Aalto-yliopisto
Perustieteiden korkeakoulu

Comparison of line search methods in unconstrained optimization

Valmiin työn esittely

Einari Tuukkanen

30.08.2019

Ohjaaja: *Juho Andelmin*

Valvoja: *Fabricio Oliveira*

Systeemianalyysin laboratorio

Tausta

- Nonlinear Unconstrained Optimization

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x})$$

- Erilaisia algoritmeja – sama perusajatus

Algorithm 1 Conceptual optimisation algorithm

initialise. iteration count $k = 0$, starting point x_0
while stopping criteria are not met **do**
    compute direction d_k
    compute step size λ_k
    $x_{k+1} = x_k + \lambda_k d_k$
    $k \leftarrow k + 1$
end while
return x_k.

Lähde: MS-E2122 - Nonlinear Optimization, Fabricio Oliveira, 2018

Käytetyt termit
- Kohdefunktio
- Päämetodi
- Viivahaku

Tutkimuskysymykset

- Vaikuttaako viivahaun valinta päämetodin tehokkuuteen?
- Kuinka suuri merkitys viivahaun valinnalla voidaan odottaa olevan päämetodin tehokkuuteen?
- Miten MS-E2122 Nonlinear Optimization kurssilla esitellyt viivahaut vertautuvat keskenään?

Rajaukset

- Tarkastellaan kahta kohdefunktiota
  - Konveksi
  - Ekstreemipiste globaali optimi
- Satunnaiset aloituspisteet
- Ongelmien dimensio n = 50

Matrix Square Sum

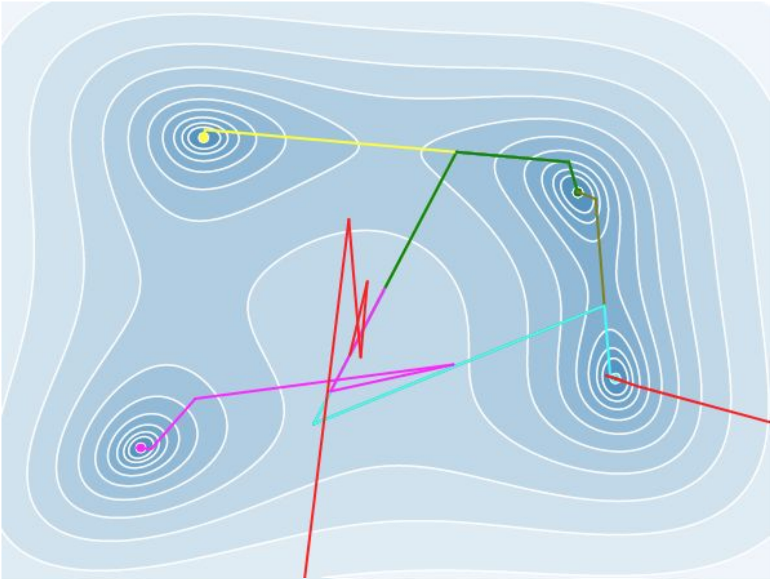
$$f(\mathbf{x}) = \|A\mathbf{x} + \mathbf{b}\|^2 + c\|\mathbf{x}\|^2$$

Negative Entropy

$$f(\mathbf{x}) = \sum_{i=1}^{n} x_i \log x_i$$

Rajaukset

- Neljä päämetodia
  - Newton’s
  - Gradient Descent
  - Conjugate Gradient
  - Heavy Ball
- Kahdeksan viivahakua
  - Constant
  - Bisection
  - Golden Section
  - Dichotomous
  - Fibonacci
  - Newton’s
  - Uniform
  - Armijo’s



Himmelblau + NM + 8 viivahakua, $x_0 = (0,0)$

- Constant
- Bisection
- Dichotomous
- Newton’s
- Armijo’s
- Golden Section, Fibonacci, Uniform

Menetelmät ja työkalut

- **Toteutetaan algoritmit itse Pythonilla**
- Määritetään parametrit
  - Valitaan parametreille muutamat vaihtoehtoiset arvot
  - Aloituspisteitä 10 kpl – muuten identtiset olosuhteet tehokkuusvertailun ajojen kanssa
  - Ajetaan jokainen kohdefunktio-päämetodi-viivahaku yhdistelmä ja valitaan parhaat parametrien arvot
- Ajetaan jokainen yhdistelmä 1000 aloituspisteellä ja vertaillaan tehokkuuksia ensisijaisesti kahdella kriteerillä
  1. Onnistumisprosentti
  2. Kesto
- Tulosten analysointi

Menetelmät ja työkalut

- Toteutetaan algoritmit itse Pythonilla
- **Määritetään parametrit**
  - Valitaan parametreille muutamat vaihtoehtoiset arvot
  - Aloituspisteitä 10 kpl – muuten identtiset olosuhteet tehokkuusvertailun ajojen kanssa
  - Ajetaan jokainen kohdefunktio-päämetodi-viivahaku yhdistelmä ja valitaan parhaat parametrien arvot
- Ajetaan jokainen yhdistelmä 1000 aloituspisteellä ja vertaillaan tehokkuuksia ensisijaisesti kahdella kriteerillä
  1. Onnistumisprosentti
  2. Kesto
- Tulosten analysointi

Parametrien määrittäminen

Esimerkkinä Matrix Square Sum + Newton's Method + Armijo's Search

Parameter	MatrixSquareSum	NegativeEntropy
λ	0.9, 1, 1.1	0.9, 1, 1.1
α	0.1, 0.25, 0.5	0.1, 0.25, 0.5
β	0.5, 0.75, 0.9	0.5, 0.75, 0.9
l	1e-07	1e-07

Ehdokkaat parametrien arvoille

Vastaava taulukko jokaiselle kohdefunktio-päämetodi-viivahaku-yhdistelmälle
→ Yhteensä 64 taulukkoa

NewtonsMethod		
(λ, α, β)	s (%)	t (ms)
(1, 0.1, 0.5)	100.0	429.7
(1, 0.25, 0.75)	100.0	484.5
(1, 0.25, 0.5)	100.0	493.8
(1, 0.25, 0.9)	100.0	493.9
(1, 0.1, 0.9)	100.0	498.7

Parhaat parametrit Matrix Square Sum funktiolle, kun käytetään Newton's Methodia

Parhaat parametrit Negative Entropy funktiolle, kun käytetään Newton's Methodia

NewtonsMethod		
(λ, α, β)	s (%)	t (ms)
(1.1, 0.5, 0.9)	100.0	2094.1
(1, 0.5, 0.75)	100.0	2457.5
(1, 0.5, 0.5)	100.0	2659.1
(1, 0.5, 0.9)	100.0	2756.4
(1.1, 0.25, 0.75)	100.0	2854.2

Menetelmät ja työkalut

- Toteutetaan algoritmit itse Pythonilla
- Määritetään parametrit
  - Valitaan parametreille muutamat vaihtoehtoiset arvot
  - Aloituspisteitä 10 kpl – muuten identtiset olosuhteet tehokkuusvertailun ajojen kanssa
  - Ajetaan jokainen kohdefunktio-päämetodi-viivahaku yhdistelmä ja valitaan parhaat parametrien arvot
- **Ajetaan jokainen yhdistelmä 1000 aloituspisteellä**
  - Vertaillaan tehokkuuksia ensisijaisesti kahdella kriteerillä
    1. Onnistumisprosentti
    2. Kesto
- Tulosten analysointi

Tulokset

Esimerkkinä Matrix Square Sum + Newton's Method

Line Search Name	s (%)	t (ms)	k	f_n	t_{LS} (ms)	k_{LS}
ConstantSearch	100.0	445.8	1.0	7.0	0.0	0.0
GoldenSectionSearch	100.0	445.0	1.0	101.0	2.6	47.0
BisectionSearch	99.6	535.9	1.0	35.0	89.0	28.0
DichotomousSearch	100.0	429.1	1.0	77.0	3.3	35.0
FibonacciSearch	100.0	453.2	1.0	119.0	4.5	55.0
UniformSearch	100.0	455.0	1.0	156.0	7.2	148.0
NewtonsSearch	100.0	443.5	1.0	8.0	0.5	0.0
ArmijoSearch	100.0	424.2	1.0	10.0	0.9	0.0

Ajojen tulokset Matrix Square Sum kohdefunktiolle Newton's Methodilla eri viivahauilla

s onnistumisprosentti
t kesto

k iteraatiot
f_n kohdefunktiokutsut
t_{LS} pelkän viivahaun kesto
k_{LS} viivahaun iteraatiot

Vastaava taulukko jokaiselle kohdefunktio-päämetodi-yhdistelmälle
→ Yhteensä 8 taulukkoa

Menetelmät ja työkalut

- Toteutetaan algoritmit itse Pythonilla
- Määritetään parametrit
  - Valitaan parametreille muutamat vaihtoehtoiset arvot
  - Aloituspisteitä 10 kpl – muuten identtiset olosuhteet tehokkuusvertailun ajojen kanssa
  - Ajetaan jokainen kohdefunktio-päämetodi-viivahaku yhdistelmä ja valitaan parhaat parametrien arvot
- Ajetaan jokainen yhdistelmä 1000 aloituspisteellä
  - Vertaillaan tehokkuuksia ensisijaisesti kahdella kriteerillä
    1. Onnistumisprosentti
    2. Kesto
- **Tulosten analysointi**

Tulosten analysointi

Line Search Method	NM	GDM	CGM	HBM
ConstantSearch	2 081,95	2 347,15	2 691,35	2 359,90
GoldenSectionSearch	1 241,65	168,55	484,60	160,05
BisectionSearch	1 363,75	457,85	957,20	399,15
DichotomousSearch	1 330,15	162,40	501,35	139,10
FibonacciSearch	1 276,95	206,35	570,75	212,65
UniformSearch	1 385,50	255,50	3 464,65	228,65
NewtonsSearch	1 354,75	456,05	1 677,25	458,05
ArmijoSearch	1 315,55	110,45	301,85	116,40

Päämetodikohtaiset keskiarvokestot eri menetelmille, kun tarkastellaan molempia kohdefunktioita yhdessä

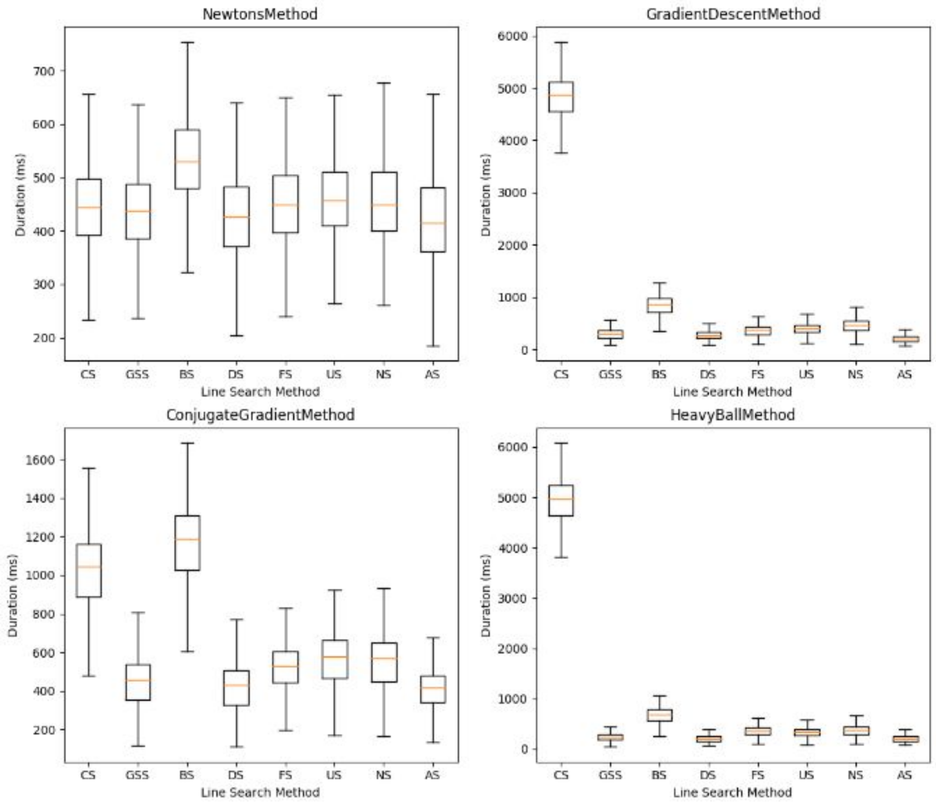
Väritys kolumneittain
Tummempi punainen = huonompi
Tummempi vihreä = parempi

Line Search Method	MSS	NE	OVERALL
ConstantSearch	2 706,95	2 033,23	2 370,09
GoldenSectionSearch	346,48	680,95	513,71
BisectionSearch	795,23	793,75	794,49
DichotomousSearch	333,73	732,78	533,25
FibonacciSearch	419,63	713,73	566,68
UniformSearch	431,20	2 235,95	1 333,58
NewtonsSearch	456,18	1 516,88	986,53
ArmijoSearch	314,03	608,10	461,06

Kohdefunktiokohtaiset keskiarvokestot eri menetelmille, kun tarkastellaan kaikkia päämetodeja yhdessä

Tulosten analysointi

Esimerkkinä Matrix Square Sum



Box plot -kuvaaja metodien kestoista Matrix Square Sum kohdefunktiolle

Vastaava kuvaaja Negative Entropy kohdefunktiolle

Havainnot

- Armijo’s Search vaikuttaa odotetusti parhaimmalta
    - Ei muutenkaan suuria yllätyksiä
- Paras metodi n. 11 % parempi kuin seuraavaksi paras
    - Neljäneksi paras n. 23 % päässä parhaasta
- Varianssit linjassa tehokkuuden kanssa

Lähteet: Convex Optimization, Boyd, Stephen; Lieven Vandenberghe (2004)

Haasteet ja jatkotutkimus

- Runsaasti oletuksia
  - Kohdefunktioiden valinta, parametrien valinta, ohjelmointikieli, toteutustapa, prosessori jne.

- Jos tekisin uudestaan
  - Parempi ratkaisu parametrien valinnalle
  - Lisää kohdefunktioita
  - Tiputetaan odotetusti huonot ratkaisut pois

Lähteet: Convex Optimization, Boyd, Stephen; Lieven Vandenberghe (2004)

Johtopäätökset

- Viivahaun käyttäminen lähes aina parempi kuin vakio askelkoko
- Viivahauilla odotetusti eroja tehokkuudessa
  - Testatuista viivahauista Armijo’s ja Golden Section olivat parhaat
  - Tehokkuuserot suuruusluokkaa 10-20%

Lisäksi

- Parametrien valinta haastavaa
- Parametrien optimointi hyödyllisempää kuin viivahakujen vertailu
  - Viivahakujen vertailu perusteltua, jos samaa kohdefunktiota halutaan optimoida pidemmälle

Lähteet: Convex Optimization, Boyd, Stephen; Lieven Vandenberghe (2004)