07 Newtonartige Verfahren, Quasi-Newton-Verfahren, nichtlineare CG-Verfahren

Nichtlineare Optimierung
WS 2020/21

Prüfung

- Die Prüfungen werden mündlich per BBB-Videokonferenz stattfinden.
- Prüfungszeitraum 05.02.2021 bis einschließlich 23.03.2021 Di Vormittag Fr Vormittag
- Prüfungsinhalte: Skript, Konsultation, Übungen
- siehe auch Ordner „Prüfung“ im OPAL

30 Minuten, keine Vorbereitungszeit

Newtonartige Verfahren

Newton-Verfahren:

$$\nabla^2 f(x_k)\, d_k = -\nabla f(x_k)$$

Newtonartige Verfahren:

$$H_k\, d_k = -\nabla f(x_k) + \zeta$$

inexaktes Newton-Verfahren:

$$\nabla^2 f(x_k)\, d_k = -\nabla f(x_k) + \zeta$$

Quasi-Newton-Verfahren:

$$H_k\, d_k = -\nabla f(x_k)$$

Quizfrage

Unter welcher Bedingung können wir die q-superlineare Konvergenz von Quasi-Newton-Verfahren erwarten?

→ Umfrage

B ⇒ D

- 0 A $H_k \to 0$ (Nullmatrix)
- 0 C $H_k \, d_k \to 0$
- 2 B $\|H_{k+1} - H_k\| \to 0$
- 7 D $\|(H_{k+1} - H_k)\, d_k\| = o(\|d_k\|)$

Satz 5.34

$m_k(x) = f(x_k) + \nabla f(x_k)^T(x - x_k) + \ldots H_k$

Quizfrage

$m_{k+1}(x) = f(x_{k+1}) + \nabla f(x_{k+1})^T(x - x_{k+1}) + \ldots H_{k+1}$

$\boxed{\nabla m_{k+1}(x_k) = \nabla f(x_k)} = \nabla m_k(x_k)$

Was ist die Sekantenbedingung?
Hinweis: $s_k = x_{k+1} - x_k$ und
$y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$

→ Umfrage

3 A $H_{k+1}\, s_k = y_k$

0 B $\nabla f(x + d) - \nabla f(x) = \nabla^2 f(x)\, d + \int_0^1 \left[\nabla^2 f(x + t\, d) - \nabla^2 f(x)\right] d \,\mathrm{d}t$

1 C $H_{k+1}\, y_k = s_k$

4 D $H_{k+1}\, \underbrace{(x_{k+1} - x_k)}_{s_k} = \underbrace{\nabla f(x_{k+1}) - \nabla f(x_k)}_{y_k}$

Konstruktionsideen

Welche Dinge spielen eine Rolle?

- Sekantenbedingung

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

- H_{k+1} „in der Nähe“ von H_k

$$\text{damit } \|(H_{k+1} - H_k)\, d_k\| = o(\|d_k\|)$$

- H_{k+1} spd

im Kontext von Liniensuchverfahren
(benötigen Abstiegsrichtungen)

Quizfrage

Es sei d_k eine Abstiegsrichtung für f an der Stelle x_k. Welche Rolle spielt die Krümmungsbedingung $f'(x_{k+1})\, d_k \geqslant \tau\, f'(x_k)\, d_k$ bei Quasi-Newton-Verfahren? (Wolfe-Liniensuche)

→ Umfrage

$$\left.\begin{aligned} s_k &= (x_{k+1} - x_k) \\ s_k &= \alpha_k d_k \end{aligned}\right\} \quad x_{k+1} = x_k + \alpha_k d_k$$

5 **A** sichert $s_k^\top H_{k+1}\, s_k > 0$

2 **C** sichert H_{k+1}^{BFGS} spd ← Lemma 5.36

1 **B** bringt guten Abstieg bei der Liniensuche

0 **D** keine besondere Rolle, man kann auch eine Armijo-Liniensuche verwenden

$$s_k^\top H_{k+1} s_k = s_k^\top y_k = (\nabla f(x_{k+1}) - \nabla f(x_k))^\top \alpha_k d_k > 0$$

$$\underline{f'(x_{k+1})\, d_k} \geqslant \tau\, \underbrace{f'(x_k)\, d_k}_{<0} > \underline{f'(x_k)\, d_k}$$

rank $(AB) \le \min\{\text{rank}(A), \text{rank}(B)\}$

Quizfrage

Wie wird die Sherman-Morrison-Woodbury-Formel

$$\underbrace{(A + U\,C\,V)^{-1}}_{\text{Aufdatierung}} = A^{-1} - A^{-1}\,U\,\underbrace{(C^{-1} + V\,A^{-1}\,U)^{-1}}_{\square}\,V\,A^{-1}$$

Rang ℓ

Rang $\le \ell \le n$

mit $A \in \mathbb{R}^{n\times n}$, $C \in \mathbb{R}^{\ell\times\ell}$, $U \in \mathbb{R}^{n\times\ell}$ und $V \in \mathbb{R}^{\ell\times n}$ typischerweise verwendet?

→ Umfrage

1 A $n = \ell$ und A^{-1} bekannt

0 B $n < \ell$ und A^{-1} bekannt

0 C $n = \ell$ und A^{-1} gesucht

8 D $\ell < n$ und A^{-1} bekannt

Quizfrage

Wozu wird die SMW-Formel im Kontext von Quasi-Newton-Verfahren verwendet?

$\rightarrow$ Umfrage

1 **A** zur Lösung von $H_k\, d_k = -\nabla f(x_k)$

0 **B** für den Konvergenzbeweis $x_k \to x^*$

8 **C** Herleiten einer Update-Formel für $H_k^{-1} \rightsquigarrow H_{k+1}^{-1}$

0 **D** für den Beweis der q-superlinearen Konvergenz

$A = H_k$

$A + UCV = H_{k+1}$

$\rightsquigarrow H_{k+1}^{-1} = H_k^{-1} - \dots$

low-rank Update (oft $\ell = 2$)

mit inverser Formel: $B_k = H_k^{-1}$

$H_k\, d_k = -\nabla f(x_k)$ LGS!

$\Longleftrightarrow \quad d_k = -B_k \nabla f(x_k)$ kein LGS!

Quizfrage

$\in \mathbb{R}^{n \times n}$

Was ist ein Nachteil der Verwendung von Quasi-Newton-Matrizen H_k bzw. der Inversen B_k?

→ Umfrage

typisch: H_0 evtl. dünnbesetzt
H_1 = Rang-2-Update von H_0
nicht mehr dünnbesetzt

0 A Berechnungsaufwand für H_k bzw. B_k (Aufdatierung)

7 B Speicherbedarf

1 C exakte Lösung der linearen Gleichungssysteme $H_k \, d_k = -\nabla f(x_k)$
(man kann das auch iterativ und inexakt machen oder gleich mit $d_k = -B_k \nabla f(x_k)$ arbeiten)

0 D fehlende Symmetrie

Quizfrage

Wie kann man den Speicherbedarf von Quasi-Newton-Matrizen sinnvoll reduzieren?

$\rightarrow$ Umfrage

keine superlineare K.

keine superlin. K. (Gradientenverfahren)

0 A Zurücksetzen von H_k auf H_0 alle $m_{\max}$ Iterationen

0 B durch Verwenden einer festen Matrix $H_k \equiv M$

1 C durch Speichern aller Vektorpaare jeweils $2n$ $(y_0, s_0), (y_1, s_1), \ldots$

7 D durch Speichern nur einiger Paare von Vektoren $\{(y_i, s_i)\}_{i=k-m,\ldots,k-1}$

z.B. inverses BFGS: $\quad B_0 g \quad , \quad B_1 g = V_0^T B_0 \underline{V_0 g} + \gamma_0 s_0 \underline{s_0^T g}$

$d_k = - B_k \nabla f(x_k)$

$= I - \gamma_0 y_0 s_0 \quad , \quad \gamma_0 = y_0^T s_0$

Auswertung von $B_k^{\overbrace{\text{LM}}^{}\text{-BFGS}} g$

limited-memory

Setze $r := g$
for $i := k-1, k-2, \ldots, k-m$ **do**
    Setze $\alpha_i := \gamma_i \, s_i^\mathsf{T} r$
    Setze $r := r - \alpha_i \, y_i$
end for // $r = V_{k-m} \, V_{k-m+1} \cdots V_{k-1} \, g$
seed matrix
Setze $d := B_0^{\text{BFGS}} \, r$
for $i := k-m, \ldots, k-2, k-1$ **do**
    Setze $\beta := \gamma_i \, y_i^\mathsf{T} d$
    Setze $d := d + (\alpha_i - \beta) \, s_i$
end for
return d // $d = B_k^{\text{LM-BFGS}} \, g$

$B_0^{\text{BFGS}} \cdot \text{Vektor} \mathrel{\hat{=}} M^{-1} \cdot \text{Vektor}$

$H_0^{\text{BFGS}} \mathrel{\hat{=}} M$ (Basismetrik im Raum der Optimierungsvariablen)

Nichtlineare CG-Verfahren

mit A spd

Minimiere $\frac{1}{2}x^\mathsf{T} A\, x - b^\mathsf{T} x$

- $r_k = Ax_k - b$
- $d_0 = -M^{-1} r_0$
- $d_k = -M^{-1} r_k + \beta_k\, d_{k-1}$
- $d_k^\mathsf{T} A\, d_{k+1} = 0$
- exakte Liniensuche

Korrektur der Gradientenrichtung

Minimiere $f(x)$

- $r_k = \nabla f(x_k)$
- $d_0 = -M^{-1} r_0$
- $d_k = \underline{-M^{-1} r_k} + \underline{\beta_k\, d_{k-1}}$
- verschiedene Möglichkeiten, β_k zu wählen
- inexakte Liniensuche (Wolfe oder starke Wolfe)

↑ Dieselben Zutaten wie Gradientenverfahren!

Nichtlineares CG-Verfahren

Setze $k := 0$
Setze $r_0 := \nabla f(x_0)$
Setze $d_0 := -M^{-1}\, r_0$
Setze $\delta_0 := -r_0^\mathsf{T} d_0$ // $\delta_0 = \|\nabla_M f(x_0)\|_M^2$
while Abbruchkriterium nicht erfüllt **do**
    Bestimme α_k über eine Liniensuche
    Setze $x_{k+1} := x_k + \alpha_k\, d_k$
    Setze $r_{k+1} := \nabla f(x_{k+1})$
    Setze $d_{k+1} := -M^{-1} r_{k+1}$
    Setze $\delta_{k+1} := -r_{k+1}^\mathsf{T} d_{k+1}$ // $\delta_{k+1} = \|\nabla_M f(x_{k+1})\|_M^2$
    Wähle β_{k+1}
    Setze $d_{k+1} := d_{k+1} + \beta_{k+1}\, d_k$
    Setze $k := k + 1$
end while
return x_k

Nächste Woche

- keine Konsultation am 16.12.2020
- Behandlung des Materials der Wochen 8 und 9 (Trust-Region-Verfahren) in der nächsten Konsultation am 06.01.2021

Zeit für Ihre Fragen

Was sind Ihre Fragen zu den Themen der Woche?

$\rightarrow$ Benutzen Sie den Chat.

Fragen und Antworten 1

Fragen und Antworten 2