Composite Self-concordant Minimization

Quoc Tran Dinh, Anastasios Kyrillidis and Volkan Cevher

Laboratory for Information and Inference Systems (LIONS)

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

lions@epfl

FNSNF FONDS NATIONAL SUISSE SCHWEIZERISCHER NATIONALFONDS FONDO NAZIONALE SVIZZERO SWISS NATIONAL SCIENCE FOUNDATION

erc

MARIE CURIE ACTIONS

Abstract

We propose an algorithmic framework for convex minimization problems of composite functions with two terms: a self-concordant part and a possibly nonsmooth regularization part. Our method is a new proximal Newton algorithm with local quadratic convergence rate. As a specific problem instance, we consider sparse precision matrix estimation problems in graph learning. Via a careful dual formulation and a novel analytic step-size selection, we instantiate an algorithm within our framework for graph learning that avoids *Cholesky decompositions and matrix inversions*, making it attractive for parallel and distributed implementations.

Composite minimization

$$\text{(P)} \qquad \min_{\mathbf{x}\in\mathbb{R}^n} \{F(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x})\}$$

f: convex and smooth — **g**: convex and possibly nonsmooth

Motivation

Problem (P) *covers many practical problems*:

- ***Unconstrained basic LASSO/Basic pursuit denoising problems***
- ***Graphical model selections*** / latent variable graphical model selection
- **Poisson imaging reconstruction** / Heteroscedastic LASSO
- Low-rank approximation
- Clustering

Problem (P) *has specific separable structure*

- ***splitting methods***
- ***augmented Lagrangian methods***

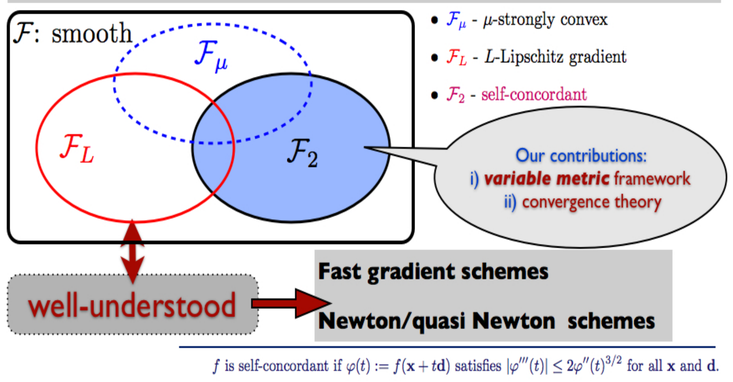
g: ℓ_1-norm, nuclear norm or indicator functions

Composite self-concordant minimization

$$\text{(P)} \qquad \min_{\mathbf{x}\in\mathbb{R}^n} \{F(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x})\}$$

f: convex and self-concordant — **g**: convex and possibly nonsmooth

Classes of smooth functions (f)



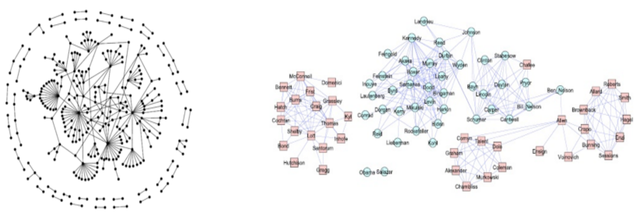
f is self-concordant if $\varphi(t) := f(\mathbf{x}+t\mathbf{d})$ satisfies $|\varphi'''(t)| \le 2\varphi''(t)^{3/2}$ for all $\mathbf{x}$ and $\mathbf{d}$.

Example: Log-determinant for LMIs

Optimization problem

$$\min_{\Theta\succ 0}\left\{\underbrace{-\log\det(\Theta)+\operatorname{trace}(\Sigma\Theta)}_{f(\mathbf{x})} + \underbrace{\rho\|\operatorname{vec}(\Theta)\|_1}_{g(\mathbf{x})}\right\}$$

- ***Application***: Graphical model selection



Log-barrier for linear/quadratic inequalities

- Poisson imaging reconstruction

$$x^* \in \operatorname{argmin}\left\{\underbrace{\sum_{i=1}^m a_i^T x - \sum_{i=1}^m y_i \log(a_i^T x + b_i)}_{f(x)} + g(x)\right\}$$

- Basic pursuit denoising problem (BPDP): Barrier formulation

$$\mathbf{x}_t^* = \operatorname*{argmin}_{\mathbf{x}}\left\{\underbrace{-t\log\left(\sigma^2 - \|\mathbf{A}\mathbf{x}-\mathbf{y}\|_2^2\right)}_{=:f(\mathbf{x})} + g(\mathbf{x})\right\}$$

- LASSO problem with unknown variance

$$\mathbf{x}^* \equiv (\phi^*,\gamma^*) = \operatorname*{argmin}_{\phi,\gamma}\left\{\underbrace{-\log(\gamma)+\frac{1}{2n}\|\gamma\mathbf{y}-\mathbf{X}\phi\|_2^2}_{=:f(\mathbf{x})} + \underbrace{\lambda\|\phi\|_1}_{=:g(\mathbf{x})}\right\}$$

Prior state-of-the-art

- Proximal point scheme with variable metric [Bonnans, 1993]

Proximal point scheme with variable metric (PPA(H))

Given x^0, generate a sequence $\{x^k\}_{k\ge 0}$ such that

$$x^{k+1} = \operatorname{prox}_{H_k}\left(x^k - H_k^{-1}\nabla f(x^k)\right)$$

where H_k is symmetric positive definite

- Special cases
  - If $H_k := \gamma_k I$ then we obtain a proximal-gradient scheme [Nesterov2007, Beck2009]
  - If $H_k \approx \nabla^2 f(x^k)$ then we obtain a proximal-quasi Newton scheme [Becker2012]
  - If $H_k = \nabla^2 f(x^k)$ then we obtain a proximal-Newton scheme [Lee2012]
  - If $\operatorname{prox}_{H_k} \equiv \mathrm{Id}$ then we obtain a classical variable metric descent scheme
- Connection to splitting
  - PPA(H) is a special case of splitting forward-backward methods

Self-concordance: A mathematical tool

- **Local norm** — f **is a self-concordant function**

	Primal norm	Dual norm
Local norm	$\|\mathbf{u}\|_{\mathbf{x}} = [\mathbf{u}^T\nabla^2 f(\mathbf{x})\mathbf{u}]^{1/2}$	$\|\mathbf{u}\|_{\mathbf{x}}^* = [\mathbf{u}^{T-1}\nabla^2 f(\mathbf{x})\mathbf{u}]^{1/2}$
Utility functions	$\omega(t) := t - \ln(1+t)$	$\omega_*(t) := -t - \ln(1-t)$

- **Main properties**

Lower approximation	$f(\mathbf{y}) \ge f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y}-\mathbf{x}) + \omega(\|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}})$	$\mathbf{x},\mathbf{y}\in\operatorname{dom}(f)$
Upper approximation	$f(\mathbf{y}) \le f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y}-\mathbf{x}) + \omega^*(\|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}})$	$\|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}} < 1$
Hessian approximation	$(1-\|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}})^2\nabla^2 f(\mathbf{x}) \preceq \nabla^2 f(\mathbf{y}) \preceq (1-\|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}})^{-2}\nabla^2 f(\mathbf{x})$	$\|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}} < 1$

Local

- **Compare to the main properties of $\mathcal{F}_{\mu,L}$**

Lower approximation	$f(\mathbf{y}) \ge f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y}-\mathbf{x}) + \frac{\mu}{2}\|\mathbf{y}-\mathbf{x}\|_2^2$	$\mathbf{x},\mathbf{y}\in\operatorname{dom}(f)$
Upper approximation	$f(\mathbf{y}) \le f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y}-\mathbf{x}) + \frac{L}{2}\|\mathbf{y}-\mathbf{x}\|_2^2$	$\mathbf{x},\mathbf{y}\in\operatorname{dom}(f)$
Hessian approximation	$\mu\mathbb{I} \preceq \nabla^2 f(\mathbf{x}) \preceq L\mathbb{I}$	$\mathbf{x},\mathbf{y}\in\operatorname{dom}(f)$

Our approach

- Proximal Newton scheme

Given $\mathbf{x}^0$, generate a sequence $\{\mathbf{x}^k\}_{k\ge 0}$ such that

$$\mathbf{x}^{k+1} := \mathbf{x}^k + \alpha_k \mathbf{d}_{\mathbf{H}_k}^k$$

where $\alpha_k \in (0,1]$ is step-size, $\mathbf{d}_{\mathbf{H}_k}^k$ is a search direction

- How to calculate the Proximal Newton direction?

$$\mathbf{d}_{\mathbf{H}_k}^k := \arg\min_{\mathbf{d}}\left\{f(\mathbf{x}^k) + \nabla f(\mathbf{x}^k)^T\mathbf{d} + \frac{1}{2}\mathbf{d}^T\mathbf{H}_k\mathbf{d} + g(\mathbf{x}^k+\mathbf{d})\right\}, \quad \mathbf{H}_k = \nabla^2 f(\mathbf{x}^k)$$

- Highlights

- **Globalization**: a new strategy for finding step-size explicitly, motivate "*forward-looking*" line-search strategy
- **Search direction**: efficient (strongly convex program)
- **Local convergence**: quadratic convergence *without boundedness of the Hessian* analytic quadratic convergence region

How do we compute the step-size?

- Upper approximation of f

$$f(\mathbf{x}^{k+1}) \le f(\mathbf{x}^k) + \nabla f(\mathbf{x}^k)^T(\mathbf{x}^{k+1}-\mathbf{x}^k) + \omega^*(\|\mathbf{x}^{k+1}-\mathbf{x}^k\|_{\mathbf{x}^k}), \quad \|\mathbf{x}^{k+1}-\mathbf{x}^k\|_{\mathbf{x}^k} < 1$$

- Convexity of g and optimality condition of the subproblem

$$g(\mathbf{x}^{k+1}) - g(\mathbf{x}^k) \le -\alpha_k\nabla f(\mathbf{x}^k)^T\mathbf{d}_{\mathbf{H}_k}^k - \alpha_k\|\mathbf{d}_{\mathbf{H}_k}^k\|_{\mathbf{H}_k}^2.$$

$$\phi(\mathbf{x}^{k+1}) \le \phi(\mathbf{x}^k) - \alpha_k\|\mathbf{d}_{\mathbf{H}_k}^k\|_{\mathbf{H}_k}^2 + \omega_*\left(\alpha_k\|\mathbf{d}_{\mathbf{H}_k}^k\|_{\mathbf{x}^k}\right)$$

- When $\mathbf{H}_k \equiv \nabla^2 f(\mathbf{x}^k)$, $\lambda_k := \|\mathbf{d}_{\mathbf{H}_k}^k\|_{\mathbf{x}^k}$

$$\phi(\mathbf{x}^{k+1}) \le \phi(\mathbf{x}^k) - \underbrace{\left[\alpha_k\lambda_k - \omega_*\left(\alpha_k\lambda_k\right)\right]}_{\psi(\alpha_k)}$$

maximize $\psi(\alpha_k)$ to get optimal α_k^*

$$\alpha_k^* = \frac{1}{\lambda_k+1} \in (0,1]$$

- Line-search on $[\alpha_k^*, 1]$ to accelerate the global convergence

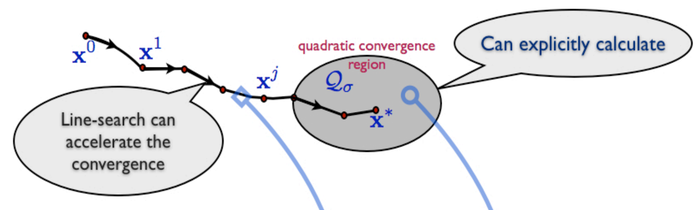
Analytic complexity

Proximal Newton decrement: $\lambda_k := \|\mathbf{d}^k\|_{\mathbf{x}^k}$

Quadratic convergence region: Let $\sigma := (5-\sqrt{17})/4 \approx 0.219224$

$$\mathcal{Q}_\sigma := \{\mathbf{x}^k \mid \lambda_k \le \sigma\}$$

Illustration the convergence behavior



- Worst-case complexity

$$\#\text{iterations} = \left\lfloor\frac{F(\mathbf{x}^0)-F(\mathbf{x}^*)}{0.021}\right\rfloor + O\left(\ln\ln\left(\frac{c}{\varepsilon}\right)\right)$$

Application to graphical model selection

- The underlying convex optimization problem

$$\min_{\Theta\succ 0}\left\{\underbrace{-\log\det(\Theta)+\operatorname{trace}(\Sigma\Theta)}_{f(\mathbf{x})} + \underbrace{\rho\|\operatorname{vec}(\Theta)\|_1}_{g(\mathbf{x})}\right\}$$

- Gradient and Hessian (large-scale, special structure)
  - Gradient of f: $\nabla f(\mathbf{x}) = \mathbf{vec}(\Sigma - \Theta^{-1})$.
  - Hessian of f: $\nabla^2 f(\mathbf{x}) = \Theta^{-1}\otimes\Theta^{-1}$
- Dual approach for solving subproblem (SP)

Unconstrained LASSO problems — No Cholesky decomposition and matrix inversion

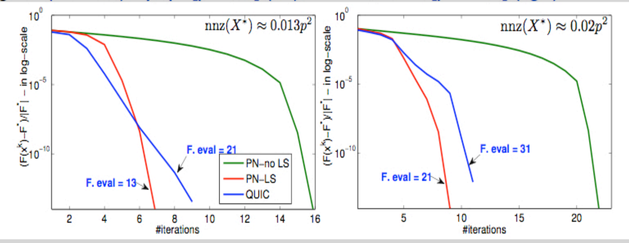
Primal subproblem	Dual subproblem (SPGL)
$\min_{\Delta}\left\{\frac{1}{2}\operatorname{trace}((\Theta_i^{-1}\Delta)^2) + \operatorname{trace}(\mathbf{R}_i\Delta) + \rho\|\operatorname{vec}(\Delta)\|_1\right\}$	$\min_{\|\operatorname{vec}(\mathbf{U})\|_\infty\le 1}\left\{\frac{1}{2}\operatorname{trace}((\Theta_i\mathbf{U})^2) + \operatorname{trace}(\mathbf{Q}_i\mathbf{U})\right\}$
$\mathbf{R}_i := \Sigma - 2\Theta_i^{-1}$	$\mathbf{Q}_i := \rho^{-1}[\Theta_i\Sigma\Theta_i - 2\Theta_i]$

- How to compute proximal Newton decrement $\lambda_i := \|\mathbf{d}^i\|_{\mathbf{x}^i}$?

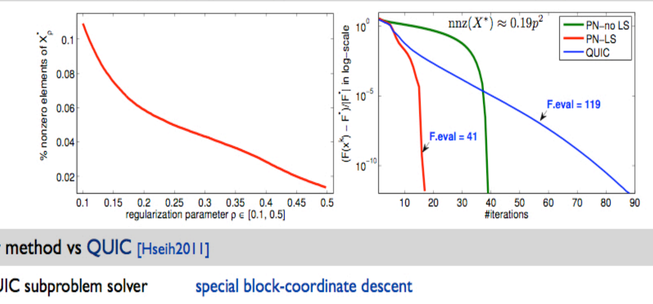
$$\lambda_i := [p - 2\operatorname{trace}(\mathbf{W}_i) + \operatorname{trace}(\mathbf{W}_i^2)]^{1/2}, \quad \mathbf{W}_i = \Theta_i(\Sigma - \rho\mathbf{U}^*)$$

Numerical experiments on graph learning

Convergence (rho = 0.5): Lymph [p = 587] (left), Leukemia [p = 1255] (right)



Sparsity w.r.t. rho for Lymph (left), Convergence for Lymph with rho = 0.05 (right)

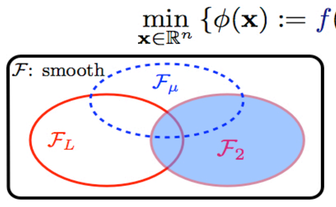


Our method vs QUIC [Hseih2011]

- QUIC subproblem solver: special block-coordinate descent
- Our subproblem solver: general proximal algorithms

Conclusions

$$\min_{\mathbf{x}\in\mathbb{R}^n}\{\phi(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x})\}$$



- $\mathcal{F}_\mu$ - μ-strongly convex
- $\mathcal{F}_L$ - L-Lipschitz gradient
- $\mathcal{F}_2$ - self-concordant

- Highlights

- **Globalization**: a new strategy for finding step-size **explicitly** motivate "*forward-looking*" line-search strategy
- **Search direction**: efficient (strongly convex program)
- **Local convergence**: quadratic convergence *without boundedness of the Hessian* **analytic** quadratic convergence region

- Practical contributions
  - software package has quasi-Newton, pure-Newton, gradient and fast gradient algorithms
  - leverage fast proximal solvers for g(x) (structured norms etc.)
  - robust to subproblem solver accuracy

Software

- A open-source software package called **SCOPT** is available at http://lions.epfl.ch/software
- **SCOPT** stands for **S**elf-**C**oncordant **OPT**imization