SCORE: Approximating Curvature Information under Self-Concordant Regularization

Adeyemi D. Adeoye & Alberto Bemporad *(IMT School for Advanced Studies, Lucca, Italy)*

IMT SCHOOL FOR ADVANCED STUDIES LUCCA

Problem

Data: $\{(\boldsymbol{x}_n, \boldsymbol{y}_n)\}_{n=1}^N, \boldsymbol{x}_n \in \mathbb{R}^{n_p}, \boldsymbol{y}_n \in \mathbb{R}^d$.
Objective:

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^{n_w}} \mathcal{L}(\boldsymbol{\theta}) \triangleq \underbrace{\sum_{n=1}^{N} \ell(\boldsymbol{y}_n, \hat{\boldsymbol{y}}_n)}_{f(\boldsymbol{\theta})} + \lambda \underbrace{\sum_{j=1}^{n_w} r_j(\boldsymbol{\theta}_j)}_{h(\boldsymbol{\theta})}. \quad (1)$$

- ℓ is the output-fit loss function.
- r_j, define a regularization term on $\boldsymbol{\theta}$, $\lambda > 0$.

Assumptions

- f, h are respectively γ_l, γ_a-strongly convex.
- f, h have γ_u, γ_b-Lipschitz continuous first derivatives $\boldsymbol{g}_f$, $\boldsymbol{g}_h$, respectively.
- f, h have γ_f, γ_h-Lipschitz continuous second derivatives $\boldsymbol{H}_f$, $\boldsymbol{H}_h$, respectively.
- λh is M_h-self-concordant:

$$\left|\boldsymbol{u}^T \left(\partial^3 h(\boldsymbol{\theta})[\boldsymbol{u}]\right) \boldsymbol{u}\right| \leq 2M_h \left(\boldsymbol{u}^T g_h \boldsymbol{u}\right)^{3/2},$$

for any $\boldsymbol{u} \in \mathbb{R}^{n_w}$, $M_h > 0$. Examples are the ℓ^2-norm and pseudo-Huber function.

Main Contributions

- We present the self-concordant regularization (SCORE) framework for (overparameterized) convex models of problem (1).
- SCORE extends the idea of *Newton decrement* in [1] (or its extension in [2]) for self-concordant functions, and requires only the regularization funtion to be self-concordant, which is less restrictive in practice.
- We employ a more general (than popular Woodbury) matrix identity to scale the generalized Gauss-Newton (GGN) computations.

Curvature Approximation

- Second-order (curvature) information in the data is highly desirable and highly expensive.

Newton update:

$$\underbrace{\boldsymbol{\theta}_{k+1} - \boldsymbol{\theta}_k}_{\delta\boldsymbol{\theta}} = -\rho(\underbrace{\boldsymbol{H}_f + \lambda \boldsymbol{H}_h}_{n_w \times n_w})^{-1}(\boldsymbol{g}_f + \boldsymbol{g}_h).$$

- Its GGN approximation is efficiently retrieved in SCORE.
- We choose a mini-batch size $m < N$ so that $dm + 1 < n_w$ (possibly $dm \ll n_w$).

$$\delta\boldsymbol{\theta} = -\rho \boldsymbol{H}_h^{-1} \boldsymbol{J}^T (\underbrace{\lambda \boldsymbol{I} + \boldsymbol{Q}\boldsymbol{J}\boldsymbol{H}_h^{-1}\boldsymbol{J}^T}_{(dm+1)\times(dm+1)})^{-1} \boldsymbol{e},$$

$$\boldsymbol{J} = \begin{bmatrix} \partial_{\boldsymbol{\theta}} \hat{\boldsymbol{y}} \\ \lambda \partial_{\boldsymbol{\theta}} r \end{bmatrix}, \boldsymbol{Q} = \begin{bmatrix} \partial^2_{\hat{\boldsymbol{y}}\hat{\boldsymbol{y}}} \ell & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \boldsymbol{e} = \begin{bmatrix} \partial_{\hat{\boldsymbol{y}}} \ell \\ \mathbf{1} \end{bmatrix}.$$

- $\boldsymbol{H}_h$ is always diagonal! ✓
- $\boldsymbol{Q}(1,1) \equiv \boldsymbol{I}$ if ℓ is the squared loss ✓

GGN with SCORE

- At each mini-batch sampling time $k \in [1, t_m]$, $t_m \triangleq \lceil \frac{N}{m} \rceil$, compute the update $\boldsymbol{\theta}_{t+1}$ using GGN-SCORE algorithm.

Algorithm 1 GGN-SCORE (mini-batch step)

1: **Input:** variables vector $\boldsymbol{\theta}_k$, data $\{(\boldsymbol{x}_n, \boldsymbol{y}_n)\}_{n=1}^m$, $\boldsymbol{H}_h$, $\boldsymbol{Q}$, $\boldsymbol{J}$, $\boldsymbol{e}$, parameters α_k, M_h, $\lambda > 0$
2: **Output:** variables vector $\boldsymbol{\theta}_{k+1}$
3: Compute $\boldsymbol{g}_h = \partial_{\boldsymbol{\theta}_k} h(\boldsymbol{\theta}_k)$
4: Choose $\eta_k = \left(\boldsymbol{g}_h^T \boldsymbol{H}_h^{-1} \boldsymbol{g}_h\right)^{1/2}$
5: Set $\rho_k = \frac{\alpha_k}{1 + M_h \eta_k}$
6: Set $\boldsymbol{G} = \boldsymbol{H}_h^{-1} \boldsymbol{J}^T \left(\lambda \boldsymbol{I} + \boldsymbol{Q}\boldsymbol{J}\boldsymbol{H}_h^{-1}\boldsymbol{J}^T\right)^{-1} \boldsymbol{e}$
7: Compute $\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \rho_k \boldsymbol{G}$

Key Remark

There exist $\beta, \tilde{\beta}, K_1 > 0$, with $\boldsymbol{Q} \leq K_1 \boldsymbol{I}$, such that $\|\boldsymbol{e}\| \leq \beta$, $\|\boldsymbol{J}\| \leq \tilde{\beta}$, and hence

$$\|\lambda \boldsymbol{I} + \boldsymbol{Q}\boldsymbol{J}\boldsymbol{H}_h^{-1}\boldsymbol{J}^T\| \leq \lambda + (K/\gamma_a), \qquad \boxed{K = K_1 \tilde{\beta}^2.}$$

Main Theorem (Local Convergence Rate)

If $\alpha_k = \alpha = \frac{\sqrt{\gamma_a}}{\beta_1}(K + \lambda\gamma_a)$, $\beta_1 = \beta\tilde{\beta}$, then the iterates of GGN-SCORE satisfy the following descent properties:

$$\mathbb{E}[\mathcal{L}(\boldsymbol{\theta}_{k+1})] \leq \mathcal{L}(\boldsymbol{\theta}_k) - \left(\frac{\lambda\omega(\zeta_k)}{M_h^2} + \frac{\gamma_l \omega''(\tilde{\zeta}_k)}{2\gamma_a} - \xi\right),$$

$$\mathbb{E}\|\boldsymbol{\theta}_{k+1} - \boldsymbol{\theta}^*\|_{\boldsymbol{\theta}_{k+1}} \leq \vartheta \|\boldsymbol{\theta}_k - \boldsymbol{\theta}^*\|_{\boldsymbol{\theta}_k} + \frac{\gamma_u}{\beta_1}\|\boldsymbol{\theta}_k - \boldsymbol{\theta}^*\| + \frac{\gamma_g}{2}\|\boldsymbol{\theta}_k - \boldsymbol{\theta}^*\|^2,$$

$$\omega(t) \triangleq t - \ln(1+t), \quad \zeta_k \triangleq \frac{M_h}{1 + M_h \eta_k},$$

$$\tilde{\zeta}_k \triangleq M_h \eta_k, \quad \xi \triangleq \frac{2(\gamma_u + \lambda\gamma_b)}{\sqrt{\gamma_a}},$$

$$\vartheta \triangleq 1 + \frac{\lambda}{\sqrt{\gamma_a}\beta_1(1 - M_h\|\boldsymbol{\theta}_k - \boldsymbol{\theta}^*\|_{\boldsymbol{\theta}_k})}.$$

Experimental Results

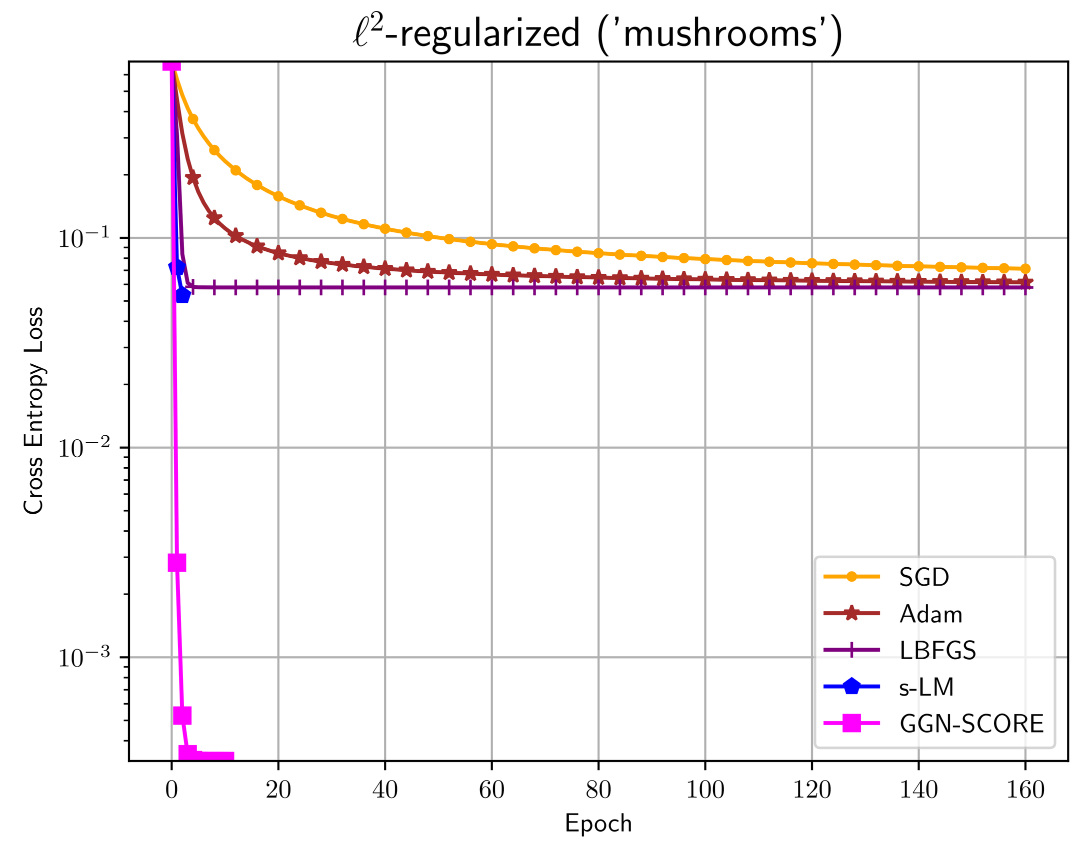

Figure 1: Convex, "overparameterized"

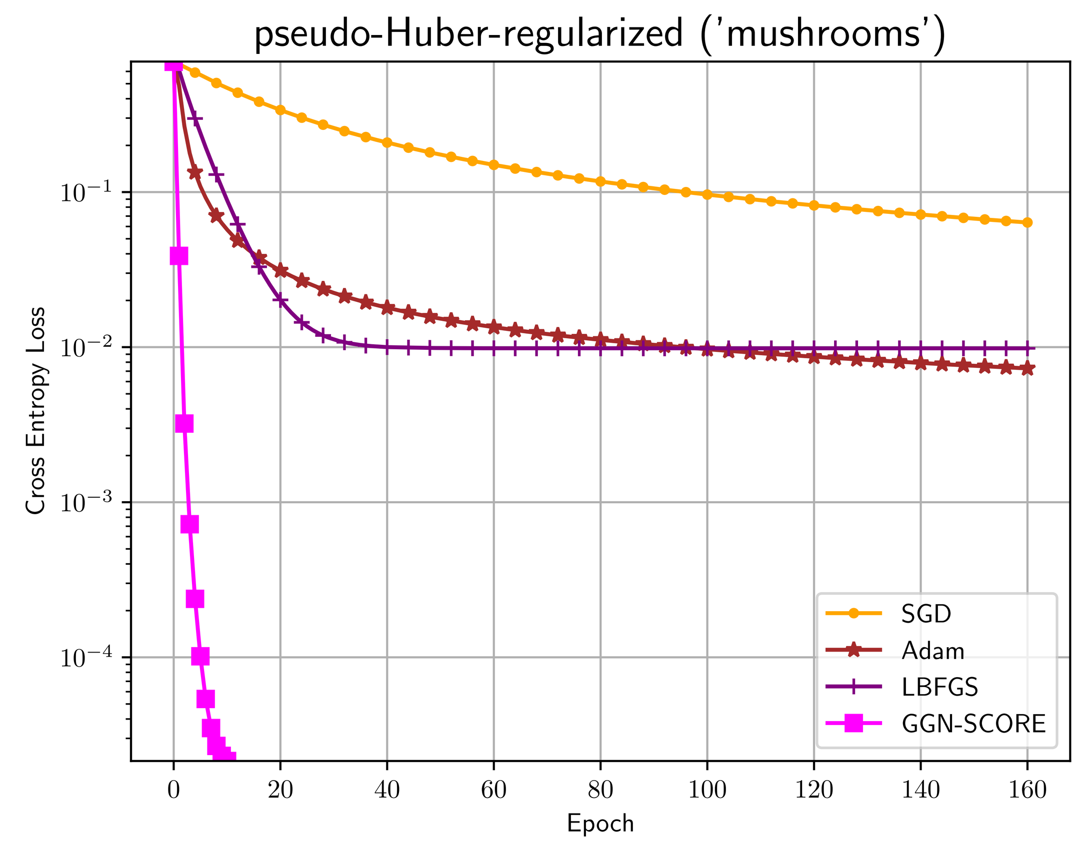

Figure 2: Nonconvex, "overparameterized"

References

[1] Yurii Nesterov and Arkadii Nemirovskii. *Interior-point polynomial algorithms in convex programming*. SIAM, 1994.

[2] Wenbo Gao and Donald Goldfarb. Quasi-newton methods: superlinear convergence without line searches for self-concordant functions. *Optimization Methods and Software*, 34(1):194–217, 2019.

Contact Information

Web adeyemiadeoye.github.io
Email adeyemi.adeoye@imtlucca.it