The Barzilai-Borwein method

Chenglong Bao

YMSC, Tsinghua University

Acknowledgement: slides based on Prof. Wotao Yin(UCLA).

Main features of the Barzilai-Borwein (BB) method

$$x_{k+1} = x_k - \underbrace{(\nabla^2 f(x_k))^{-1} \nabla f(x_k)}_{\text{Newton.}}$$

- The BB method was published in a 8-page paper[1] in 1988
- It is a gradient method with special step sizes. The method is motivated by Newton's method but does not compute Hessian
- At nearly no extra cost over the standard gradient method, the method is often found to significantly outperform the standard gradient method
- The method is used along with non-monotone line search as a convergence safeguard for non-quadratic problems

[1] J. Barzilai and J. Borwein. Two-point step size gradient method. IMA J. Numerical Analysis 8, 141–148, 1988.

Background

Goal: $\text{minimize}_{\boldsymbol{x}\in\mathbb{R}^n}\ f(\boldsymbol{x})$, where f is a smooth function

Let $\boldsymbol{g}^{(k)} = \nabla f(\boldsymbol{x}^{(k)})$ and $\boldsymbol{F}^{(k)} = \nabla^2 f(\boldsymbol{x}^{(k)})$.

- **gradient method**: $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \alpha_k \boldsymbol{g}^{(k)}$
  - choice of α_k: fixed, exact line search, *or* backtracking line search
  - **pros**: simple
  - **cons**: no use of 2nd order information, relatively slow progress
- **Newton's method**: $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - (\boldsymbol{F}^{(k)})^{-1}\boldsymbol{g}^{(k)}$
  - **pros**: 2nd-order information, 1-step for quadratic function, fast convergence near solution
  - **cons**: forming and computing $(\boldsymbol{F}^{(k)})^{-1}$ is expensive, need modifications if $\boldsymbol{F}^{(k)} \not\succ 0$
- **BB method**: choose α_k so that $\alpha_k \boldsymbol{g}^{(k)}$ "approximates" $(\boldsymbol{F}^{(k)})^{-1}\boldsymbol{g}^{(k)}$

$(\alpha_k I)$

Derive the BB method

$f(x)=0$

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

- Consider quadratic optimization

$$f'(x_k) = \frac{f(x_k)-f(x_{k-1})}{x_k - x_{k-1}}.$$

$$\underset{\boldsymbol{x}}{\text{minimize}}\ q(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}^T\boldsymbol{x},$$

$$\Rightarrow f'(x_k)(x_k - x_{k-1}) = f(x_k) - f(x_{k-1})$$

where $\boldsymbol{A} \succ 0$ is symmetric. Gradient is $\boldsymbol{g}^{(k)} = \boldsymbol{A}\boldsymbol{x}^{(k)} - \boldsymbol{b}$. Hessian is $\boldsymbol{A}$.

- Newton step: $\boldsymbol{d}_{\text{newton}}^{(k)} = -\boldsymbol{A}^{-1}\boldsymbol{g}^{(k)}$

$\approx A^{-1}$

- **Goal:** choose α_k so that $-\alpha_k\boldsymbol{g}^{(k)} = -(\alpha_k^{-1}I)^{-1}\boldsymbol{g}^{(k)}$ approximates $\boldsymbol{d}_{\text{newton}}^{(k)}$
- Define: $\boldsymbol{s}^{(k-1)} := \boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}$ and $\boldsymbol{y}^{(k-1)} := \boldsymbol{g}^{(k)} - \boldsymbol{g}^{(k-1)}$. $\boldsymbol{A}$ satisfies:

$$\boldsymbol{A}\boldsymbol{s}^{(k-1)} = \boldsymbol{y}^{(k-1)}. \leftarrow \text{secant equation.}$$

- Therefore, given $\boldsymbol{s}^{(k-1)}$ and $\boldsymbol{y}^{(k-1)}$, how about choose α_k so that

$$(\alpha_k^{-1}I)\boldsymbol{s}^{(k-1)} \approx \boldsymbol{y}^{(k-1)}$$

- **Goal:**

$$\underline{(\alpha_k^{-1} I)\boldsymbol{s}^{(k-1)} \approx \boldsymbol{y}^{(k-1)}}. \iff s^{(k-1)} \approx (\alpha_k I)\, y^{(k-1)}$$

- **BB method:**
  - Least-squares problem: (let $\beta = \alpha^{-1}$) (BB-1).

$$\alpha_k^{-1} = \arg\min_{\beta} \frac{1}{2}\|\boldsymbol{s}^{(k-1)}\beta - \boldsymbol{y}^{(k-1)}\|^2 \implies \alpha_k^1 = \frac{(\boldsymbol{s}^{(k-1)})^T \boldsymbol{s}^{(k-1)}}{(\boldsymbol{s}^{(k-1)})^T \boldsymbol{y}^{(k-1)}} \gtrsim 0.$$

  - Alternative Least-squares problem: (BB-2)

$$\alpha_k = \arg\min_{\alpha} \frac{1}{2}\|\boldsymbol{s}^{(k-1)} - \boldsymbol{y}^{(k-1)}\alpha\|^2 \implies \alpha_k^2 = \frac{(\boldsymbol{s}^{(k-1)})^T \boldsymbol{y}^{(k-1)}}{(\boldsymbol{y}^{(k-1)})^T \boldsymbol{y}^{(k-1)}}$$

- α_k^1 and α_k^2 are called the BB step sizes.

$$y^{(k-1)} = \nabla f(x^{(k)}) - \nabla f(x^{k-1})$$

$$\langle y^{k-1}, s^{k-1} \rangle \geq 0 \quad \Leftarrow \quad s^{(k-1)} = x^k - x^{k-1}$$

Apply the BB method

- At $k = 0$, $\boldsymbol{x}^{(k-1)}$ and $\boldsymbol{g}^{(k-1)}$ (and thus $\boldsymbol{s}^{(k-1)}$ and $\boldsymbol{y}^{(k-1)}$) are unavailable, so apply 1 iteration of the standard gradient descent.
- Then, switch to the BB method at $k = 1$
- We can use either α_k^1 or α_k^2 for all $k \geq 1$, or alternate between them
- We can also fix $\alpha_k = \alpha_k^1$ or $\alpha_k = \alpha_k^2$ for a few consecutive steps and then alternate.
- It performs very well on minimizing both quadratic and other differentiable functions
- However, f_k and $\|\nabla f_k\|$ are **not** monotonic!

Numerical: steepest descent vs BB on quadratic programming

- **Model:**

$$\underset{\boldsymbol{x}}{\text{minimize}}\ f(\boldsymbol{x}) := \frac{1}{2}\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} - \mathbf{b}^T\boldsymbol{x}.$$

- The template of a gradient iteration

$$\boldsymbol{x}^{k+1} \leftarrow \boldsymbol{x}^{(k)} - \alpha_k(\boldsymbol{A}\boldsymbol{x}^{(k)} - \mathbf{b}).$$

- **Steepest descent** selects $\alpha_k = \arg\min_\alpha f(\boldsymbol{x}^{(k)} - \alpha_k(\boldsymbol{A}\boldsymbol{x}^{(k)} - \mathbf{b}))$, so

$$\alpha_k = \frac{(\boldsymbol{r}^k)^T\boldsymbol{r}^{(k)}}{(\boldsymbol{r}^k)^T\boldsymbol{A}\boldsymbol{r}^{(k)}}$$

  where $\boldsymbol{r}^{(k)} := \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^{(k)}$.

- **BB** selects α_k as

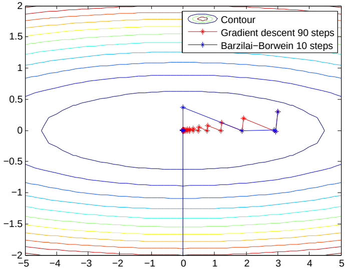
$$\alpha_k^1 = \frac{(\boldsymbol{s}^{(k-1)})^T\boldsymbol{s}^{(k-1)}}{(\boldsymbol{s}^{(k-1)})^T\boldsymbol{y}^{(k-1)}}$$

Numerical example

- Set symmetric matrix $\boldsymbol{A}$ to have the condition number $\frac{\lambda_{\max}(\boldsymbol{A})}{\lambda_{\min}(\boldsymbol{A})} = 50$.
- Stopping criterion:

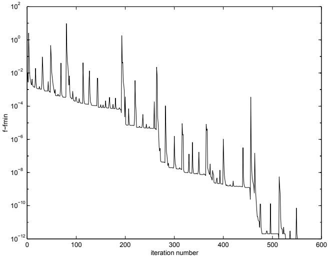
$$\|\boldsymbol{r}^{(k)}\| < 10^{-8}$$

- **Steepest descent** took 90 iterations to stop
- **BB** took only 10 iterations to stop (went very far temporarily and then came back)



Properties of Barzilai-Borwein

- For quadratic functions, it has R-linear convergence[2]
- For 2D quadratic function, it has Q-superlinear convergence[3]
- No convergence guarantee for smooth convex problems. On these problems, we pair up BB with non-monotone line search.



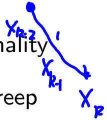
BB on *Laplace2*: $\min \frac{1}{2}\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}^T\boldsymbol{x} + \frac{h^2}{4}\sum_{ijk} u_{ijk}^4$.

[2] Dai and Liao [2002]
[3] Barzilai and Borwein [1988], Dai [2013]

Safeguard: nonmonotone line search

- Definition: line search that permits temporary growth but enforces overall descent of the function value

- For nonconvex problems, they improve the likelihood of global optimality



- Improve convergence speed when a monotone scheme is forced to creep along the bottom of a narrow curved valley

- Early nonmonotone line search method[4] developed for Newton's methods

$d^{k^T}\nabla f(x^k) < 0$

$$f(\boldsymbol{x}^{(k)} + \underline{\alpha \boldsymbol{d}^{(k)}}) \le \boxed{\max_{0\le j\le m_k} f(\boldsymbol{x}^{k-j})} + c_1\alpha\nabla f_k^T \boldsymbol{d}^{(k)}$$

However, it may still kill R-linear convergence. **Example**: $x \in \mathbb{R}$,

$$\text{minimize}_x \; f(x) = \frac{1}{2}x^2, \quad x^0 \neq 0, \quad d^{(k)} = -x^{(k)}.$$

$$x_{k+1} = \begin{cases} -x_k, & \text{if } k \neq i \\ \frac{1}{2^{-k}} x_k, & \text{if } k \neq i^2 \end{cases}$$

$$\alpha_k = \begin{cases} 1 - 2^{-k}, & k = i^2 \text{ for some integer } i, \\ 2, & \text{otherwise,} \end{cases}$$

converges R-linear but fails to satisfy the condition for k large.

[4] Grippo, Lampariello, and Lucidi [1986]

Zhang-Hager nonmonotone line search[5]

1. initialize $0 < c_1 < c_2 < 1$, $C_0 \leftarrow f(\boldsymbol{x}^0)$, $Q_0 \leftarrow 1$, $\eta < 1$, $k \leftarrow 0$
2. while *not converged* do

3a. compute α_k satisfying the modified Wolfe conditions OR

3b. find α_k by backtracking, to satisfy the modified Armijo condition:

$$\text{sufficient decrease:} \quad f(\boldsymbol{x}^{(k)} + \alpha_k \boldsymbol{d}^{(k)}) \le C_k + c_1 \alpha_k \nabla f_k^T \boldsymbol{d}^{(k)}$$

4. $\boldsymbol{x}^{k+1} \leftarrow \boldsymbol{x}^{(k)} + \alpha_k \boldsymbol{d}^{(k)}$
5. $\boxed{Q_{k+1} \leftarrow \eta Q_k + 1, \; C_{k+1} \leftarrow (\eta Q_k C_k + f(\boldsymbol{x}^{k+1}))/Q_{k+1}.}$

$$\Rightarrow \frac{\eta Q_k}{\eta Q_k + 1} C_k + \frac{1}{\eta Q_k + 1} f(x^{k+1})$$

Comments:

- If $\eta = 1$, then $C_k = \frac{1}{k+1} \sum_{j=0}^{k} f_j$.
- Since $\eta < 1$, C_k is a weighted sum of all past f_j, more weights on recent f_j.

[5] Zhang and Hager [2004]

Convergence (advanced topic)

The results below are left to the reader as an exercise.

If $f \in C^1$ and bounded below, $\nabla f_k^T \boldsymbol{d}^{(k)} < 0$, then

- $f_k \le C_k \le \frac{1}{k+1} \sum_{j=0}^{(k)} f_j$
- there exists α_k satisfying the modified Wolfe or Armijo conditions

In addition, if ∇f is Lipschitz with constant L, then

- $\alpha_k > C \frac{|\nabla f_k^T \boldsymbol{d}^{(k)}|}{\|\boldsymbol{d}^{(k)}\|}$ for some constant depending on c_1, c_2, L and the backing factor

$d^k = -\nabla f(x^k)$

Furthermore, if for all sufficiently large k, we have uniform bounds

$$\nabla f_k^T \boldsymbol{d}^{(k)} \le -c_3 \|\nabla f_k\|^2 \quad \text{and} \quad \|\boldsymbol{d}^{(k)}\| \le c_4 \|\nabla f_k\|$$

then
- $\lim_{k \to \infty} \nabla f_k = 0$

Once again, pairing with non-monotone linear search, Barzilai-Borwein gradient methods *work every well on general unconstrained differentiable problems*.

References:

Yu-Hong Dai and Li-Zhi Liao. R-linear convergence of the Barzilai and Borwein gradient method. *IMA Journal of Numerical Analysis*, 22(1):1–10, 2002.

J. Barzilai and J.M. Borwein. Two-point step size gradient methods. *IMA Journal of Numerical Analysis*, 8(1):141–148, 1988.

Yu-Hong Dai. A new analysis on the barzilai-borwein gradient method. *Journal of the Operations Research Society of China*, pages 1–12, 2013.

Luigi Grippo, Francesco Lampariello, and Stephano Lucidi. A nonmonotone line search technique for Newton's method. *SIAM Journal on Numerical Analysis*, 23(4): 707–716, 1986.

Hongchao Zhang and William W Hager. A nonmonotone line search technique and its application to unconstrained optimization. *SIAM Journal on Optimization*, 14(4): 1043–1056, 2004.