Benefiting from Negative Curvature

Daniel P. Robinson
Johns Hopkins University
Department of Applied Mathematics and Statistics

Collaborator:
Frank E. Curtis (Lehigh University)

US and Mexico Workshop on Optimization and Its Applications
Huatulco, Mexico
January 8, 2018

Outline

Outline

Problem of interest: deterministic setting

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \; f(x)$$

- $f : \mathbb{R}^n \to \mathbb{R}$ assumed to be twice-continuously differentiable.
- L will denote the Lipschitz constant for ∇f
- σ will denote the Lipschitz constant for $\nabla^2 f$
- f may be nonconvex
- Notation:

$$g(x) := \nabla f(x)$$
$$H(x) := \nabla^2 f(x)$$

Much work has been done on convergence two second-order points:

- D. Goldfarb (1979) [6]
  - prove convergence result to second-order optimal points (unconstrained)
  - curvilinear search using descent direction and negative curvature direction
- D. Goldfarb, C. Mu, J. Wright, and C. Zhou (2017) [7]
  - consider equality constrained problems
  - prove convergence result to second-order optimal points
  - extend curvilinear search for unconstrained
- F. Facchinei and S. Lucidi (1998) [3]
  - consider inequality constrained problems
  - exact penalty function, directions of negative curvature, and line search
- P. Gill, V. Kungurtsev, and D. Robinson (2017) [4, 5]
  - consider inequality constrained problems
  - convergence to second-order optimal points under weak assumptions
- J. Moré and D. Sorensen (1979), A. Forsgren, P. Gill, and W. Murray (1995), and many more . . .

None consistently perform better by using directions of negative curvature!

Others hope to avoid saddle-points:

- J. Lee, M. Simchowich, M. Jordan, and B. Recht (2016) [8]
  - Gradient descent converges to local minimizer almost surely.
  - Uses random initialization.
- Y. Dauphin et al. (2016) [2]
  - Present a saddle-free Newton method (it is a modified-Newton method)
  - Goal is to escape saddle points (move away when close)

These (and others) try to avoid the ill-effects of negative curvature.

Purpose of this research:

- Design a method that consistently performs better by using directions of negative curvature.
- Do not try to avoid negative curvature. Use it!

Outline

Outline

Overview:

- Compute descent direction (s_k) and negative curvature direction (d_k).
- Predict which step will make more progress in reducing the objective f.
- If predicted decrease is not realized, adjust parameters.
- Iterate until an approximate second-order solution is obtained.

Requirements on the descent direction s_k

Compute s_k to satisfy

$$-g(x_k)^T s_k \geq \delta \|s_k\|_2 \|g(x_k)\|_2 \qquad \big(\text{some } \delta \in (0,1]\big)$$

Examples:

- $s_k = -g(x_k)$
- $B_k s_k = -g_k$ with B_k appropriately chosen

Requirements on the negative curvature direction d_k

Compute d_k to satisfy

$$\begin{aligned} d_k^T H(x_k) d_k &\leq \gamma \lambda_k \|d_k\|_2^2 < 0 \quad \big(\text{some } \gamma \in (0,1]\big) \\ g(x_k)^T d_k &\leq 0 \end{aligned}$$

Examples:

- $d_k = \pm v_k$ with (λ_k, v_k) being the left-most eigenpair of $H(x_k)$
- d_k a sufficiently accurate estimate of $\pm v_k$

How to use s_k and d_k?

- Use both in a curvilinear linesearch?
  - Often taints good descent directions by "poorly scaled" directions of negative curvature.
  - No consistent performance gains!
- Start using d_k only once $\|g(x_k)\|$ is “small"?
  - No consistent performance gains!
  - Misses areas of the space in which great decrease in f is possible.
- Use s_k when $\|g(x_k)\|$ is big relative to $|(\lambda_k)_-|$. Otherwise, use d_k?
  - Better, but still inconsistent performance gains!

We propose to use upper-bounding models. It works!

Predicted decrease along descent direction s_k

If $L_k \geq L$, then

$$f(x_k + \alpha s_k) \leq f(x_k) - m_{s,k}(\alpha) \qquad (\text{for all } \alpha)$$

with

$$m_{s,k}(\alpha) := -\alpha g(x_k)^T s_k - \tfrac{1}{2} L_k \alpha^2 \|s_k\|_2^2$$

and define the quantity

$$\alpha_k := \frac{-g(x_k)^T s_k}{L_k \|s_k\|_2^2} = \operatorname*{argmax}_{\alpha \geq 0} \; m_{s,k}(\alpha)$$

Comments

- $m_{s,k}(\alpha_k)$ is the best predicted decrease along s_k
- If $s_k = -g(x_k)$, then $\alpha_k = 1/L_k$

Predicted decrease along the negative curvature direction d_k

If $\sigma_k \geq \sigma$, then

$$f(x_k + \beta d_k) \leq f(x_k) - m_{d,k}(\beta) \qquad \big(\text{for all } \beta\big)$$

with

$$m_{d,k}(\beta) := -\beta g(x_k)^T d_k - \tfrac{1}{2}\beta^2 d_k^T H(x_k) d_k - \tfrac{\sigma_k}{6}\beta^3 \|d_k\|_2^3$$

and define, with $c_k := d_k^T H(x_k) d_k$, the quantity

$$\beta_k := \frac{\left(-c_k + \sqrt{c_k^2 - 2\sigma_k \|d_k\|_2^3 g(x_k)^T d_k}\right)}{\sigma_k \|d_k\|_2^3} = \operatorname*{argmax}_{\beta \geq 0} \; m_{d,k}(\beta)$$

Comments

- $m_{d,k}(\beta_k)$ is the best predicted decrease along d_k

Choose the step that predicts the largest decrease in f.

- If $m_{s,k}(\alpha_k) \geq m_{d,k}(\beta_k)$, then Try the step s_k
- If $m_{d,k}(\beta_k) > m_{s,k}(\alpha_k)$, then Try the step d_k

Question: Why “Try" instead of “Use"?
Answer: We do not know if $L_k \geq L$ and $\sigma_k \geq \sigma$

- If $L_k < L$, then it could be the case that

$$f(x_k + \alpha_k s_k) > f(x_k) - m_{s,k}(\alpha_k)$$

- If $\sigma_k < \sigma$, then it could be the case that

$$f(x_k + \beta_k d_k) > f(x_k) - m_{d,k}(\beta_k)$$

Dynamic Step-Size Algorithm

for $k \in \mathbb{N}$ **do**
  compute s_k and d_k satisfying the required step conditions
  loop
    compute $\alpha_k = \underset{\alpha \geq 0}{\operatorname{argmax}}\ m_{s,k}(\alpha)$ and $\beta_k = \underset{\beta \geq 0}{\operatorname{argmax}}\ m_{d,k}(\beta)$
    if $m_{s,k}(\alpha_k) \geq m_{d,k}(\beta_k)$ **then**
      if $f(x_k + \alpha_k s_k) \leq f(x_k) - m_{s,k}(\alpha_k)$ **then**
        set $x_{k+1} \leftarrow x_k + \alpha_k s_k$ and then exit loop
      else
        set $L_k \leftarrow \rho L_k$ $[\rho \in (1, \infty)]$
    else
      if $f(x_k + \beta_k d_k) \leq f(x_k) - m_{d,k}(\beta_k)$ **then**
        set $x_{k+1} \leftarrow x_k + \beta_k d_k$ and then exit loop
      else
        set $\sigma_k \leftarrow \rho \sigma_k$
  set $(L_{k+1}, \sigma_{k+1}) \in (L_{\min}, L_k] \times (\sigma_{\min}, \sigma_k]$

Outline

Key decrease inequality: For all $k \in \mathbb{N}$ it holds that

$$f(x_k) - f(x_{k+1}) \geq \max\left\{ \frac{\delta^2}{2L_k}\|g(x_k)\|_2^2, \frac{2\gamma^3}{3\sigma_k^2}|(\lambda_k)_-|^3 \right\}.$$

Comments:

- First term in the max holds when $x_{k+1} = x_k + \alpha_k s_k$.
- Second term in the max holds when $x_{k+1} = x_k + \beta_k d_k$.
- The above max holds because we choose whether to try s_k or d_k based on

$$m_{s,k}(\alpha_k) \geq m_{d,k}(\beta_k)$$

- Can prove that $\{L_k\}$ and $\{\sigma_k\}$ remain uniformly bounded.

Theorem (Limit points satisfy second-order necessary conditions)

The computed iterates satisfy

$$\lim_{k\to\infty} \|g(x_k)\|_2 = 0 \ \textit{and} \ \liminf_{k\to\infty} \lambda_k \geq 0$$

Theorem (Complexity result)

The number of iterations, function, and derivative (i.e., gradient and Hessian) evaluations required until some iteration $k \in \mathbb{N}$ is reached with

$$\|g(x_k)\|_2 \leq \epsilon_g \ \textit{and} \ |(\lambda_k)_-| \leq \epsilon_H$$

is at most

$$\mathcal{O}(\max\{\epsilon_g^{-2}, \epsilon_H^{-3}\})$$

Outline

Refined parameter increase strategy

$$\hat{L}_k \leftarrow L_k + \frac{2\big(f(x_k + \alpha_k s_k) - f(x_k) + m_{s,k}(\alpha_k)\big)}{\alpha_k^2 \|s_k\|^2}$$

$$\hat{\sigma}_k \leftarrow \sigma_k + \frac{6\big(f(x_k + \beta_k d_k) - f(x_k) + m_{d,k}(\beta_k)\big)}{\beta_k^3 \|d_k\|^3}$$

then, with $\rho \leftarrow 2$, use the update

$$L_k \leftarrow \max\{\rho L_k, \min\{10^3 L_k, \hat{L}_k\}\}$$

$$\sigma_k \leftarrow \max\{\rho \sigma_k, \min\{10^3 \sigma_k, \hat{\sigma}_k\}\}$$

Refined parameter decrease strategy

$$L_{k+1} \leftarrow \max\{10^{-3}, 10^{-3} L_k, \hat{L}_k\} \text{ and } \sigma_{k+1} \leftarrow \sigma_k \text{ when } x_{k+1} \leftarrow x_k + \alpha_k s_k$$

$$\sigma_{k+1} \leftarrow \max\{10^{-3}, 10^{-3} \sigma_k, \hat{\sigma}_k\} \text{ and } L_{k+1} \leftarrow L_k \text{ when } x_{k+1} \leftarrow x_k + \beta_k d_k$$

Termination condition

$$\|g(x_k)\| \leq 10^{-5} \max\{1, \|g(x_0)\|\} \text{ and } |(\lambda_k)_-| \leq 10^{-5} \max\{1, |(\lambda_0)_-|\}.$$

Measures of interest

- Final objective value:

$$\frac{f_{\text{final}}(s_k) - f_{\text{final}}(s_k, d_k)}{\max\{|f_{\text{final}}(s_k)|, |f_{\text{final}}(s_k, d_k)|, 1\}} \in [-1, 1]$$

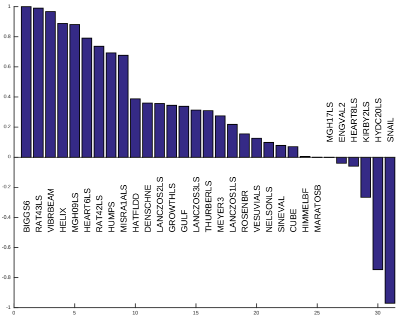
- Required number of iterations:

$$\frac{\#its(s_k) - \#its(s_k, d_k)}{\max\{\#its(s_k), \#its(s_k, d_k), 1\}} \in [-1, 1]$$

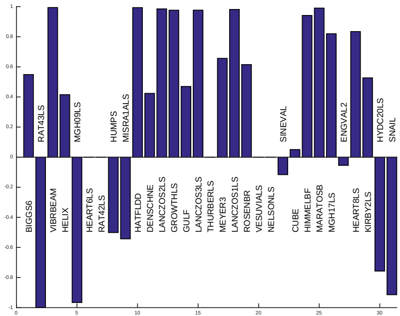
- Required number of function evaluations:

$$\frac{\#fevals(s_k) - \#fevals(s_k, d_k)}{\max\{\#fevals(s_k), \#fevals(s_k, d_k), 1\}} \in [-1, 1]$$

Steepest descent: $s_k = -g(x_k)$ and $d_k = \pm v_k$



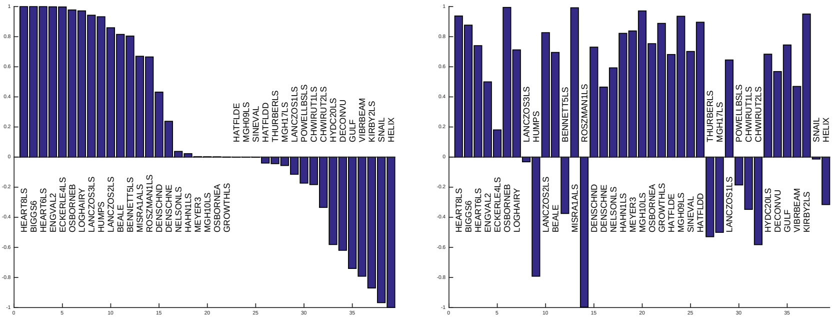
(a) Final objective value.



(b) Required number of iterations.

Figure: Only problems for which at least one negative curvature direction is used and the difference in final f-values is larger than 10^{-5} in absolute value are presented.

Shifted Newton: $B_k = H(x_k) + \delta_k I,\ B_k s_k = -g(x_k),$ and $d_k = \pm v_k$



(a) Final objective value. (b) Required number of iterations.

Figure: Only problems for which at least one negative curvature direction is used and the difference in final f-values is larger than 10^{-5} in absolute value are presented.

Outline

1. Motivation

2. Deterministic Setting
   - The Method
   - Convergence Results
   - Numerical Results
   - Comments

3. Stochastic Setting

Comments:

- If L and σ are known, do not need to ever update L_k and σ_k, in theory. In practice, still allow increase and decrease for efficiency.
- Currently, one function evaluation each trial step. If evaluating f is very cheap, could consider evaluating both trial steps during each iteration.
- Relevance to strict saddle points
  - We do not make any non-degenerate assumption.
  - Our convergence result holds regardless of the types of saddle points.
  - When the strict saddle point property holds, our theory implies that
    - Any limit point of the sequence $\{x_k\}$ is a minimizer of f.
    - Iterates eventually enter a region that only contains minimizers.
  - We get a stronger convergence theory (cf. Paternain, Mokhtari, and Ribeiro (2017)) because we incorporate directions of negative curvature.
- The complexity result for our method is not "optimal" based on a traditional complexity perspective.
- F. Curtis and I have been intrigued by alternate complexity perspectives:
  - Typically, results are for general problems and based on worst case.
  - From some perspective, the algorithm I presented today is "optimal".
  - See his talk later this afternoon!

Outline

Summary

- Apply same ideas as in the deterministic case, but in the mini-batch case.
- Add a negative curvature direction $d_k = \pm v_k$ with the sign chosen randomly. Can be thought of as a “smart noise" approach.
- Small gain in performance relative to similar algorithm without d_k.
- See our paper [1] for additional details.

References I

[1] F. E. Curtis and D. P. Robinson, Exploiting negative curvature directions in stochastic optimization, in http://arxiv.org/abs/1703.00412, Submitted to Mathematical Programming (Special Issue on Nonconvex Optimization for Statistical Learning), 2017.

[2] Y. N. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio, Identifying and attacking the saddle point problem in high-dimensional non-convex optimization, in Advances in Neural Information Processing Systems 27, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, eds., Curran Associates, Inc., 2014, pp. 2933–2941.

[3] F. Facchinei and S. Lucidi, Convergence to second order stationary points in inequality constrained optimization, Mathematics of Operations Research, 23 (1998), pp. 746–766.

References II

[4] P. E. Gill, V. Kungurtsev, and D. P. Robinson, A stabilized sqp method: global convergence, IMA Journal of Numerical Analysis, 37 (2017), pp. 407–443.

[5] ——, A stabilized sqp method: superlinear convergence, Mathematical Programming, 163 (2017), pp. 369–410.

[6] D. Goldfarb, Curvilinear path steplength algorithms for minimization which use directions of negative curvature, Mathematical programming, 18 (1980), pp. 31–40.

[7] D. Goldfarb, C. Mu, J. Wright, and C. Zhou, Using negative curvature in solving nonlinear programs, arXiv preprint arXiv:1706.00896, (2017).

References III

[8] J. D. Lee, M. Simchowitz, M. I. Jordan, and B. Recht, Gradient descent only converges to minimizers, in Conference on Learning Theory, 2016, pp. 1246–1257.