq(x_min, x_max, length.out = x_res) # x_1 Raum
x_2 = seq(x_min, x_max, length.out = x_res) # x_2 Raum
X = expand.grid(x_1, x_2)                  # X = (x_1, x_2)^T Raum
fX = matrix(as.matrix(f(X)), nrow = x_res) # f(x)

# Abbildungsparameter
library(latex2exp)
par(
  family = "sans",
  mfcol = c(1,1),
  pty = "s",
  bty = "l",
  lwd = 1,
  las = 1,
  xaxs = "i",
  yaxs = "i",
  font.main = 1,
  cex = 1.2,
  cex.main = 1.2)

# Funktion
contour(x_1, x_2, fX,
  xlim = c(x_min, x_max),
  ylim = c(x_min, x_max),
  xlab = TeX("$x_1$"),
  ylab = TeX("$x_2$"),
  levels = c(0.1, 0.5, 1:8),
  main = "")

# Punkte und Pfeile
for(i in 1:ncol(x12)){
  points(x12[1,i], x12[2,i],
    col = cols[i])
  arrows(
    x0 = x12[1,i],
    y0 = x12[2,i],
    x1 = nabla_f_x12[1,i],
    y1 = nabla_f_x12[2,i],
    col = cols[i],
    angle = 30,
    length = 0.1,
```

```

    lwd = 2)
}
# Speichern
dev.copy2pdf(
  file = "Abbildungen/Nichtlineare_Optimierung_1.pdf",
  width = 5,
  height = 5)

```

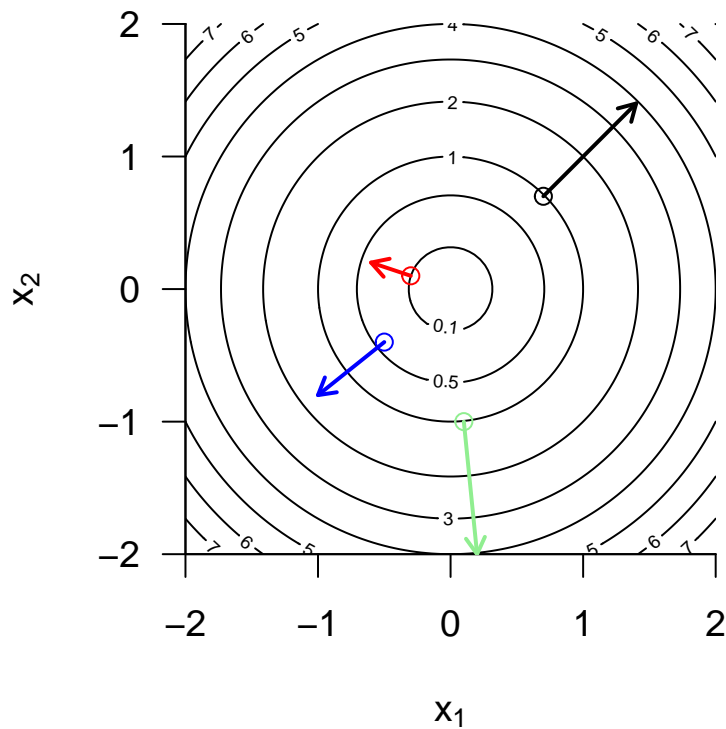


Abbildung 1. Beispielhafte Gradienten von $f(x) = x_1^2 + x_2^2$.

11.5 Zweite Programmieraufgabe

Führen Sie das Gradientenverfahren für f durch, d.h. die Suche nach dem Argument $x \in \mathbb{R}^2$, an der die Funktion den kleinsten Funktionswert annimmt. Gehen Sie dazu wie folgt vor:

- Legen Sie die Lernrate $\alpha = 0.1$ und das Konvergenzkriterium $\delta = 0.01$ fest.

- Initialisieren Sie \mathbf{x}_k mithilfe eines von Ihnen selbstgewählten Startwertes in der Menge $[-1, +1]^2$. Ein zulässiger Wert wäre beispielsweise $x_0 = (-0.93 \ 0.27)^T$.
- Legen Sie folgende Variablen an: \mathbf{x} , deren Wert dem von \mathbf{x}_k identisch sein soll; \mathbf{fx} , deren Wert dem Funktionswert an der Stelle \mathbf{x}_k entsprechen soll; und \mathbf{crt} , deren Wert die Länge des Gradienten an der Stelle \mathbf{x}_k sein soll. Die Länge eines Vektors kann in R mithilfe von `norm()` berechnet werden.
- Starten Sie eine `while`-Schleife, deren Fortführungsbedingung darin besteht, dass die Länge (`norm()`) des aktuellen Gradienten (`nabla_f(x_k)`) größer als das Konvergenzkriterium (`delta`) ist. Die Befehle innerhalb der Schleife werden solange ausgeführt, wie die Norm des Gradienten das Konvergenzkriterium nicht unterschreitet.
- Einerseits soll innerhalb dieser Schleife das nächste Funktionsargument berechnet werden. Ersetzen Sie die Variable \mathbf{x}_k durch den Wert von x_{k+1} gemäß der oben angegebenen Formel.
- Andererseits soll innerhalb der Schleife der Verlauf des Gradientenverfahrens dokumentiert werden. Erweitern Sie dazu die zuvor initialisierten Variablen wie folgt: \mathbf{x} um den neuen Wert \mathbf{x}_k mittels `cbind()`; \mathbf{fx} um den Funktionswert an der Stelle \mathbf{x}_k mittels `c()`; \mathbf{crt} um die Länge des Gradienten an der Stelle \mathbf{x}_k mittels `c()`.
- Führen Sie den folgenden R-Code aus:

```
# Definitionsmengenraum
x_min = -1                # x_i Minimum
x_max = 1                # x_i Maximum
x_res = 1e2              # x_i Auflösung
x_1 = seq(x_min, x_max, length.out = x_res) # x_1 Raum
x_2 = seq(x_min, x_max, length.out = x_res) # x_2 Raum
X = expand.grid(x_1, x_2) # X = (x_1, x_2)^T Raum
fX = matrix(as.matrix(f(X)), nrow = x_res ) # f(x)
cols = hcl.colors(10, "Yl0rRd") # Farben

# Abbildungsparameter
library(latex2exp)
par(
  family = "sans",
  mfcol = c(1,3),
  pty = "s",
  bty = "l",
  lwd = 1,
  las = 1,
  xaxs = "i",
  yaxs = "i",
  font.main = 1,
  cex = 1.2,
  cex.main = 1.2)

# Funktion
contour(x_1, x_2, fX,
  xlim = c(x_min, x_max),
  ylim = c(x_min, x_max),
  xlab = TeX("$x_1$"),
```

```

ylab      = TeX("$x_2$"),
nlevels   = 10,
col       = cols,
main      = "f(x)"

# Argumente
points(x[1,], x[2,],
       type = "o")

# Funktionswerte
plot(1:length(fx), fx,
     type = "l",
     lwd  = 1,
     xlab = "k",
     ylab = "",
     ylim = c(0,1),
     main = TeX("$f(x^k)$"))

# Kriterien
plot(1:length(crt), crt,
     type = "l",
     lwd  = 1,
     xlab = "k",
     ylab = "",
     ylim = c(0,2),
     main = TeX("$|| \nabla f(x^k) ||$"))

# Speichern
dev.copy2pdf(
  file = "Abbildungen/Nichtlineare_Optimierung_2.pdf",
  width = 12,
  height = 4)

```

- Sie sollten folgende Abbildung erhalten:

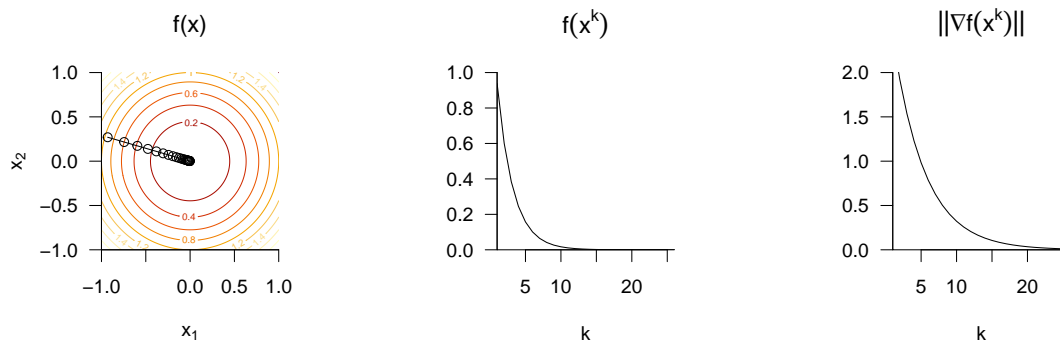


Abbildung 2. Gradientenverfahren für die Funktion $f(x) = x_1^2 + x_2^2$.

11.6 Lückentext

Füllen Sie mit den in der Übung gewonnenen Erkenntnissen den folgenden Lückentext aus und präsentieren Sie die Ergebnisse im Seminar:

Lückentext: Eine multivariate reellwertige Funktion hat die Definitionsmenge _____ und die Zielmenge _____. Der Gradient einer multivariaten reellwertigen Funktion f an der Stelle $x \in \mathbb{R}^n$ ist _____, die Hesse-Matrix einer multivariaten reellwertigen Funktion f an der Stelle $x \in \mathbb{R}^n$ ist _____. Die Lernrate eines Gradientenverfahrens bestimmt, _____; das Konvergenzkriterium eines Gradientenverfahrens bestimmt, _____. Die Durchführung des Gradientenverfahrens für die Funktion f erfordert _____ (die Funktion/den Gradienten), aber nicht _____ (den Gradienten/die Hesse-Matrix). In dem Fall, dass das Gradientenverfahren für die Funktion f konvergiert, ist das Ergebnis eine _____, im besten Fall ist es die _____.

11.7 Mögliche Klausurfrage

Präsentieren Sie im Seminar folgende Klausurfrage und erklären Sie die richtige Antwort:

Frage: Gegeben sei die Funktion $f(x_1, x_2) = 2x_1^2 + 3x_2^3$. Was ist der Gradient dieser Funktion?

a) $\nabla f(x) = \begin{pmatrix} x_1 \\ x_2^2 \end{pmatrix}$

b) $\nabla f(x) = \begin{pmatrix} 2x_1 \\ 2x_2 \end{pmatrix}$

c) $\nabla f(x) = \begin{pmatrix} 2x_1 \\ 3x_2^2 \end{pmatrix}$

d) $\nabla f(x) = \begin{pmatrix} 4x_1 \\ 9x_2^2 \end{pmatrix}$

11.8 Kinderwitz

Wie heißt das Haus, in dem die Katzen wohnen?

Antwort: Mietzhaus.