Metoder för icke-linjära problem med bivillkor

Metoder för att lösa icke-linjära problem med bivillkor kan delas in i två typer:

- *Tillåtna-punkter-metoder*, som ser till att varje iterat x_i är en tillåten punkt.
- *Straffmetoder*, som som tillåter att någon punkt x_i är otillåten men $\lim x_i = x_*$ är tillåten.

Tillåtna-punkter-metoder, linjära bivillkor

Om vårt problem har endast linjära bivillkor

$$\begin{array}{ll} \min\limits_{p \in \Re^n} & f(\bar{x} + p) \\ \text{s.t.} & Ap = 0 \end{array}$$

så kan vi i stället lösa

$$\min_{v \in \Re^{n-m}} \phi(v) = f(\bar{x} + Zv),$$

där Z är en nollrumsmatris till A.

Sökriktningen v tas fram genom att lösa systemet

$$Z^T \nabla^2 f(x) Z v = Z^T \nabla f(x),$$

och sedan beräknas $p = Zv$.

Tillåtna-punkter-metoder, linjära bivillkor, forts.

Teoretiskt fungerar detta oberoende av vilken nollrumsmatris Z som konstrueras, men störningskänsligheten påverkas och

$$\kappa(Z^T \nabla^2 f(x) Z) \leq \kappa(\nabla^2 f(x)) \times \kappa(Z^T Z),$$

där $\kappa(A)$ är konditionstalet till A. Detta innebär att QR-faktoriseringen är det numeriskt bästa valet. Den kan dock förstöra strukturen hos hessianen: Om

$$\nabla^2 f(x) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 8 \end{bmatrix} \text{ och } Z = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 \\ 1 & -1 & -1 \\ -1 & 1 & -1 \\ -1 & -1 & 1 \end{bmatrix}$$

blir

$$Z^T \nabla^2 f(x) Z = \frac{1}{4} \begin{bmatrix} 15 & 3 & -5 \\ 3 & 15 & -9 \\ -5 & -9 & 15 \end{bmatrix}.$$

Tillåtna-punkter-metoder, linjära bivillkor, forts.

Om vårt problem har linjära olikhetsbivillkor finns *aktiva-mängds*-metoder, som söker minimum på en mängd av aktiva bivillkor och modifierar mängden tills lösningen till det fullständiga problemet funnits.

Algoritm:

- Om x_k optimal på den aktiva mängden:
  - Beräkna Lagrange-multiplikatorerna λ_i för alla aktiva bivillkor.
  - Om den aktiva mängden är tom eller alla $\lambda_i \geq 0$, avsluta. x_k är lokal minimerare till problemet.
  - Annars, ta bort ett bivillkor som svarar mot ett $\lambda_i < 0$.
- Beräkna en sökriktning p som är tillåten m a p de aktiva bivillkoren.
- Bestäm en steglängd α som uppfyller $f(x_k + \alpha p) < f(x_k)$ och inte överskrider något bivillkor.
- Uppdatera punkten $x_{k+1} = x_k + \alpha p$. Modifiera den aktiva mängden om något nytt bivillkor aktiverats.

Tillåtna-punkter-metoder, icke-linjära bivillkor

Om vi utgår från ett icke-linjärt problem med icke-linjära likhetsbivillkor:

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & g(x) = 0 \end{array}$$

och konstruerar Lagrange-funktionen

$$\mathcal{L}(x,\lambda) = f(x) - \lambda^T g(x),$$

så kan vi tillämpa Newtons metod på först ordningens villkor

$$\nabla \mathcal{L}(x,\lambda) = 0.$$

Newtons formel för detta problem blir

$$\begin{bmatrix} x_{k+1} \\ \lambda_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ \lambda_k \end{bmatrix} + \begin{bmatrix} p_k \\ \nu_k \end{bmatrix},$$

där p_k och ν_k tagits fram som lösningen till

$$\nabla^2 \mathcal{L}(x_k,\lambda_k) \begin{bmatrix} p_k \\ \nu_k \end{bmatrix} = -\nabla \mathcal{L}(x_k,\lambda_k).$$

Tillåtna-punkter-metoder, icke-linjära bivillkor, forts.

Det systemet har formen

$$\begin{bmatrix} \nabla^2_{xx}\mathcal{L}(x_k,\lambda_k) & -\nabla g(x_k) \\ -\nabla g(x_k)^T & 0 \end{bmatrix} \begin{bmatrix} p_k \\ \nu_k \end{bmatrix} = \begin{bmatrix} -\nabla_x \mathcal{L}(x_k,\lambda_k) \\ g(x_k) \end{bmatrix}, \tag{1}$$

vilket motsvarar första ordningens villkor för problemet

$$\begin{array}{ll} \min\limits_p & \frac{1}{2} p^T \nabla^2_{xx}\mathcal{L}(x_k,\lambda_k) p + p^T \nabla_x \mathcal{L}(x_k,\lambda_k) \\ \text{s.t.} & \nabla g(x_k)^T p + g(x_k) = 0, \end{array} \tag{2}$$

dvs vi använder en kvadratisk Taylor-approximation av Lagrange-funktionen och en linjär approximation av bivillkoren. Detta kallas för *sekvensiell kvadratisk programmering*, då vi löser en sekvens av kvadratiska problem.

Sekvensiell kvadratisk programmering, härledning

Studera problemet (2)

$$\begin{array}{ll} \min\limits_p & \frac{1}{2} p^T \nabla^2_{xx}\mathcal{L} p + p^T \nabla_x \mathcal{L} \\ \text{s.t.} & \nabla g^T p + g = 0. \end{array}$$

och definiera Lagrangefunktionen

$$\mathcal{M}(p,v) = \frac{1}{2} p^T \nabla^2_{xx}\mathcal{L} p + p^T \nabla_x \mathcal{L} - (\nabla g^T p + g)^T v.$$

Första ordningen villkor blir då

$$\begin{aligned} \nabla_p \mathcal{M} &= \nabla^2_{xx}\mathcal{L} p + \nabla_x \mathcal{L} - \nabla g v = 0 \\ \nabla_v \mathcal{M} &= \nabla g^T p + g = 0 \end{aligned}$$

eller

$$\begin{bmatrix} \nabla^2_{xx}\mathcal{L} & -\nabla g \\ -\nabla g^T & 0 \end{bmatrix} \begin{bmatrix} p \\ v \end{bmatrix} = \begin{bmatrix} -\nabla_x \mathcal{L} \\ g \end{bmatrix}.$$

Sekvensiell kvadratisk programmering, minkva

För ett minsta-kvadratproblem

$$\begin{array}{ll} \min\limits_x & f(x) = \frac{1}{2}\|F(x)\|^2 \\ \text{s.t.} & g(x) = 0 \end{array}$$

har vi

$$\begin{aligned} \nabla f &= J^T F \\ \nabla^2 f &= J^T J + Q \end{aligned}$$

Lagrangefunktionen blir då

$$\mathcal{L} = \frac{1}{2}\|F\|^2 - g^T \lambda$$

med partialderivatorna

$$\nabla_x \mathcal{L} = J^T F - \nabla g \lambda, \;\; \nabla_\lambda \mathcal{L} = -g,$$

$$\nabla^2_{xx}\mathcal{L} = J^T J + Q - \underbrace{\sum \lambda_i \nabla^2 g_i}_{Q_g},$$

$$\nabla^2_{x\lambda}\mathcal{L} = -\nabla g, \;\; \nabla^2_{\lambda x}\mathcal{L} = -\nabla g^T, \;\; \nabla^2_{\lambda\lambda}\mathcal{L} = 0$$

$$\begin{aligned} \nabla \mathcal{L} &= \begin{bmatrix} J^T F - \nabla g \lambda \\ -g \end{bmatrix} \\ \nabla^2 \mathcal{L} &= \begin{bmatrix} J^T J + Q - Q_g & -\nabla g \\ -\nabla g^T & 0 \end{bmatrix}. \end{aligned}$$

Sekvensiell kvadratisk programmering, minkva

Om vi ignorerar krökningarna Q och Q_g och sätter $A = \nabla g^T$, så få vi

$$\begin{bmatrix} J^TJ & -A^T \\ -A & 0 \end{bmatrix} \begin{bmatrix} p \\ v \end{bmatrix} = \begin{bmatrix} -J^TF + A^T\lambda \\ g \end{bmatrix}, \qquad (3)$$

vilket motsvarar att vi löser

$$\begin{aligned} \min_p \quad & \frac{1}{2}\|Jp + F\|^2 \\ \text{s.t.} \quad & Ap + g = 0. \end{aligned}$$

Om vi sätter $r = -(Jp + F)$ så blir rad 1 av ekvation (3)

$$\begin{aligned} J^TJp - A^Tv &= -J^TF + A^T\lambda & (4) \\ J^Tr + A^T\lambda &= -A^Tv. & (5) \end{aligned}$$

Om vi då sätter $v = 0$ så kan vi lösa följande systemekvation i stället för ekvation (3)

$$\begin{bmatrix} 0 & 0 & A \\ 0 & I & J \\ A^T & J^T & 0 \end{bmatrix} \begin{bmatrix} \lambda \\ r \\ p \end{bmatrix} = \begin{bmatrix} -g \\ -F \\ 0 \end{bmatrix}, \qquad (6)$$

dvs vi bestämmer en uppskattning till lagrangemultiplikatorerna i varje iterationssteg i stället för att uppdatera approximationerna.

Tillåtna-punkter-metoder, icke-linjära bivillkor, forts.

Som vanligt måste Newtons metod modifieras för att garantera konvergens mot ett minimum till objektfunktionen.

För att uppfylla kraven på sökriktningen är det vanligt att hålla den reducerade hessianen till det kvadratiska problemet positivt definit.

För att uppfylla kraven på steglängden är det vanligt att använda en *meritfunktion* för att avgöra om $[x_{k+1}, \lambda_{k+1}]^T$ är en bättre approximation av lösningen än $[x_k, \lambda_k]^T$. Ett exempel på en meritfunktion är

$$\begin{aligned} \mathcal{M}(x) &= f(x) + \rho g(x)^T g(x) \\ &= f(x) + \rho \sum_{i=1}^{m} g_i(x)^2, \quad \rho > 0. \end{aligned}$$

Tillåtna-punkter-metoder, icke-linjära bivillkor, forts.

En liknande metod som också utgår från ekvationssystemet

$$\begin{bmatrix} \nabla^2_{xx}\mathcal{L}(x_k, \lambda_k) & -\nabla g(x_k) \\ -\nabla g(x_k)^T & 0 \end{bmatrix} \begin{bmatrix} p_k \\ \nu_k \end{bmatrix} = \begin{bmatrix} -\nabla_x \mathcal{L}(x_k, \lambda_k) \\ g(x_k) \end{bmatrix},$$

är den *reducerade gradient-metoden*. Den delar upp sökriktningen i två ortogonala komponenter p_z och p_y:

$$p_k = Z_k p_z + Y_k p_y,$$

där Z_k är en bas för nollrummet till jacobianen för bivillkoren och Y_k är en bas för värderummet till jacobianen. Här bestäms först p_z som lösningen till

$$Z_K^T \nabla^2_{xx}\mathcal{L}(x_k, \lambda_k) Z_k p_z = -Z_k^T \nabla_x \mathcal{L}(x_k, \lambda_k)$$

och sedan p_y som lösningen till det icke-linjära kvadratiska systemet

$$g(x_k + Z_k p_z + Y_k p_y) = 0.$$