Conjugate gradient methods = cg – Methods

1. Conjugate directions -definition
2. Orthogonalization
3. cg-method for strictly convex quadratic functions
4. Preconditioning
5. cg-method for smooth non quadratic functions
6. Application
7. Experiments

Conjugate directions -definition

Bijective affine mapping:

circle ∪ square of tangents ∪ cross of diameters

orthogonal to each other

orthogonal w.r.t. usual scalar product

$$\boldsymbol{x} \bullet \boldsymbol{y} = \boldsymbol{x}^T \boldsymbol{E} \boldsymbol{y}$$

$$\downarrow$$

Ellipsis ∪ parallelogram of tangents ∪ cross of diameters

conjugate to each other

orthogonal w.r.t. a new skalarproduct

$$\boldsymbol{x} \bullet \boldsymbol{y} = \boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{y}, \ \boldsymbol{Q} \in \mathbb{SPD}$$

Def.: $\boldsymbol{Q} \in \mathbb{SPD}$,

$\boldsymbol{x}$ **$\boldsymbol{Q}$ -conjugate ($\boldsymbol{Q}$ -orthogonal)** to $\boldsymbol{y} :\Longleftrightarrow \boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{y} = 0$

Orthogonalization by E. Schmidt (recall)

Algorithm A7 (GRAM-SCHMIDT)

S0. Given linear independent $\boldsymbol{p}^0, \cdots, \boldsymbol{p}^{m-1} \in \mathbb{R}^n, m \leq n-1$, $\boldsymbol{Q} \in \mathbb{SPD}^n$, set $\boldsymbol{d}^0 := \boldsymbol{p}^0, k := 0$

S1. **while** $k < m$

$$\boldsymbol{d}^k = \boldsymbol{p}^k - \sum_{i=0}^{k-1} \beta_{ik} \boldsymbol{d}^i \qquad (\boldsymbol{Q} \text{ orthogonal projection})$$

with

$$\beta_{ik} := \frac{(\boldsymbol{p}^k)^T \boldsymbol{Q} \boldsymbol{d}^i}{(\boldsymbol{d}^i)^T \boldsymbol{Q} \boldsymbol{d}^i} \qquad \text{(Fourier's coefficients)}$$

S2. Set $k := k + 1$ and goto S1.

Minimal property of variety $\boldsymbol{x}^0 + \mathcal{L}\{\boldsymbol{d}^0, ..., \boldsymbol{d}^k\}$

Theorem:cf. [Reinhardt et.al. Th. 3.68] Assume:

1. $f(\boldsymbol{x}) := \boldsymbol{b}^T \boldsymbol{x} + \frac{1}{2} \boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x}, \quad \boldsymbol{Q} \in \mathbb{SPD}^n.$
2. $\boldsymbol{d}^0, \ldots, \boldsymbol{d}^{n-1}$ $\boldsymbol{Q}-$conj. system and $\boldsymbol{x}^0 \in \mathbb{R}^n$
3. $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \lambda_k \boldsymbol{d}^k$ for $k = 0, \ldots, n-1$
   $\lambda_k > 0$ perfect step length for $f(\boldsymbol{x}^k + \lambda \boldsymbol{d}^k) \to \min$
4. $V_{k+1} := \{\boldsymbol{x} \mid \boldsymbol{x} = \boldsymbol{x}^0 + \sum_{i=0}^{k} \mu_i \boldsymbol{d}^i, \ \mu_i \in \mathbb{R}\}$

Then hold (!! $\nabla f(\boldsymbol{x}^{k+1}) = \boldsymbol{Q}\boldsymbol{x}^{k+1} + \boldsymbol{b} = \boldsymbol{Q}\boldsymbol{x}^k + \boldsymbol{b} + \lambda \boldsymbol{Q}\boldsymbol{d}^k$)

i) $\nabla f(\boldsymbol{x}^{k+1})^T \boldsymbol{d}^j = 0$ for $0 \leq j \leq k, \ k = 0, 1, ..., n-1$

ii) $\boldsymbol{x}^{k+1} = \text{argmin}\{f(\boldsymbol{x}) \mid \boldsymbol{x} \in V_{k+1}\}, \quad k = 0, 1, ..., n-1$

iii) $\nabla f(\boldsymbol{x}^{i+1}) - \nabla f(\boldsymbol{x}^i) = \lambda_i \boldsymbol{Q} \boldsymbol{d}^i, \quad i = 0, 1, ..., n-1$

cg-method - Algorithm

Algorithm A8 (cg-method for strictly convex quadratic functions)

S0. $\boldsymbol{x}^0 \in \mathbb{R}^n,\ \boldsymbol{d}^0 := -\nabla f(\boldsymbol{x}^0),\ \varepsilon > 0,\ k := 0$

S1. **if** $\|\nabla f(\boldsymbol{x}^k)\| = 0 (< \varepsilon$ on computer$)$, STOPP

S2. perfect step length: $\lambda_k := -\dfrac{\nabla f(\boldsymbol{x}^k)^T \boldsymbol{d}^k}{(\boldsymbol{d}^k)^T \boldsymbol{Q} \boldsymbol{d}^k}$

update: $\boldsymbol{x}^{k+1} := \boldsymbol{x}^k + \lambda_k \boldsymbol{d}^k$

S3. New descent direction: (!! only last summand)

$\boldsymbol{d}^{k+1} := -\nabla f(\boldsymbol{x}^{k+1}) + \beta_k \boldsymbol{d}^k$ with $\beta_k = \dfrac{\nabla f(\boldsymbol{x}^{k+1})^T \boldsymbol{Q} \boldsymbol{d}^k}{(\boldsymbol{d}^k)^T \boldsymbol{Q} \boldsymbol{d}^k}$

S4. Set $k := k + 1$ and goto S1.

Properties of Algorithm A8

Theorem:cf. [Reinhardt et al. Th. 3.69] Let $\boldsymbol{x}^k$, $k = 0, 1, \ldots, m+1 \leq n$ generated by Algorithm A8. Then (a)-(f) hold for $k = 1, 2, \ldots, m$

(a) $\nabla f(\boldsymbol{x}^j)^T \nabla f(\boldsymbol{x}^i)$	$= 0,$	$0 \leq i < j \leq k$	OGS
(b) $(\boldsymbol{d}^j)^T \boldsymbol{Q} \boldsymbol{d}^i$	$= 0,$	$0 \leq i < j \leq k,$	$\boldsymbol{Q}$ -OGS
(c) $\nabla f(\boldsymbol{x}^j)^T \boldsymbol{d}^i$	$= 0,$	$0 \leq i < j \leq k$	minimality w.r.t. V_k
(d) $\nabla f(\boldsymbol{x}^j)^T \boldsymbol{d}^j$	$= -\|\nabla f(\boldsymbol{x}^j)\|^2,$	$0 \leq j \leq k,$	descent cond.(DC)
(e) $V_k - \boldsymbol{x}^0$	$= \mathcal{L}\{\boldsymbol{d}^0, \ldots, \boldsymbol{d}^{k-1}\} = \mathcal{L}\{\nabla f(\boldsymbol{x}^0), \ldots, \nabla f(\boldsymbol{x}^{k-1})\}$		
(f) $V_k - \boldsymbol{x}^0$	$= \mathcal{L}\{\nabla f(\boldsymbol{x}^0), \boldsymbol{Q}\nabla f(\boldsymbol{x}^0), \ldots, \boldsymbol{Q}^{k-1}\nabla f(\boldsymbol{x}^0)\}$		
(g) $\nabla f(\boldsymbol{x}^j)^T \boldsymbol{Q} \boldsymbol{d}^i$	$= 0,$	$0 \leq i < j - 1 \leq k - 1$ for $k = 2, 3, \ldots, m$	

The algorithm A8 stops after $k \leq n$ steps at $\boldsymbol{x}^k = \boldsymbol{x}^*$ **(finite stop property)**

Lemma: BFGS / DFP - method, f quadratic, strictly convex, perfect step length $\Longrightarrow$

1. $\boldsymbol{d}^k$ create a $\boldsymbol{Q}$ -OGS ($\Rightarrow \boldsymbol{x}^* = \boldsymbol{x}^k,\ k \leq n$)
2. $\boldsymbol{H}_0 = \boldsymbol{E} \Rightarrow$ BFGS ($\equiv$ DFP) $\equiv$ CG

Numerical realization of A8

S0. $\boldsymbol{x}^0 \in \mathbb{R}^n,\ \boldsymbol{g}^0 := \nabla f(\boldsymbol{x}^0), \boldsymbol{d}^0 = -\boldsymbol{g}^0,\ \varepsilon > 0,\ k := 0$

S1. **if** $\|\boldsymbol{g}^k\| = 0\ (< \varepsilon$ on computer$)$, STOPP

S2. perfect step length: $\lambda_k := \dfrac{\|\boldsymbol{g}^k\|^2}{(\boldsymbol{d}^k)^T \boldsymbol{Q} \boldsymbol{d}^k}$

$\boldsymbol{x}^{k+1} := \boldsymbol{x}^k + \lambda_k \boldsymbol{d}^k$

$\boldsymbol{g}^{k+1} := \boldsymbol{g}^k + \lambda_k \boldsymbol{Q} \boldsymbol{d}^k\ (= \nabla f(\boldsymbol{x}^{k+1}))$

S3. New descent direction:

$\boldsymbol{d}^{k+1} := -\boldsymbol{g}^{k+1} + \beta_k \boldsymbol{d}^k$ with $\beta_k = \dfrac{\|\boldsymbol{g}^{k+1}\|^2}{\|\boldsymbol{g}^k\|^2}$

S4. Set $k := k + 1$ and goto S1.

Calculate $\boldsymbol{g}^{k+1}$ sometimes directly (error transmission!!)

Structure of eigenvalues

Theorem: [Bertsekas(1999), Pytlak(2009), Reinhardt et. al. Th. 3.72] Given f quadratic, strictly convex with minimum point $\boldsymbol{x}^*$ and EV(Q): $\lambda_1, ..., \lambda_n \in \mathbb{R}$, iterations $\boldsymbol{x}^k$, $k = 0, 1, \ldots, m$ w.r.t. Algorithm A8 such that $\boldsymbol{x}^m = \boldsymbol{x}^*$. Then the estimation

$$f(\boldsymbol{x}^{k+1}) - f(\boldsymbol{x}^*) \leq \max_{j=1,\ldots,n} \left(P_{k+1}(\lambda_j)\right)^2 \left(f(\boldsymbol{x}^0) - f(\boldsymbol{x}^*)\right)$$

is valid for $k = 0, ..., m-1$ and **each** Polynom $P_{k+1}(\lambda)$ of degree $k+1$ with $P_{k+1}(0) = 1$.

Corollary: If $0 < a < \lambda_1 \leq ... \leq \lambda_{n-k} \leq b < \lambda_{n-k+1} \leq ... \leq \lambda_n$ then

$$f(\boldsymbol{x}^{k+1}) - f(\boldsymbol{x}^*) \leq \left(\frac{b-a}{b+a}\right)^2 \left(f(\boldsymbol{x}^0) - f(\boldsymbol{x}^*)\right)$$

Corollary: If only $r \leq n$ of eigenvalues of $\boldsymbol{Q}$ are pairwise disjoint then

$$\boldsymbol{x}^r = \boldsymbol{x}^*$$

Preconditioning – intention: $B \approx Q$

S regular, Transformation: $Sy := x$, $f_S(y) := f(Sy)$,
chain rule: $\nabla f_S(y) = S^T \nabla f(x)$. cg-method for f_S yields with
$B^{-1} := SS^T$ the cg-method with preconditioning in x-data

Algorithm A8 (cg-method for strictly convex quadratic functions)

S0. $x^0 \in \mathbb{R}^n$, $g^0 := \nabla f(x^0), d^0 = -B^{-1}g^0$, $\varepsilon > 0$, $k := 0$

S1. **if** $\|g^k\| = 0$ ($< \varepsilon$ on computer), STOPP

S2. transformed perfect step length: $\lambda_k := \dfrac{(g^k)^T B^{-1} g^k}{(d^k)^T Q d^k}$

$x^{k+1} := x^k + \lambda_k d^k$ (instead $B^{-1}g =: h$ solve fast (!!) $Bh = g$)

$g^{k+1} := g^k + \lambda_k Q d^k$ ($= \nabla f(x^{k+1})$)

S3. New descent direction:

$d^{k+1} := -B^{-1}g^{k+1} + \beta_k d^k$ with $\beta_k = \dfrac{(g^{k+1})^T B^{-1} g^{k+1}}{(g^k)^T B^{-1} g^k}$

S4. Set $k := k + 1$ and goto S1.

Examples for Preconditioning with $B \approx Q$

1. incomplete Cholesky decomposition preconditioner (several strategies): $B = L^T L$
   Application: $L^T y = g$, $Lh = y \Leftrightarrow h = B^{-1}g$
   `fminunc`: Chol. decomp. for r-diagonals of Q
   `EdOptLab`: `cholinc` with drop param. $p \in (0, \infty)$
2. SSOR (symmetric successive overrelaxation, $\omega \in [1, 2)$) preconditioner:
   $B = (D + \omega L)D^{-1}(D + \omega L)^T/(2 - \omega)$ where
   $L_{ij} = Q_{ij}, i > j; L_{ij} = 0$ elsewhere, $D = \text{diag}\,(Q)$
   Application: $(D + \omega L)u = g$; $w = Du$;
   $(D + \omega L)^T h = (2 - \omega)w$
3. Jacobi-preconditioner: $B = \text{diag}\,(Q)$
   Application: $h_i = (Q_{ii})^{-1} g_i$

NLO-adaption

Use of identical reformulation in the quadratic case

$$\beta_k = \frac{\nabla f(x^{k+1})^T Q d^k}{(d^k)^T Q d^k} \qquad f \text{ quadratic}$$

smooth NLO-problem

$$\underset{\text{i)+d) + iii)}}{=} \frac{\nabla f(x^{k+1})\left(\nabla f(x^{k+1}) - \nabla f(x^k)\right)}{\|\nabla f(x^k)\|^2} \qquad \text{POLAK/RIBIERE}$$

$$= \frac{\|\nabla f(x^{k+1})\|^2}{\|\nabla f(x^k)\|^2} \qquad \text{FLETCHER/REEVES}$$

- Perf. step length. $\mapsto$ AR with IP., PW, Backtracking,... the more perfect the better, but effort observe
- There are: the cg for QP, but a lot of cg for NLO

About convergence theory

- **P**oljak/**R**ibiere: $f \in C^2$, f'' unif. $\mathbb{SPD}$, λ_k perfect $\Longrightarrow$ R-linearly convergent [SCHW79, S. 236]
- **F**letcher/**R**eeves: under same assumptions only convergent [Geig90, S. 231]
- **F/R**: $c \le f \in C^{1,L}$, PW $\Longrightarrow$ culm. point (CP) is stationary [Geig90, S. 229]
- **P/R** $c \le f \in C^{1,L}$, PW, $\|x^{k+1} - x^k\| \to 0 \Longrightarrow$ CP is stationary [Geig90, S. 232]
- **P/R**, **F/R**: $f \in C^3$, f'' unif. $\mathbb{SPD}$, λ_k perfect $\Longrightarrow$ n-step quadratic. conv. with restart [COHEN 72], [Schw79, S. 238]
- **F/R**: $d^k \to$ parallel $\Longrightarrow$ restart with $d^k = -\nabla f(x^k)$

Application of cg-methods

- approximate solution of linear Newton-equation, see inexact Newton methods and TR-methods
- large scale NLO using preconditioning at periodical restart [Pytlak 2009], faster than BFGS and limited memory BFGS, sophisticated algorithms
- (large scale) strictly convex QP with preconditioning

Experiments with F/R, F/R Restart and P/R

- `CG14`: non perfect step length is bad
- `CG03`: Comparison CG-PR, CG-FR, CG-Q, BFGS, Newton for strictly convex quadratic function till `dim x = 100` for $\kappa(\boldsymbol{Q}) = 100$
- `CG04`: As `CG03` but with $\kappa(\boldsymbol{Q}) = 10000$
- `CG06, CG07, CG09, CG10`: BFGS for strictly convex quadratic function with different preconditioning $(\boldsymbol{H}_0 = \boldsymbol{B})$
- `CG11`: CG-Q with preconditioning
- `CG12`: Comparison CG-PR, CG-FR, BFGS, Banana function, perfect step length
- `CG15`: CG-methods with several restarts
- `CG16`: Effectivity of CG-PR, CG-FR(restart), CG-PR(restart), BFGS for Banana till dimension 100 with parameter $a = 100$ and $a = 1$ (simulation of preconditioning)