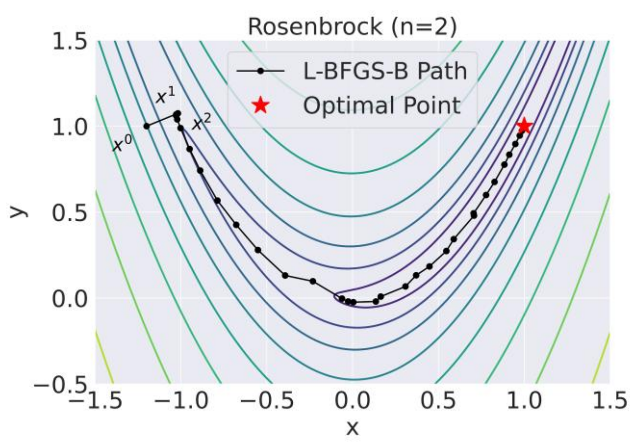
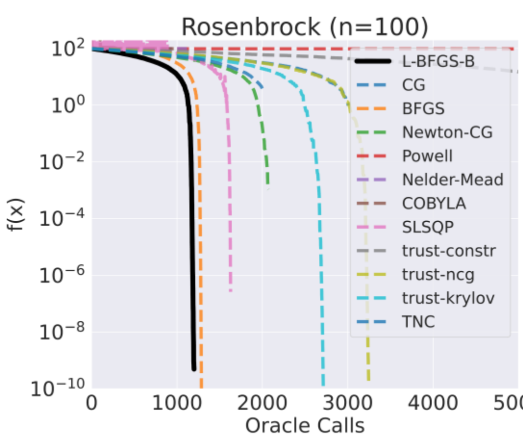
正則化と関数値非依存最適化による誤差耐性準ニュートン法の開発

浜口広樹, 丸茂直貴, 武田朗子 (東京大学)

研究背景: 準ニュートン法(L-BFGS法)と近似ヘッセ行列の正定値性

問題設定: $\min_{x\in\mathbb{R}_n} f(x)$ (目的関数は連続可微分)
$f(x), \nabla f(x)$が計算可能 (殆どの研究は誤差を考慮しない)
準ニュートン法: SciPy(著名 OSS) において、高速な最適化手法

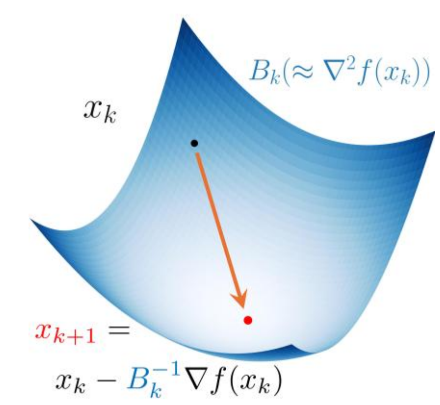




準ニュートン法は、近似ヘッセ行列を求めることが特徴:

$$x_{k+1} = x_k - \alpha_k B_k^{-1} \nabla f(x_k). \quad (\alpha_k \in \mathbb{R}_+)$$

- $B_k \approx \nabla^2 f(x_k) \in \mathbb{R}^{n\times n}$
- 二次近似による部分問題
- そのヘッセ行列が B_k である
- 解が唯一であるためにも、B_kは**正定値**な必要がある
- その解を x_{k+1}として反復



問題意識: 誤差による不十分な最適化と正則化準ニュートン法

- **勾配ノルム** $\|\nabla f(x)\|$が小さくなるまで最適化 (=: 問題を解く)
- 現実の最適化では計算機上で行うので、**数値誤差が発生**
  - 関数値差分 $f(x_k) - f(x_{k+1})$は勾配よりも**誤差に弱い**
  - 従来の準ニュートン法は**関数値差分の正しい評価に依拠**
- ベンチマーク問題 221 個 (CUTEst)にて、デフォルトのSciPy は 141 個の問題で**関数値差分の小ささに由来し停止**
  - うち 39 個はデフォルト以外でも、**本質的に解けない**
  - 単精度 (32bit) 浮動小数点数などビックデータでは顕著

従来の直線探索:

$$x_{k+1} = x_k - \alpha_k (B_k^{-1} \nabla f(x_k))$$

(B_kの正定値性が必須)

正則化準ニュートン法(Kanzow, 2023)

$$x_{k+1} = x_k - (B_k + \mu_k I)^{-1} \nabla f(x_k)$$

(B_kが正定値でなくともよい)

他にも複数の利点:
負定値で良いので失敗となる試行が少なく済む、ステップ間でシームレスなパラメータ調整が可能、非線形なパラメータ探索、 理論的な簡潔による拡張性など

本研究の貢献: 関数値非依存最適化による誤差の問題の本質的解決

Main Issue: 数値誤差に本質的に頑健な準ニュートン法は開発可能か?

Previous Work: 正則化準ニュートン法では近似ヘッセ行列に課せられる条件が緩い
この条件の緩さが不正確な関数値差分を許容し誤差耐性を生むが、本質的解決には至らず

Our Work: 正則化準ニュートン法を用いながら、数値誤差を陽に考慮する
関数値非依存最適化を援用し、関数値差分を部分的に回避

Keyword: 関数値非依存最適化 (OFFO) (Gratton and Toint, 2023)

機械学習分野での需要などから近年発展してきた手法
Adam や AdaGrad などの最適化手法は、勾配だけで最適化
誤差の文脈で言えば、関数値差分を利用せずに最適化が可能
近似ヘッセ行列の活用も可能で正則化準ニュートン法と好相性

提案手法の概略

探索方向は $d_k \leftarrow -(B_k + \mu_k I)^{-1} g_k$ (g_k は勾配)

正則化準ニュートン法

$$\mu_{k+1} \leftarrow \begin{cases} \sigma_1 \mu_k & \text{if success} \\ \mu_k & \text{if normal} \\ \sigma_2 \mu_k & \text{if failure} \end{cases}$$

μ_k: 正則化パラメータ
信頼領域法と類似した更新式
関数値差分をここでは利用

有限回
反復

関数値非依存最適化

$$\mu_{k+1} \leftarrow \theta \sqrt{(\mu_k/\theta)^2 + \|g_{k+1}\|^2}$$

誤差が大きくなった場合
AdaGrad-Norm 的更新式
関数値差分に依存しない

理論保証

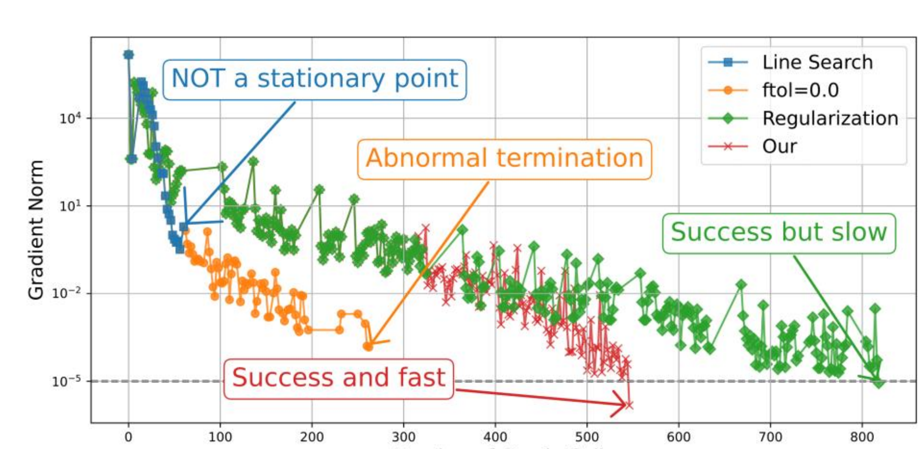
L-平滑と有界性の仮定の下、**標準的な収束率を証明可能**
→ 関数値に誤差がある状況下でも収束しうる点が利点

$$\text{average}_{0\le j\le k} \|g_j\|^2 = \mathcal{O}(1/k).$$

実験結果の概略

問題例:BDQRTIC

提案手法は**高速かつ安定に収束**



今後の課題

- 特に数値実験が未完であり、鋭意論文執筆中
- アルゴリズムの細部でどう詰められるか?
- 修正セカント条件を用いると高速化の余地