Curvature-Exploiting Acceleration of Elastic Net Computation

Vien V. Mai and Mikael Johansson
KTH - Royal Institute of Technology
The elastic net problem

Workhorse in ML and modern statistics

\[
\minimize_{x \in \mathbb{R}^d} \left\{ \frac{1}{2n} \|Ax - b\|_2^2 + \frac{\gamma_2}{2} \|x\|_2^2 + \gamma_1 \|x\|_1 \right\}
\]

Special instances: \(\gamma_1 = 0 \Rightarrow \text{Ridge regression}; \gamma_2 = 0 \Rightarrow \text{Lasso}\)

In many real-world data sets, Hessian of the smooth part

\[
\nabla^2 f(x) = \frac{1}{n} A^\top A + \gamma_2 I = C + \gamma_2 I
\]

has \textit{rapidly decaying spectrum}. 
Related work

Deterministic first-order methods:

- **PGD**: $O\left(dn\kappa \log \frac{1}{\epsilon}\right)$
- **FISTA**: $O\left(dn\sqrt{\kappa} \log \frac{1}{\epsilon}\right)$

\[
\kappa = \frac{\lambda_1(C + \gamma_2 I)}{\lambda_d(C + \gamma_2 I)}
\]

Stochastic first-order methods:

- **ProxSVRG**: $O\left(d(n + \tilde{\kappa}) \log \frac{1}{\epsilon}\right)$
- **Katyusha**: $O\left(d(n + \sqrt{n}\tilde{\kappa}) \log \frac{1}{\epsilon}\right)$

\[
\tilde{\kappa} = \frac{\text{tr}(C + \gamma_2 I)}{\lambda_d(C + \gamma_2 I)}
\]

**Challenge**: exploit second-order information despite non-smoothness.
Main contribution

Novel 2nd-order optimization algorithm computes \( \varepsilon \)-optimal solution in time

\[
\mathcal{O}(d(n + c \tilde{\kappa}) \log \frac{1}{\varepsilon})
\]

Stochastic first-order methods have \( c = 1 \), our method has

\[
c = \frac{r \lambda_r + \sum_{i > r} \lambda_i}{\sum_{i=1}^r \lambda_i + \sum_{i > r} \lambda_i} \ll 1
\]

Dramatic improvement when \( C \) has rapidly decaying spectrum
Proposed algorithm

Two building blocks:

1. Approximation of smooth Hessian using randomized block Lanczos

2. Proximal Newton method with stochastic gradients
   - Exploits finite-sum structure
   - Uses momentum acceleration to increase mini-batch size
   - Makes clever use of error control and warm start
Experimental results: suboptimality vs. iteration counts

![Graphs showing suboptimality vs. epoch for different datasets and optimization methods]

- **gisette-scale**
  - y-axis: Suboptimality
  - x-axis: Epoch
  - Different lines represent different optimization methods: FISTA, Katyusha, ProxSVRG, BCD, Ours

- **australian**
  - y-axis: Suboptimality
  - x-axis: Epoch
  - Different lines represent different optimization methods: FISTA, Katyusha, ProxSVRG, BCD, Ours

- **cina0**
  - y-axis: Suboptimality
  - x-axis: Epoch
  - Different lines represent different optimization methods: FISTA, Katyusha, ProxSVRG, BCD, Ours

- **real-sim**
  - y-axis: Suboptimality
  - x-axis: Epoch
  - Different lines represent different optimization methods: FISTA, Katyusha, ProxSVRG, BCD, Ours
Experimental results: suboptimality vs. run-times

- **gisette-scale, \( b = 500 \)**
  - Suboptimality vs. Time [s]
  - Log-log scale

- **australian**
  - Suboptimality vs. Time [s]
  - Log-log scale

- **cina0**
  - Suboptimality vs. Time [s]
  - Log-log scale

- **real-sim, \( b = 2000 \)**
  - Suboptimality vs. Time [s]
  - Log-log scale
Thank you!

Please come visit our poster at:

**Room Pacific Ballroom #196**

Code: https://github.com/vienmai/elasticnet