Mathematics of Data: From Theory to Computation

Prof. Volkan Cevher
volkan.cevher@epfl.ch

Lecture 9: Composite convex minimization II

Laboratory for Information and Inference Systems (LIONS)
École Polytechnique Fédérale de Lausanne (EPFL)

EE-556 (Fall 2015)

License Information for Mathematics of Data Slides

Outline

- Today
  1. Proximal Newton-type methods.
  2. Composite self-concordant minimization
- Next week
  1. Sourse separation
  2. Convex geometry of linear inverse problems

Recommended reading material

- A. Beck and M. Tebulle, A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems, SIAM J. Imaging Sciences, 2(1), 183–202, 2009.
- Y. Nesterov, Smooth minimization of non-smooth functions, Math. Program, 103(1), 127–152, 2005.
- Q. Tran-Dinh, A. Kyrillidis and V. Cevher, Composite Self-Concordant Minimization, LIONS-EPFL Tech. Report. `http://arxiv.org/abs/1308.2867`, 2013.
- N. Parikh and S. Boyd, Proximal Algorithms, Foundations and Trends in Optimization, 1(3):123-231, 2014.

Motivation

Motivation

Data analytics problems in various disciplines can often be simplified to nonsmooth **composite convex minimization** problems. To this end, this lecture provides **efficient numerical solution methods** for such problems.

Intriguingly, composite minimization problems are far from generic nonsmooth problems and we can exploit individual function structures to obtain numerical solutions nearly as efficiently as if they are smooth problems.

Composite convex minimization

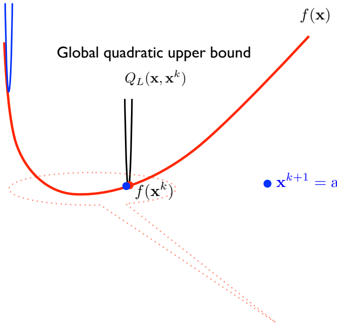
Problem (Unconstrained composite convex minimization)

$$F^\star := \min_{\mathbf{x}\in\mathbb{R}^p} \{F(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x})\} \tag{1}$$

- *f and g are both proper, closed, and convex.*
- *dom(F) := dom$(f) \cap$ dom$(g) \neq \emptyset$ and $-\infty < F^\star < +\infty$.*
- *The solution set $\mathcal{S}^\star := \{\mathbf{x}^\star \in \text{dom}(F) : F(\mathbf{x}^\star) = F^\star\}$ is nonempty.*

How can we better adapt to the local geometry?

Non-adaptive:

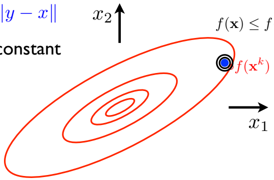


- $\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}} \left\{ f(\mathbf{x}^k) + \langle \nabla f(\mathbf{x}^k), \mathbf{x} - \mathbf{x}^k \rangle + \frac{L}{2} \|\mathbf{x} - \mathbf{x}^k\|_2^2 \right\}$

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|y - x\|$$

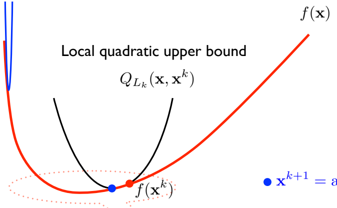
L is a global worst-case constant

$$f(\mathbf{x}) \leq f(\mathbf{x}^k) + \nabla f(\mathbf{x}^k)^T(\mathbf{x} - \mathbf{x}^k) + \frac{L}{2}\|\mathbf{x} - \mathbf{x}^k\|_2^2$$



How can we better adapt to the local geometry?

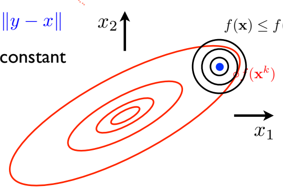
Line-search:



- $\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}} \left\{ f(\mathbf{x}^k) + \langle \nabla f(\mathbf{x}^k), \mathbf{x} - \mathbf{x}^k \rangle + \frac{L_k}{2} \|\mathbf{x} - \mathbf{x}^k\|_2^2 \right\}$

$\|\nabla f(x) - \nabla f(y)\| \leq L\|y - x\|$

L is a global worst-case constant

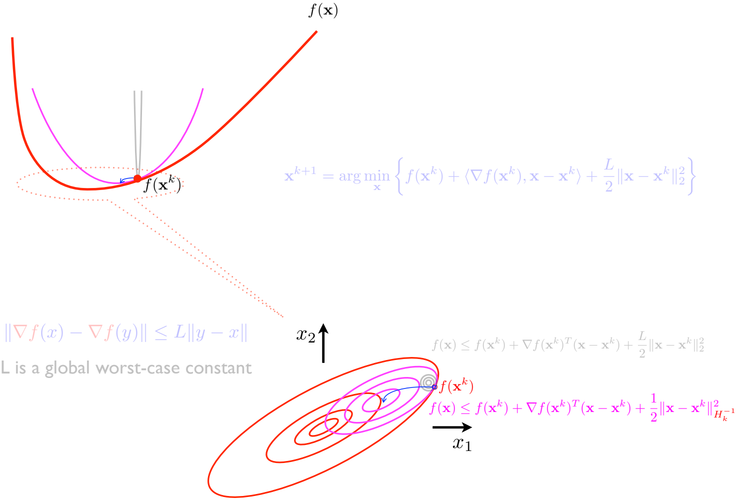


$f(\mathbf{x}) \leq f(\mathbf{x}^k) + \nabla f(\mathbf{x}^k)^T(\mathbf{x} - \mathbf{x}^k) + \frac{L}{2}\|\mathbf{x} - \mathbf{x}^k\|_2^2$

applies only locally

How can we better adapt to the local geometry?

Variable metric:



The idea of the proximal-Newton method

Assumptions A.2

Assume that $f \in \mathcal{F}^{2,1}_{L,\mu}(\mathbb{R}^p)$ and $g \in \mathcal{F}_{\mathrm{prox}}(\mathbb{R}^p)$.

The idea of the proximal-Newton method

Assumptions A.2

Assume that $f \in \mathcal{F}^{2,1}_{L,\mu}(\mathbb{R}^p)$ and $g \in \mathcal{F}_{\text{prox}}(\mathbb{R}^p)$.

The idea of proximal-Newton method

- Under Assumptions A.2, we can linearize the smooth term of the optimality condition of (1): $0 \in \nabla f(\mathbf{x}^\star) + \partial g(\mathbf{x}^\star)$ as

$$0 \in \nabla f(\mathbf{x}^\star) + \partial g(\mathbf{x}^\star) \approx \nabla f(\mathbf{x}^k) + \nabla^2 f(\mathbf{x}^k)^T(\mathbf{x}^\star - \mathbf{x}^k) + \partial g(\mathbf{x}^\star).$$

- Similar to the classical Newton method in Lecture 3, we can generate an iterative sequence $\{\mathbf{x}^k\}_{k\geq 0}$ by solving the **inclusion**:

$$0 \in \nabla f(\mathbf{x}^k) + \nabla^2 f(\mathbf{x}^k)^T(\mathbf{x} - \mathbf{x}^k) + \partial g(\mathbf{x}) \tag{2}$$

  to obtain $\mathbf{x}^{k+1}$.

- The last condition is equivalent to

$$\mathbf{x}^{k+1} := \arg\min_{\mathbf{x}\in\mathbb{R}^p}\Big\{\frac{1}{2}(\mathbf{x}-\mathbf{x}^k)^T\nabla^2 f(\mathbf{x}^k)(\mathbf{x}-\mathbf{x}^k) + \nabla f(\mathbf{x}^k)^T(\mathbf{x}-\mathbf{x}^k) + g(\mathbf{x})\Big\}. \tag{3}$$

Proximal-Newton-type scheme

- The sequence $\{\mathbf{x}^k\}$ generated by (3) is not necessarily convergent. Hence, a sufficient descent condition is required.
- We can replace $\nabla^2 f(\mathbf{x}^k)$ by a given approximate matrix $\mathbf{H}_k$.

Proximal-Newton-type scheme

- The sequence $\{\mathbf{x}^k\}$ generated by (3) is not necessarily convergent. Hence, a sufficient descent condition is required.
- We can replace $\nabla^2 f(\mathbf{x}^k)$ by a given approximate matrix $\mathbf{H}_k$.

Proximal-quasi-Newton-type algorithms:

- Let $\mathbf{H}_k \approx \nabla^2 f(\mathbf{x}^k)$ be a symmetric positive definite (SDP) matrix. From (2), we have

$$\mathbf{x}^k - \mathbf{H}_k^{-1}\nabla f(\mathbf{x}^k) \in (\mathbb{I} + \mathbf{H}_k^{-1}\partial g)(\mathbf{x}),$$

  which leads to

$$\mathbf{x}^{k+1} := \mathrm{prox}_{\mathbf{H}_k^{-1}g}\left(\mathbf{x}^k - \mathbf{H}_k^{-1}\nabla f(\mathbf{x}^k)\right). \quad (4)$$

- By letting $\mathbf{d}^k := \mathbf{x}^{k+1} - \mathbf{x}^k$, (4) is equivalent to

$$\mathbf{d}^k := \arg\min_{\mathbf{d}\in\mathbb{R}^p} \left\{\frac{1}{2}\mathbf{d}^T\mathbf{H}_k\mathbf{d} + \nabla f(\mathbf{x}^k)^T\mathbf{d} + g(\mathbf{x}^k + \mathbf{d})\right\}. \quad (5)$$

  Then $\mathbf{d}^k$ is called a proximal-Newton-type direction.
- Proximal-Newton-type algorithm generates a sequence $\{\mathbf{x}^k\}_{k\geq 0}$ starting from $\mathbf{x}^0 \in \mathbb{R}^p$ and update:

$$\mathbf{x}^{k+1} := \mathbf{x}^k + \alpha_k\mathbf{d}^k, \quad (6)$$

  where $\mathbf{d}^k$ is given by (5) and $\alpha_k \in (0, 1]$ is a damped step-size.

How to find step size α_k?

Lemma (Descent lemma [5])

Let $\mathbf{x}^k(\alpha) := \mathbf{x}^k + \alpha\mathbf{d}^k$ *for sufficiently small* $\alpha \in (0,1]$ *and* $\mathbf{H}_k \succ 0$. *Then, we have:*

$$F(\mathbf{x}^k(\alpha)) \leq F(\mathbf{x}^k) - (1/2)\alpha(\mathbf{d}^k)^T\mathbf{H}_k\mathbf{d}^k + \mathcal{O}(\alpha^2).$$

How to find step size α_k?

Lemma (Descent lemma [5])

Let $\mathbf{x}^k(\alpha) := \mathbf{x}^k + \alpha\mathbf{d}^k$ for sufficiently small $\alpha \in (0, 1]$ and $\mathbf{H}_k \succ 0$. Then, we have:

$$F(\mathbf{x}^k(\alpha)) \leq F(\mathbf{x}^k) - (1/2)\alpha(\mathbf{d}^k)^T\mathbf{H}_k\mathbf{d}^k + \mathcal{O}(\alpha^2).$$

Since $\mathbf{H}_k \succ 0$, this lemma tells us that:

- If $\mathbf{d}^k \neq 0$, then there exists $\alpha > 0$ such that $F(\mathbf{x}^k(\alpha)) < F(\mathbf{x}^k)$.
- The value of α can be computed via **backtracking line search**.
- If $\mathbf{d}^k = 0$, then we can easily check that $\mathbf{x}^k$ is a solution of (1).

How to find step size α_k?

Lemma (Descent lemma [5])

Let $\mathbf{x}^k(\alpha) := \mathbf{x}^k + \alpha\mathbf{d}^k$ *for sufficiently small* $\alpha \in (0, 1]$ *and* $\mathbf{H}_k \succ 0$*. Then, we have:*

$$F(\mathbf{x}^k(\alpha)) \leq F(\mathbf{x}^k) - (1/2)\alpha(\mathbf{d}^k)^T\mathbf{H}_k\mathbf{d}^k + \mathcal{O}(\alpha^2).$$

Since $\mathbf{H}_k \succ 0$, this lemma tells us that:

- If $\mathbf{d}^k \neq 0$, then there exists $\alpha > 0$ such that $F(\mathbf{x}^k(\alpha)) < F(\mathbf{x}^k)$.
- The value of α can be computed via **backtracking line search**.
- If $\mathbf{d}^k = 0$, then we can easily check that $\mathbf{x}^k$ is a solution of (1).

Backtracking line-search

- Let
$$r_k := \nabla f(\mathbf{x}^k)^T\mathbf{d}^k + g(\mathbf{x}^k + \mathbf{d}^k) - g(\mathbf{x}^k).$$
- Find the smallest integer number $j \geq 0$ such that $\alpha_k := \beta^j$ and
$$F(\mathbf{x}^k + \alpha_k\mathbf{d}^k) \leq F(\mathbf{x}^k) + c\alpha_k r_k, \tag{7}$$
where $c \in (0, 0.5]$ and $\beta \in (0, 1)$ are two given constants (e.g., $c = 0.1$ and $\beta = 0.5$).

The proximal-Newton-type algorithm

We can summary the **proximal-Newton-type method** as follows:

Proximal-Newton algorithm (PNA)

1. Given $\mathbf{x}^0 \in \mathbb{R}^p$ as a starting point. Choose $c := 0.1$ and $\beta := 0.5$

2. For $k = 0, 1, \cdots$, perform the following steps:

2.1. Evaluate an SDP matrix $\mathbf{H}_k \approx \nabla^2 f(\mathbf{x}^k)$ and $\nabla f(\mathbf{x}^k)$.

2.2. Compute $\mathbf{d}^k := \mathrm{prox}_{\mathbf{H}_k^{-1} g}\left(\mathbf{x}^k - \mathbf{H}_k^{-1}\nabla f(\mathbf{x}^k)\right) - \mathbf{x}^k$.

2.3. Find the smallest integer number $j \geq 0$ such that

$$F(\mathbf{x}^k + \beta^j \mathbf{d}^k) \leq F(\mathbf{x}^k) + c\beta^j r_k$$

and set $\alpha_k := \beta^j$.

2.4. Update $\mathbf{x}^{k+1} := \mathbf{x}^k + \alpha_k \mathbf{d}^k$.

The proximal-Newton-type algorithm

We can summary the **proximal-Newton-type method** as follows:

Proximal-Newton algorithm (PNA)

1. Given $\mathbf{x}^0 \in \mathbb{R}^p$ as a starting point. Choose $c := 0.1$ and $\beta := 0.5$

2. For $k = 0, 1, \cdots$, perform the following steps:

2.1. Evaluate an SDP matrix $\mathbf{H}_k \approx \nabla^2 f(\mathbf{x}^k)$ and $\nabla f(\mathbf{x}^k)$.

2.2. Compute $\mathbf{d}^k := \mathrm{prox}_{\mathbf{H}_k^{-1} g}\left(\mathbf{x}^k - \mathbf{H}_k^{-1}\nabla f(\mathbf{x}^k)\right) - \mathbf{x}^k$.

2.3. Find the smallest integer number $j \geq 0$ such that

$$F(\mathbf{x}^k + \beta^j \mathbf{d}^k) \leq F(\mathbf{x}^k) + c\beta^j r_k$$

and set $\alpha_k := \beta^j$.

2.4. Update $\mathbf{x}^{k+1} := \mathbf{x}^k + \alpha_k \mathbf{d}^k$.

- If $\mathbf{H}_k \equiv \nabla^2 f(\mathbf{x}^k)$, then **PNA** becomes a pure proximal-Newton algorithm.
- If $\mathbf{H}_k \approx \nabla^2 f(\mathbf{x}^k)$, then **PNA** becomes a proximal-quasi-Newton algorithm.
- Main computation is Step 2.2, which requires a generalized prox-operator: $\mathrm{prox}_{\mathbf{H}_k^{-1} g}\left(\mathbf{x}^k + \mathbf{H}_k^{-1}\nabla f(\mathbf{x}^k)\right)$.
- Let $g(\mathbf{x}) = \rho\|\mathbf{x}\|_1$. When $\mathbf{H}_k$ is not diagonal, the cost is the same as solving an ℓ_1-regularized least squares, otherwise it is simply soft thresholding.

Convergence analysis

Assumption A.3.

- Problem (1): $\min_{\mathbf{x}}\{F(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x})\}$ admits a solution $\mathbf{x}^\star$.
- The subproblem $\mathrm{prox}_{\mathbf{H}_k^{-1} g}\left(\mathbf{x}^k + \mathbf{H}_k^{-1}\nabla f(\mathbf{x}^k)\right)$ is solved exactly for all $k \geq 0$.

Convergence analysis

Assumption A.3.

- Problem (1): $\min_{\mathbf{x}}\{F(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x})\}$ admits a solution $\mathbf{x}^\star$.
- The subproblem $\mathrm{prox}_{\mathbf{H}_k^{-1} g}\left(\mathbf{x}^k + \mathbf{H}_k^{-1}\nabla f(\mathbf{x}^k)\right)$ is solved exactly for all $k \geq 0$.

Theorem (Global convergence [5])

Assumptions:

- *The sequence $\{\mathbf{x}^k\}_{k\geq 0}$ is generated by PNA.*
- *Assumption A.3. is satisfied.*
- *There exists $\mu > 0$ such that $\mathbf{H}_k \succeq \mu\mathbb{I}$ for all $k \geq 0$.*

Conclusion:

- $\{\mathbf{x}^k\}_{k\geq 0}$ *globally converges to a solution* $\mathbf{x}^\star$ *of* (1).

- We have not yet obtained a **global convergence rate** of proximal-Newton methods.

Convergence analysis

Assumption A.3.

- Problem (1): $\min_{\mathbf{x}}\{F(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x})\}$ admits a solution $\mathbf{x}^\star$.
- The subproblem $\mathrm{prox}_{\mathbf{H}_k^{-1}g}\left(\mathbf{x}^k + \mathbf{H}_k^{-1}\nabla f(\mathbf{x}^k)\right)$ is solved exactly for all $k \geq 0$.

Theorem (Local convergence [5])

Assumptions:

- *The sequence $\{\mathbf{x}^k\}_{k\geq 0}$ is generated by PNA.*
- *Assumption A.3. is satisfied.*
- *Exist $0 < \mu \leq L_2 < +\infty$ such that $\mu\mathbb{I} \preceq \mathbf{H}_k \preceq L_2\mathbb{I}$ for all sufficiently large k.*

Conclusion:

- *If $\mathbf{H}_k \equiv \nabla^2 f(\mathbf{x}^k)$, then $\alpha_k = 1$ for k sufficiently large (full-step).*
- *If $\mathbf{H}_k \equiv \nabla^2 f(\mathbf{x}^k)$, then $\{\mathbf{x}^k\}$ locally converges to $\mathbf{x}^\star$ at a quadratic rate.*
- *If $\mathbf{H}_k$ satisfies the Dennis-Moré condition:*

$$\lim_{k\to+\infty} \frac{\|(\mathbf{H}_k - \nabla^2 f(\mathbf{x}^\star))(\mathbf{x}^{k+1} - \mathbf{x}^k)\|}{\|\mathbf{x}^{k+1} - \mathbf{x}^k\|} = 0, \tag{8}$$

then $\{\mathbf{x}^k\}$ locally converges to $\mathbf{x}^\star$ at a super linear rate.

How to compute the approximation $\mathbf{H}_k$?

- Solving $\mathrm{prox}_{\mathbf{H}_k^{-1} g}\left(\mathbf{x}^k + \mathbf{H}_k^{-1}\nabla f(\mathbf{x}^k)\right)$ exactly for a non-diagonal matrix $\mathbf{H}_k$ is impractical.
- This problem is solved iteratively by using, e.g., FISTA except for the special cases of $\mathbf{H}_k$.

How to compute the approximation $\mathbf{H}_k$?

- Solving $\mathrm{prox}_{\mathbf{H}_k^{-1} g}\left(\mathbf{x}^k + \mathbf{H}_k^{-1}\nabla f(\mathbf{x}^k)\right)$ exactly for a non-diagonal matrix $\mathbf{H}_k$ is impractical.
- This problem is solved iteratively by using, e.g., FISTA except for the special cases of $\mathbf{H}_k$.

How to update $\mathbf{H}_k$?

Matrix $\mathbf{H}_k$ can be updated by using low-rank updates.

- **BFGS update**: maintain the **Dennis-Moré condition** and $\mathbf{H}_k \succ 0$.

$$\mathbf{H}_{k+1} := \mathbf{H}_k + \frac{\mathbf{y}_k \mathbf{y}_k^T}{\mathbf{s}_k^T \mathbf{y}_k} - \frac{\mathbf{H}_k \mathbf{s}_k \mathbf{s}_k^T \mathbf{H}_k}{\mathbf{s}_k^T \mathbf{H}_k \mathbf{s}_k}, \quad \mathbf{H}_0 := \gamma \mathbb{I}, \ (\gamma > 0).$$

  where $\mathbf{y}_k := \nabla f(\mathbf{x}^{k+1}) - \nabla f(\mathbf{x}^k)$ and $\mathbf{s}_k := \mathbf{x}^{k+1} - \mathbf{x}^k$.
- **Diagonal+Rank-1 [2]**: computing PN direction $\mathbf{d}^k$ is in polynomial time, but it does not maintain the Dennis-Moré condition:

$$\mathbf{H}_k := \mathbf{D}_k + \mathbf{u}_k \mathbf{u}_k^T, \quad \mathbf{u}_k := (\mathbf{s}_k - \mathbf{H}_0 \mathbf{y}_k) / \sqrt{(\mathbf{s}_k - \mathbf{H}_0 \mathbf{y}_k)^T \mathbf{y}_k},$$

  where $\mathbf{D}_k$ is a positive diagonal matrix.

Advantages and disadvantages

Advantages

- PNA has fast local convergence rate (super-linear or quadratic)
- Numerical robustness under the inexactness/noise (inexact proximal-Newton method [5]).
- Quasi-Newton method is useful if the evaluation of $\nabla^2 f$ is expensive.
- Suitable for problems with many data points but few parameters. For example, problems of the form:

$$F^* := \min_{\mathbf{x}\in\mathbb{R}^p} \left\{ \sum_{j=1}^{n} \ell_j(\mathbf{a}_j^T\mathbf{x} + b_j) + g(\mathbf{x}) \right\},$$

where ℓ_j is twice continuously differentiable and convex, $g \in \mathcal{F}_{\text{prox}}$, $p \ll n$.

Advantages and disadvantages

Advantages

- PNA has fast local convergence rate (super-linear or quadratic)
- Numerical robustness under the inexactness/noise (inexact proximal-Newton method [5]).
- Quasi-Newton method is useful if the evaluation of $\nabla^2 f$ is expensive.
- Suitable for problems with many data points but few parameters. For example, problems of the form:

$$F^* := \min_{\mathbf{x}\in\mathbb{R}^p} \left\{ \sum_{j=1}^{n} \ell_j(\mathbf{a}_j^T\mathbf{x} + b_j) + g(\mathbf{x}) \right\},$$

where ℓ_j is twice continuously differentiable and convex, $g \in \mathcal{F}_{\text{prox}}$, $p \ll n$.

Disadvantages

- Expensive iteration compared to proximal-gradient methods.
- Global convergence rate may be worse than accelerated proximal-gradient methods.
- Requires a good initial point to get fast local convergence, which is hard to find.
- Requires strict conditions for global/local convergence analysis.

Example 1: Sparse logistic regression

Problem (**Sparse logistic regression**)

Given a sample vector $\mathbf{a} \in \mathbb{R}^p$ *and a binary class label vector* $\mathbf{b} \in \{-1, +1\}^n$*. The conditional probability of a label* b *given* $\mathbf{a}$ *is defined as:*

$$\mathbb{P}(b|\mathbf{a}, \mathbf{x}, \mu) = 1/(1 + e^{-b(\mathbf{x}^T\mathbf{a}+\mu)}),$$

where $\mathbf{x} \in \mathbb{R}^p$ *is a weight vector,* μ *is called the intercept.*
Goal: *Find a sparse-weight vector* $\mathbf{x}$ *via the maximum likelihood principle.*

Example 1: Sparse logistic regression

Problem (Sparse logistic regression)

Given a sample vector $\mathbf{a} \in \mathbb{R}^p$ *and a binary class label vector* $\mathbf{b} \in \{-1, +1\}^n$*. The conditional probability of a label* b *given* $\mathbf{a}$ *is defined as:*

$$\mathbb{P}(b|\mathbf{a}, \mathbf{x}, \mu) = 1/(1 + e^{-b(\mathbf{x}^T\mathbf{a}+\mu)}),$$

where $\mathbf{x} \in \mathbb{R}^p$ *is a weight vector,* μ *is called the intercept.*
Goal: *Find a sparse-weight vector* $\mathbf{x}$ *via the maximum likelihood principle.*

Optimization formulation

$$\min_{\mathbf{x}\in\mathbb{R}^p} \Bigg\{ \underbrace{\frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(b_i(\mathbf{a}_i^T\mathbf{x} + \mu))}_{f(\mathbf{x})} + \underbrace{\rho\|\mathbf{x}\|_1}_{g(\mathbf{x})} \Bigg\}, \tag{9}$$

where $\mathbf{a}_i$ is the i-th row of data matrix $\mathbf{A}$ in $\mathbb{R}^{n\times p}$, $\rho > 0$ is a regularization parameter, and ℓ is the logistic loss function $\mathcal{L}(\tau) := \log(1 + e^{-\tau})$.

Example: Sparse logistic regression

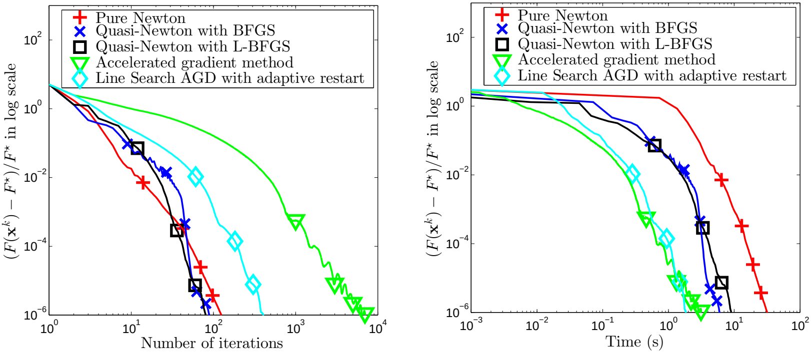
Real data

- Real data: w2a with $n = 3470$ data points, $p = 300$ features
- Available at http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html.

Parameters

- Tolerance 10^{-6}.
- L-BFGS memory $m = 50$.
- Ground truth: Get a high accuracy approximation of $\mathbf{x}^\star$ and $f^\star$ by TFOCS with tolerance 10^{-12}.

Example: Sparse logistic regression-Numerical results



Example 2: ℓ_1-regularized least squares

Problem (ℓ_1-regularized least squares)

Given $\mathbf{A} \in \mathbb{R}^{n \times p}$ *and* $\mathbf{b} \in \mathbb{R}^n$, *solve:*

$$F^\star := \min_{\mathbf{x} \in \mathbb{R}^p} \left\{ F(\mathbf{x}) := \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \rho\|\mathbf{x}\|_1 \right\}, \qquad (10)$$

where $\rho > 0$ *is a regularization parameter.*

Complexity per iterations

- Evaluating $\nabla f(\mathbf{x}^k) = \mathbf{A}^T(\mathbf{A}\mathbf{x}^k - \mathbf{b})$ requires one $\mathbf{A}\mathbf{x}$ and one $\mathbf{A}^T\mathbf{y}$.
- One soft-thresholding operator $\mathrm{prox}_{\lambda g}(\mathbf{x}) = \mathrm{sign}(\mathbf{x}) \otimes \max\{|\mathbf{x}| - \rho, 0\}$.
- **Optional**: Evaluating $L = \|\mathbf{A}^T\mathbf{A}\|$ (spectral norm) - via **power iterations** (e.g., 20 iterations, each iteration requires one $\mathbf{A}\mathbf{x}$ and one $\mathbf{A}^T\mathbf{y}$).

Example 2: ℓ_1-regularized least squares

Problem (ℓ_1-regularized least squares)

Given $\mathbf{A} \in \mathbb{R}^{n \times p}$ *and* $\mathbf{b} \in \mathbb{R}^n$, *solve:*

$$F^\star := \min_{\mathbf{x} \in \mathbb{R}^p} \left\{ F(\mathbf{x}) := \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \rho\|\mathbf{x}\|_1 \right\}, \tag{10}$$

where $\rho > 0$ *is a regularization parameter.*

Complexity per iterations

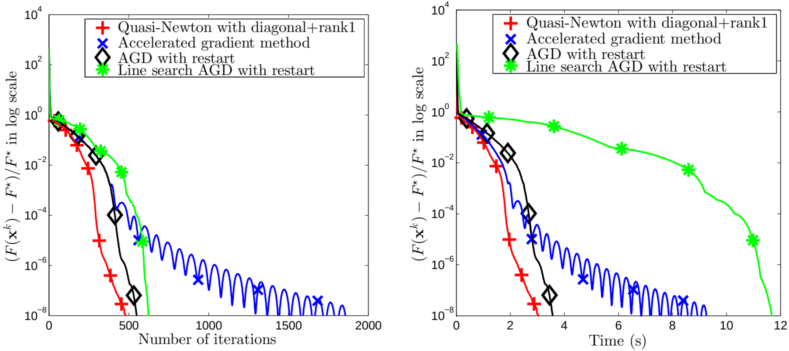
- Evaluating $\nabla f(\mathbf{x}^k) = \mathbf{A}^T(\mathbf{A}\mathbf{x}^k - \mathbf{b})$ requires one $\mathbf{A}\mathbf{x}$ and one $\mathbf{A}^T\mathbf{y}$.
- One soft-thresholding operator $\mathrm{prox}_{\lambda g}(\mathbf{x}) = \mathrm{sign}(\mathbf{x}) \otimes \max\{|\mathbf{x}| - \rho, 0\}$.
- **Optional**: Evaluating $L = \|\mathbf{A}^T\mathbf{A}\|$ (spectral norm) - via **power iterations** (e.g., 20 iterations, each iteration requires one $\mathbf{A}\mathbf{x}$ and one $\mathbf{A}^T\mathbf{y}$).

Synthetic data generation

- $\mathbf{A} := \mathrm{randn}(n, p)$ - standard Gaussian $\mathcal{N}(0, \mathbb{I})$.
- $\mathbf{x}^\star$ is a s-sparse vector generated randomly.
- $\mathbf{b} := \mathbf{A}\mathbf{x}^\star + \mathcal{N}(0, 10^{-3})$.

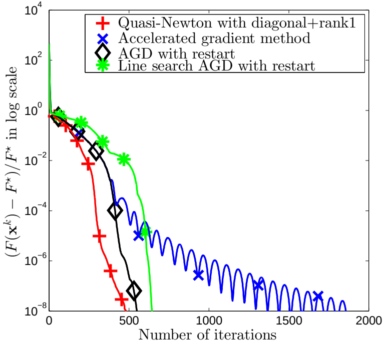
Example 2: ℓ_1-regularized least squares - Numerical results - Trial 1

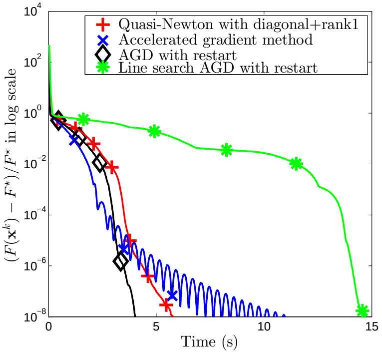
Parameters: $n = 750, p = 2000, s = 200, \rho = 1$



Example 2: ℓ_1-regularized least squares - Numerical results - Trial 2

Parameters: $n = 750, p = 2000, s = 200, \rho = 1$





Outline

- Today
  1. Proximal Newton-type methods.
  2. Composite self-concordant minimization
- Next week
  1. Sourse separation
  2. Convex geometry of linear inverse problems

Composite self-concordant minimization

Composite self-concordant minimization (CSM) problem [11]

$$F^\star := \min_{\mathbf{x}\in\mathrm{dom}(F)} \Big\{ F(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x}) \Big\}, \tag{11}$$

- $f \in \mathcal{F}_2(\mathrm{dom}(f))$ - self-concordant on $\mathrm{dom}(f) := \{\mathbf{x} \in \mathbb{R}^p \ : \ f(\mathbf{x}) < +\infty\}$
- $g \in \mathcal{F}_{\mathrm{prox}}(\mathbb{R}^p)$
- $\mathrm{dom}(F) := \mathrm{dom}(f) \cap \mathrm{dom}(g)$

Composite self-concordant minimization

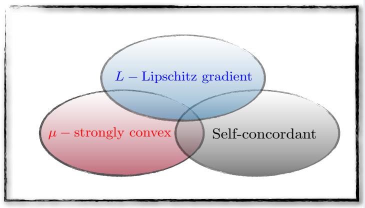
Composite self-concordant minimization (CSM) problem [11]

$$F^\star := \min_{\mathbf{x}\in\mathrm{dom}(F)} \left\{ F(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x}) \right\}, \tag{11}$$

- $f \in \mathcal{F}_2(\mathrm{dom}(f))$ - self-concordant on $\mathrm{dom}(f) := \{\mathbf{x} \in \mathbb{R}^p \ : \ f(\mathbf{x}) < +\infty\}$
- $g \in \mathcal{F}_{\text{prox}}(\mathbb{R}^p)$
- $\mathrm{dom}(F) := \mathrm{dom}(f) \cap \mathrm{dom}(g)$

Why is composite self-concordant minimization?

- A self-concordant function is not necessarily Lipschitz gradient.



- Covers many well-known examples.

Self-concordant functions in higher dimensions

Definition (Self-concordant functions [7, 6])

- A function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be self-concordant with parameter $M \geq 0$ if

$$|\varphi'''(t)| \leq M\varphi''(t)^{3/2},$$

  where $\varphi(t) := f(\mathbf{x} + t\mathbf{v})$ for all $t \in \mathbb{R}$, $\mathbf{x} \in \text{dom}(f)$ and $\mathbf{v} \in \mathbb{R}^n$ and $\mathbf{x} + t\mathbf{v} \in \text{dom}(f)$.
- When $M = 2$, the function f is said to be a standard self-concordant.

Self-concordant functions in higher dimensions

Definition (Self-concordant functions [7, 6])

- A function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be self-concordant with parameter $M \geq 0$ if

$$|\varphi'''(t)| \leq M\varphi''(t)^{3/2},$$

where $\varphi(t) := f(\mathbf{x} + t\mathbf{v})$ for all $t \in \mathbb{R}$, $\mathbf{x} \in \text{dom}(f)$ and $\mathbf{v} \in \mathbb{R}^n$ and $\mathbf{x} + t\mathbf{v} \in \text{dom}(f)$.
- When $M = 2$, the function f is said to be a standard self-concordant.

Example

The function $f(x) = -\log x$ is self-concordant. To see this, observe:

$$f''(x) = 1/x^2, \quad f'''(x) = -2/x^3.$$

Thus:

$$\frac{|f'''(x)|}{2f''(x)^{3/2}} = \frac{2/x^3}{2(1/x^2)^{3/2}} = 1$$

Self-concordant functions in higher dimensions

Definition (Self-concordant functions [7, 6])

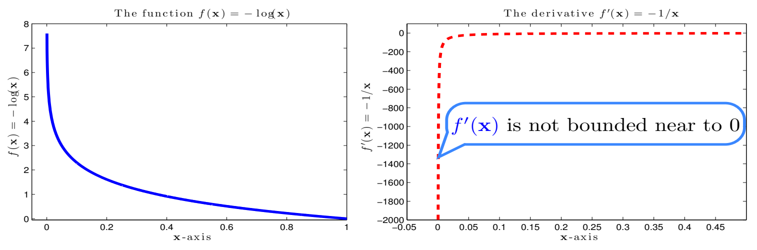
- A function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be self-concordant with parameter $M \geq 0$ if

$$|\varphi'''(t)| \leq M\varphi''(t)^{3/2},$$

where $\varphi(t) := f(\mathbf{x} + t\mathbf{v})$ for all $t \in \mathbb{R}$, $\mathbf{x} \in \mathrm{dom}(f)$ and $\mathbf{v} \in \mathbb{R}^n$ and $\mathbf{x} + t\mathbf{v} \in \mathrm{dom}(f)$.

- When $M = 2$, the function f is said to be a standard self-concordant.

$f(\mathbf{x}) = -\log(\mathbf{x})$ and its derivative $f'(\mathbf{x})$



Self-concordant functions in higher dimensions

Definition (Self-concordant functions [7, 6])

- A function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be self-concordant with parameter $M \geq 0$ if

$$|\varphi'''(t)| \leq M\varphi''(t)^{3/2},$$

where $\varphi(t) := f(\mathbf{x} + t\mathbf{v})$ for all $t \in \mathbb{R}$, $\mathbf{x} \in \mathrm{dom}(f)$ and $\mathbf{v} \in \mathbb{R}^n$ and $\mathbf{x} + t\mathbf{v} \in \mathrm{dom}(f)$.
- When $M = 2$, the function f is said to be a standard self-concordant.

Example

Similarly, the following example functions are self-concordant

1. $f(x) = x \log x - \log x$,
2. $f(x) = \sum_{i=1}^m \log(b_i - \mathbf{a}_i^T \mathbf{x})$ with domain $\mathrm{dom}(f) = \left\{\mathbf{x} \ : \ \mathbf{a}_i^T \mathbf{x} < b_i, i = 1, \ldots, m\right\}$,
3. $f(\mathbf{X}) = -\log\det(\mathbf{X})$ with domain $\mathrm{dom}(f) = \mathbb{S}_n^{++}$,
4. $f(\mathbf{x}) = -\log\left(\mathbf{x}^T\mathbf{P}\mathbf{x} + \mathbf{q}^T\mathbf{x} + r\right)$ with domain $\mathrm{dom}(f) = \left\{\mathbf{x} \ : \ \mathbf{x}^T\mathbf{P}\mathbf{x} + \mathbf{q}^T\mathbf{x} + r > 0\right\}$ and $-\mathbf{P} \in \mathbb{S}_n^{++}$.

Two well-known examples

Graphical model selection

$$\min_{\Theta \succ 0} \left\{ \underbrace{\mathrm{tr}(\Sigma\Theta) - \log\det(\Theta)}_{f(\mathbf{x})} + \underbrace{\rho\|\mathrm{vec}(\Theta)\|_1}_{g(\mathbf{x})} \right\} \tag{12}$$

where $\Theta \succ 0$ means that Θ is symmetric and positive definite and $\rho > 0$ is a regularization parameter and vec is the vectorization operator.

Two well-known examples

Graphical model selection

$$\min_{\Theta \succ 0} \Big\{ \underbrace{\mathrm{tr}(\Sigma\Theta) - \log\det(\Theta)}_{f(\mathbf{x})} + \underbrace{\rho\|\mathrm{vec}(\Theta)\|_1}_{g(\mathbf{x})} \Big\} \tag{12}$$

where $\Theta \succ 0$ means that Θ is symmetric and positive definite and $\rho > 0$ is a regularization parameter and vec is the vectorization operator.

Poisson imaging reconstruction (with TV-norm regularizer)

$$\min_{\mathbf{x} \in \mathbb{R}^{n \times p}} \Big\{ \underbrace{\sum_{i=1}^{n} (\mathbf{K}\mathbf{x})_i - \sum_{i=1}^{n} y_i \log((\mathbf{K}\mathbf{x})_i)}_{f(\mathbf{x})} + \underbrace{\rho\|\mathbf{x}\|_{\mathrm{TV}}}_{g(\mathbf{x})} \Big\} \tag{13}$$

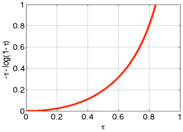
- $\mathbf{K}$ is a linear operator, $\mathbf{y} = (y_1, \ldots, y_n)^T \in \mathbb{Z}_+^n$ is the observed vector of photon counts.
- $\rho > 0$ is a regularization parameter,
- $\|\mathbf{x}\|_{\mathrm{TV}}$ is the TV-norm of $\mathbf{x}$ (see the above example).

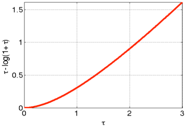
Some geometric intuition behind self-concordant functions

Local norm

Local norm: $\|\mathbf{u}\|_{\mathbf{x}} := \left[\mathbf{u}^T \nabla^2 f(\mathbf{x}) \mathbf{u}\right]^{1/2}$

Utility functions: $\omega_*(\tau) = -\tau - \ln(1-\tau),\ \tau \in [0,1)$ $\qquad$ $\omega(\tau) = \tau - \ln(1+\tau),\ \tau \geq 0$



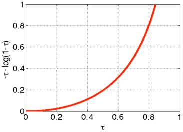


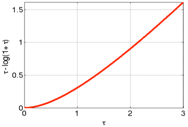
Some geometric intuition behind self-concordant functions

Local norm

Local norm: $\|\mathbf{u}\|_{\mathbf{x}} := \left[\mathbf{u}^T \nabla^2 f(\mathbf{x}) \mathbf{u}\right]^{1/2}$

Utility functions: $\omega_*(\tau) = -\tau - \ln(1-\tau),\ \tau \in [0,1)$ $\quad$ $\omega(\tau) = \tau - \ln(1+\tau),\ \tau \geq 0$





Basic properties [6]

Lower surrogate	$f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y}-\mathbf{x}) + \omega\left(\|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}}\right)$	$\mathbf{x}, \mathbf{y} \in \mathrm{dom}(f)$
Upper surrogate	$f(\mathbf{y}) \leq f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y}-\mathbf{x}) + \omega_*\left(\|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}}\right)$	$\|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}} < 1$
Hessian surrogates	$(1-\|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}})^2 \nabla^2 f(\mathbf{x}) \preceq \nabla^2 f(\mathbf{y}) \preceq (1-\|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}})^{-2} \nabla^2 f(\mathbf{x})$	$\|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}} < 1$

Local

Bound on gradient:

$$\frac{\|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}}^2}{1+\|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}}} \leq \langle \nabla f(\mathbf{y}) - \nabla f(\mathbf{x}), \mathbf{y}-\mathbf{x} \rangle \leq \frac{\|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}}^2}{1-\|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}}}, \quad \forall \mathbf{x}, \mathbf{y} \in \mathrm{dom}(f).$$

The right-hand side inequality holds for $\|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}} < 1$.

Variable metric proximal-gradient algorithm for SCM

Variable metric proximal operator

Given $\mathbf{H} \succ 0$ and $g \in \mathcal{F}(\mathbb{R}^p)$. The variable metric proximal operator of g is defined as

$$\boxed{\mathrm{prox}_{\mathbf{H}g}(\mathbf{x}) := \arg\min_{\mathbf{y}\in\mathbb{R}^p} \left\{ g(\mathbf{y}) + (1/2)(\mathbf{y}-\mathbf{x})^T\mathbf{H}^{-1}(\mathbf{y}-\mathbf{x}) \right\}} \tag{14}$$

Variable metric proximal-gradient algorithm for SCM

Variable metric proximal operator

Given $\mathbf{H} \succ 0$ and $g \in \mathcal{F}(\mathbb{R}^p)$. The variable metric proximal operator of g is defined as

$$\mathrm{prox}_{\mathbf{H}g}(\mathbf{x}) := \arg\min_{\mathbf{y}\in\mathbb{R}^p}\left\{g(\mathbf{y}) + (1/2)(\mathbf{y}-\mathbf{x})^T\mathbf{H}^{-1}(\mathbf{y}-\mathbf{x})\right\} \tag{14}$$

Property (Basis properties of variable metric proximal operator)

1. $\mathrm{prox}_{\mathbf{H}g}(\mathbf{x})$ *is well-defined and single-valued (i.e.,* (14) *has unique solution).*
2. *Optimality condition:*

$$\mathbf{x} \in \mathrm{prox}_{\mathbf{H}g}(\mathbf{x}) + \mathbf{H}\partial g(\mathrm{prox}_{\mathbf{H}g}(\mathbf{x})),\ \mathbf{x} \in \mathbb{R}^p.$$

3. $\mathbf{x}^\star$ *is a fixed point of* $\mathrm{prox}_{\mathbf{H}g}(\cdot)$*:*

$$0 \in \partial g(\mathbf{x}^\star) \quad \Leftrightarrow \quad \mathbf{x}^\star = \mathrm{prox}_{\mathbf{H}g}(\mathbf{x}^\star).$$

4. *Non-expansiveness:*

$$\|\mathrm{prox}_{\mathbf{H}g}(\mathbf{x}) - \mathrm{prox}_{\mathbf{H}g}(\tilde{\mathbf{x}})\|_{\mathbf{H}}^* \leq \|\mathbf{x}-\tilde{\mathbf{x}}\|_{\mathbf{H}},\quad \forall \mathbf{x},\tilde{\mathbf{x}} \in \mathbb{R}^p.$$

Variable metric proximal-gradient algorithm

Variable metric proximal-gradient algorithm [11]

1. Choose $\mathbf{x}^0 \in \mathbb{R}^p$ as a starting point and $\mathbf{H}_0 \succ 0$.
2. For $k = 0, 1, \cdots,$ perform:

$$\begin{cases} \mathbf{d}^k & := \mathrm{prox}_{\mathbf{H}_k g}\left(\mathbf{x}^k - \mathbf{H}_k \nabla f(\mathbf{x}^k)\right) - \mathbf{x}^k, \\ \mathbf{x}^{k+1} & := \mathbf{x}^k + \alpha_k \mathbf{d}^k, \end{cases} \tag{15}$$

where $\alpha_k \in (0, 1]$ is a given step size. Update $\mathbf{H}_{k+1} \succ 0$ if necessary.

Variable metric proximal-gradient algorithm

Variable metric proximal-gradient algorithm [11]

1. Choose $\mathbf{x}^0 \in \mathbb{R}^p$ as a starting point and $\mathbf{H}_0 \succ 0$.
2. For $k = 0, 1, \cdots$, perform:

$$\begin{cases} \mathbf{d}^k & := \mathrm{prox}_{\mathbf{H}_k g}\left(\mathbf{x}^k - \mathbf{H}_k \nabla f(\mathbf{x}^k)\right) - \mathbf{x}^k, \\ \mathbf{x}^{k+1} & := \mathbf{x}^k + \alpha_k \mathbf{d}^k, \end{cases} \qquad (15)$$

where $\alpha_k \in (0, 1]$ is a given step size. Update $\mathbf{H}_{k+1} \succ 0$ if necessary.

Common choices of $\mathbf{H}_k$

- $\boxed{\mathbf{H}_k := \lambda_k \mathbb{I}}$, we have $\mathrm{prox}_{\mathbf{H}g} \equiv \mathrm{prox}_{\lambda g}$ and obtain a proximal-gradient method.
- $\boxed{\mathbf{H}_k := \mathbf{D}}$ a diagonal matrix, $\mathrm{prox}_{\mathbf{H}g}$ can be transformed into $\mathrm{prox}_{\lambda g}$ (by scaling the variables) and we obtain a proximal-gradient method.
- $\boxed{\mathbf{H}_k := \nabla^2 f(\mathbf{x}^k)^{-1}}$, we obtain a proximal-Newton method.
- $\boxed{\mathbf{H}_k \approx \nabla^2 f(\mathbf{x}^k)^{-1}}$, we obtain a proximal quasi-Newton method.

Proximal-Newton method for CSM

Proximal-Newton algorithm (PNA)

1. Choose $\mathbf{x}^0 \in \text{dom}(F)$ as a starting point.
2. For $k = 0, 1, \cdots,$ perform:

$$\begin{cases} \mathbf{B}_k & := \nabla^2 f(\mathbf{x}^k), \\ \mathbf{d}^k & := \text{prox}_{\mathbf{B}_k^{-1} g}\left(\mathbf{x}^k - \mathbf{B}_k^{-1}\nabla f(\mathbf{x}^k)\right) - \mathbf{x}^k, \quad \text{(PN direction)} \\ \lambda_k & := \|\mathbf{d}\|_{\mathbf{x}^k}, \quad \text{(PN decrement)} \\ \alpha_k & = (1+\lambda_k)^{-1}, \quad \text{(step-size)} \\ \mathbf{x}^{k+1} & := \mathbf{x}^k + \alpha_k \mathbf{d}^k. \end{cases} \tag{16}$$

Proximal-Newton method for CSM

Proximal-Newton algorithm (PNA)

1. Choose $\mathbf{x}^0 \in \mathrm{dom}(F)$ as a starting point.
2. For $k = 0, 1, \cdots$, perform:

$$\begin{cases} \mathbf{B}_k & := \nabla^2 f(\mathbf{x}^k), \\ \mathbf{d}^k & := \mathrm{prox}_{\mathbf{B}_k^{-1} g}\left(\mathbf{x}^k - \mathbf{B}_k^{-1}\nabla f(\mathbf{x}^k)\right) - \mathbf{x}^k, \quad \text{(PN direction)} \\ \lambda_k & := \|\mathbf{d}\|_{\mathbf{x}^k}, \quad \text{(PN decrement)} \\ \alpha_k & = (1+\lambda_k)^{-1}, \quad \text{(step-size)} \\ \mathbf{x}^{k+1} & := \mathbf{x}^k + \alpha_k \mathbf{d}^k. \end{cases} \tag{16}$$

Complexity per iteration

- Evaluation of $\nabla^2 f(\mathbf{x}^k)$ and $\nabla f(\mathbf{x}^k)$ (closed form expressions).
- Computing $\mathrm{prox}_{\mathbf{H}_k g}$ requires to solve a strongly convex program (14).
- Computing proximal-Newton decrement λ_k requires $(\mathbf{d}^k)^T \nabla f^2(\mathbf{x}^k)\mathbf{d}^k$.

Global convergence

Lemma (Descent lemma [11])

Let $\{\mathbf{x}^k\}_{k\geq 0}$ *be the sequence generated by PNA. Then*

$$\boxed{F(\mathbf{x}^{k+1}) \leq F(\mathbf{x}^k) - \omega(\lambda_k)} \tag{17}$$

where $\omega(\tau) := \tau - \ln(1+\tau) > 0$ *for* $\tau > 0$.

Global convergence

Lemma (Descent lemma [11])

Let $\{\mathbf{x}^k\}_{k\geq 0}$ *be the sequence generated by PNA. Then*

$$\boxed{F(\mathbf{x}^{k+1}) \leq F(\mathbf{x}^k) - \omega(\lambda_k)} \quad (17)$$

where $\omega(\tau) := \tau - \ln(1+\tau) > 0$ *for* $\tau > 0$.

Consequences

- $[F(\mathbf{x}^{k+1}) - F^\star] \leq [F(\mathbf{x}^k) - F^\star] - \omega(\lambda_k)$ for all $k \geq 0$.
- $[F(\mathbf{x}^k) - F(\mathbf{x}^\star] \leq [F(\mathbf{x}^0) - F^\star] - \sum_{j=0}^{k-1} \omega(\lambda_j)$.
- If $\lambda_k \geq \lambda > 0$ for $k = 0, \ldots, K$, then

  $$[F(\mathbf{x}^K) - F^\star] \leq [F(\mathbf{x}^0) - F^\star] - K\omega(\lambda).$$

  The **number of iterations** to reach $F(\mathbf{x}^K) - F^\star \leq \varepsilon$ is

  $$K := \left\lfloor \frac{[F(\mathbf{x}^0) - F^\star] - \varepsilon}{\omega(\lambda)} \right\rfloor + 1.$$

- Global convergence rate is just sublinear, i.e., $\mathcal{O}(1/k)$.

Proof of (17)

Sketch of proof.

- Let $\mathbf{s}^k := \mathbf{x}^k + \mathbf{d}^k$. We have $\mathbf{x}^{k+1} - \mathbf{x}^k = \alpha_k \mathbf{d}^k$ and $\mathbf{x}^{k+1} = (1-\alpha_k)\mathbf{x}^k + \alpha_k \mathbf{s}^k$.
- By convexity of g:

$$g(\mathbf{x}^{k+1}) \leq (1-\alpha_k)g(\mathbf{x}^k) + \alpha_k g(\mathbf{s}^k), \ \ \alpha_k \in (0,1]. \tag{18}$$

- By subgradient definition:

$$g(\mathbf{s}^k) \leq g(\mathbf{x}^k) + \mathbf{v}(\mathbf{s}^k)^T(\mathbf{s}^k - \mathbf{x}^k), \ \ \forall\ \mathbf{v}(\mathbf{s}^k) \in \partial g(\mathbf{s}^k). \tag{19}$$

- Substituting (19) into (18) we get

$$g(\mathbf{x}^{k+1}) \leq g(\mathbf{x}^k) + \alpha_k \mathbf{v}(\mathbf{s}^k)^T \mathbf{d}^k. \tag{20}$$

- By self-concordance of f (upper bound inequality):

$$f(\mathbf{x}^{k+1}) \leq f(\mathbf{x}^k) + \nabla f(\mathbf{x}^k)(\mathbf{x}^{k+1} - \mathbf{x}^k) + \omega_*(\|\mathbf{x}^{k+1} - \mathbf{x}^k\|_{\mathbf{x}^k}), \tag{21}$$

  under condition $\|\mathbf{x}^{k+1} - \mathbf{x}^k\|_{\mathbf{x}^k} < 1$.

□

Proof of (17) (cont)

Sketch of proof (cont).

- Summing up (20) and (21) and using $F := f + g$, we get

$$F(\mathbf{x}^{k+1}) \leq F(\mathbf{x}^k) + \alpha_k[\nabla f(\mathbf{x}^k) + \mathbf{v}(\mathbf{s}^k)]^T \mathbf{d}^k + \omega_*(\alpha_k \|\mathbf{d}^k\|_{\mathbf{x}^k}). \tag{22}$$

- From the optimality property 2 of (14) we have

$$\nabla f(\mathbf{x}^k) + \mathbf{v}(\mathbf{s}^k) = -\nabla^2 f(\mathbf{x}^k)\mathbf{d}^k. \tag{23}$$

- Plug (24) into (22) and use $\lambda_k := \|\mathbf{d}^k\|_{\mathbf{x}^k}$, we get

$$F(\mathbf{x}^{k+1}) \leq F(\mathbf{x}^k) - \alpha_k \lambda_k^2 + \omega_*(\alpha_k \lambda_k). \tag{24}$$

- Let $\psi(\alpha) := \alpha\lambda_k^2 - \omega_*(\alpha\lambda_k) = \alpha\lambda_k^2 + \alpha\lambda_k + \ln(1 - \alpha\lambda_k)$. This function attains the maximum at $\alpha_k = (1 + \lambda_k)^{-1}$ and $\psi(\alpha_k) = \lambda_k - \ln(1 + \lambda_k)$. Hence, we have

$$F(\mathbf{x}^{k+1}) \leq F(\mathbf{x}^k) - \omega(\lambda_k),$$

which is (17).

□

Local convergence

Theorem (Local quadratic convergence [11])

Let $\{\mathbf{x}^k\}$ be the sequence generated by ***PNA****. If $\|\mathbf{x}^0 - \mathbf{x}^\star\|_{\mathbf{x}^\star} \leq \sigma_0 := 0.08763$, then*

$$\boxed{\|\mathbf{x}^{k+1} - \mathbf{x}^\star\|_{\mathbf{x}^\star} \leq c^* \|\mathbf{x}^k - \mathbf{x}^\star\|^2_{\mathbf{x}^\star}}, \quad k \geq 0,$$

where $c^* := 3.57$.
Consequently, $\{\mathbf{x}^k\}_{k\geq 0}$ converges to $\mathbf{x}^\star$ at a quadratic rate.

Local convergence

Theorem (Local quadratic convergence [11])

Let $\{\mathbf{x}^k\}$ *be the sequence generated by* ***PNA***. *If* $\|\mathbf{x}^0 - \mathbf{x}^\star\|_{\mathbf{x}^\star} \leq \sigma_0 := 0.08763$, *then*

$$\boxed{\|\mathbf{x}^{k+1} - \mathbf{x}^\star\|_{\mathbf{x}^\star} \leq c^* \|\mathbf{x}^k - \mathbf{x}^\star\|^2_{\mathbf{x}^\star}}, \quad k \geq 0,$$

where $c^* := 3.57$.
Consequently, $\{\mathbf{x}^k\}_{k\geq 0}$ *converges to* $\mathbf{x}^\star$ *at a quadratic rate.*

Quadratic convergence region

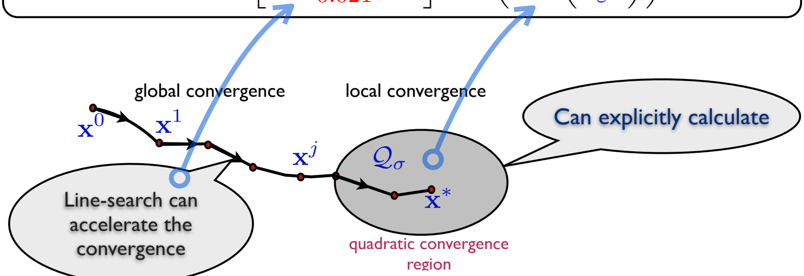
Let $\sigma := 0.08763$. Then the **quadratic convergence region** $\mathcal{Q}_\sigma$ is defined as:

$$\boxed{\mathcal{Q}_\sigma := \{\mathbf{x} \in \mathrm{dom}(F) \ : \ \|\mathbf{x} - \mathbf{x}^\star\|_{\mathbf{x}^\star} \leq \sigma\}.}$$

For any $\mathbf{x}^0 \in \mathcal{Q}_\sigma$, $\{\mathbf{x}^k\}$ converges to $\mathbf{x}^\star$ at a quadratic rate.

Overall analytical worst-case complexity

$$\#\text{iterations} = \left\lfloor \frac{F(\mathbf{x}^0) - F^\star}{0.021} \right\rfloor + O\left(\ln\ln\left(\frac{4.56}{\varepsilon}\right)\right)$$



$$\mathcal{Q}_\sigma := \{\mathbf{x} \in \mathrm{dom}(F) \;:\; \|\mathbf{x} - \mathbf{x}^\star\|_{\mathbf{x}^\star} \leq \sigma\}$$

Enhancements

Two new line-search strategies

The optimal step-size $\alpha_k^* := (1+\lambda_k)^{-1}$ provides a **lower bound**. Perform line-search on $[\alpha_k^*, 1]$.

- **Forward line-search**: Start from α_k and increase the step-size until meet 1.
- **Enhanced backtracking**: Start from 1 and decrease the step size until meet α_k^*

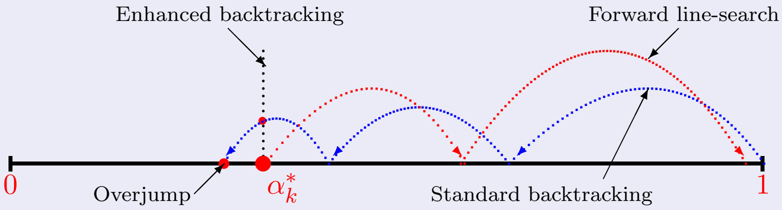
Enhancements

Two new line-search strategies

The optimal step-size $\alpha_k^* := (1+\lambda_k)^{-1}$ provides a **lower bound**. Perform line-search on $[\alpha_k^*, 1]$.

- **Forward line-search**: Start from α_k and increase the step-size until meet 1.
- **Enhanced backtracking**: Start from 1 and decrease the step size until meet α_k^*

Illustration of three line-search strategies



Example: Graphical model selection

Graphical model selection

$$\min_{\Theta \succ 0} \Big\{ \underbrace{\operatorname{tr}(\Sigma\Theta) - \log\det(\Theta)}_{f(\Theta)} + \underbrace{\rho \|\operatorname{vec}(\Theta)\|_1}_{g(\Theta)} \Big\}.$$

Example: Graphical model selection

Graphical model selection

$$\min_{\Theta \succ 0} \Big\{ \underbrace{\mathrm{tr}(\Sigma\Theta) - \log\det(\Theta)}_{f(\Theta)} + \underbrace{\rho\|\mathrm{vec}(\Theta)\|_1}_{g(\Theta)} \Big\}.$$

Computational cost

- $\nabla f(\Theta) = \mathrm{vec}(\Sigma - \Theta_k^{-1})$ and $\nabla^2 f(\Theta^k) = \Theta_k^{-1} \otimes \Theta_k^{-1}$ ($\otimes$-Kronecker product).
- Compute the **search direction** $\mathbf{d}_k$ via dualization:

$$\mathbf{U}_k = \arg\min_{\|\mathrm{vec}(\mathbf{U})\|_\infty \leq 1} \Big\{ (1/2)\mathrm{trace}((\Theta_k \mathbf{U})^2) + \mathrm{trace}(\mathbf{Q}_k \mathbf{U}) \Big\},$$

  where $\mathbf{Q}_k := \rho^{-1}(\Theta_k \Sigma \Theta_k - 2\Theta_k)$. Then $\mathbf{d}^k := -((\Theta_k\Sigma - \mathbb{I})\Theta_k + \rho\Theta_k \mathbf{U}_k \Theta_k)$.
- The proximal-Newton decrement λ_k:

$$\lambda_k := (p - 2\mathrm{trace}(\mathbf{W}_k) + \mathrm{trace}(\mathbf{W}_k^2))^{1/2}, \quad \mathbf{W}_k := \Theta_k(\Sigma + \rho\mathbf{U}_k).$$

Example: Graphical model selection

Graphical model selection

$$\min_{\Theta \succ 0} \Big\{ \underbrace{\mathrm{tr}(\Sigma\Theta) - \log\det(\Theta)}_{f(\Theta)} + \underbrace{\rho\|\mathrm{vec}(\Theta)\|_1}_{g(\Theta)} \Big\}.$$

Computational cost

- $\nabla f(\Theta) = \mathrm{vec}(\Sigma - \Theta_k^{-1})$ and $\nabla^2 f(\Theta^k) = \Theta_k^{-1} \otimes \Theta_k^{-1}$ ($\otimes$-Kronecker product).
- Compute the **search direction** $\mathbf{d}_k$ via dualization:

$$\mathbf{U}_k = \arg\min_{\|\mathrm{vec}(\mathbf{U})\|_\infty \leq 1} \Big\{ (1/2)\mathrm{trace}((\Theta_k \mathbf{U})^2) + \mathrm{trace}(\mathbf{Q}_k \mathbf{U}) \Big\},$$

  where $\mathbf{Q}_k := \rho^{-1}(\Theta_k \Sigma \Theta_k - 2\Theta_k)$. Then $\mathbf{d}^k := -((\Theta_k \Sigma - \mathbb{I})\Theta_k + \rho \Theta_k \mathbf{U}_k \Theta_k)$.
- The proximal-Newton decrement λ_k:

$$\lambda_k := (p - 2\mathrm{trace}(\mathbf{W}_k) + \mathrm{trace}(\mathbf{W}_k^2))^{1/2}, \quad \mathbf{W}_k := \Theta_k(\Sigma + \rho \mathbf{U}_k).$$

Only need **matrix-matrix multiplications**
No Cholesky factorizations or matrix inversions

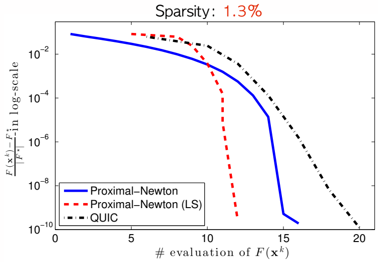
cf. Lecture 5 @ `http://lions.epfl.ch/mathematics_of_data`

Test on the real-data: Lymph and Leukemia

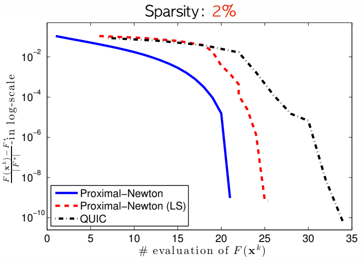
- **PNA vs. QUIC:**
  - QUIC subproblem solver: special block-coordinate descent algorithm.
  - PNA subproblem solver: general proximal-gradient algorithms.

On the average $\times 5$ acceleration (up to $\times 15$) over Matlab QUIC

- **Convergence behavior:** $\rho = 0.5$ - Gene data (Genetic regulatory network)



Lymph [p = 587] ~ 350,000 variables



Leukemia [p = 1255] ~ 1.5 millions variables

[0] Details: Composite self-concordant minimization, Journal of Machine Learning Research, vol. 16, 2015

Proximal-gradient method for CSM

Choice of variable matrix and line-search condition

$$\boxed{\mathbf{H}_k := L_k \mathbb{I}, \quad L_k > 0}$$

Line search condition: Find the largest L_k such that:

$$\boxed{L_k \leq \eta_k := \frac{\lambda_k^2}{\|\mathbf{d}^k\|_2^2}.} \tag{25}$$

Proximal-gradient method for CSM

Choice of variable matrix and line-search condition

$$\boxed{\mathbf{H}_k := L_k \mathbb{I}, \quad L_k > 0}$$

Line search condition: Find the largest L_k such that:

$$\boxed{L_k \leq \eta_k := \frac{\lambda_k^2}{\|\mathbf{d}^k\|_2^2}.} \tag{25}$$

Proximal-gradient algorithm (PGA)

1. Given $\varepsilon > 0$. Choose $\mathbf{x}^0 \in \text{dom}(F)$ as a starting point.
2. For $k = 0, 1, \cdots$, perform:
   - 2.1. Choose $L_k > 0$ satisfies (25).
   - 2.2. $\mathbf{d}^k := \text{prox}_{\lambda_k g}(\mathbf{x}^k - \gamma_k \nabla f(\mathbf{x}^k)) - \mathbf{x}^k$, with $\gamma_k := 1/L_k$.
   - 2.3. $\lambda_k := \|\mathbf{d}^k\|_{\mathbf{x}^k}$, $\beta_k := \sqrt{L_k}\|\mathbf{d}^k\|_2$.
   - 2.4. If $\beta_k \leq \varepsilon$, terminate.
   - 2.5. *Step size*: $\alpha_k := \beta_k^2/(\lambda_k(\lambda_k + \beta_k^2))$.
   - 2.6. Update $\mathbf{x}^{k+1} := \mathbf{x}^k + \alpha_k \mathbf{d}^k$.

Global convergence and local convergence

Theorem (Global convergence [11])

- *If* $L_k \geq \underline{L} > 0$ *for all* $k \geq 0$ *and* $\mathcal{L}_F(F(\mathbf{x}^0)) := \{\mathbf{x} \in \mathit{dom}(F) : F(\mathbf{x}) \leq F(\mathbf{x}^0)\}$ *is bounded from below, then* $\{\mathbf{x}^k\}$ *generated by PGA converges to* $\mathbf{x}^\star$.
- *Let*

$$\bar{\mathbf{x}}^k := S_k^{-1} \sum_{j=0}^{k} \alpha_k \mathbf{x}^j, \quad \text{where } S_k := \sum_{j=0}^{k} \alpha_j > 0.$$

Then $\boxed{F(\bar{\mathbf{x}}^k) - F^\star \leq \frac{\bar{L}_k}{2S_k} \|\mathbf{x}^0 - \mathbf{x}^\star\|_2^2}$, *where* $\bar{L}_k := \max_{0 \leq j \leq k} L_j$.

Global convergence and local convergence

Theorem (Global convergence [11])

- *If* $L_k \geq \underline{L} > 0$ *for all* $k \geq 0$ *and* $\mathcal{L}_F(F(\mathbf{x}^0)) := \{\mathbf{x} \in \mathrm{dom}(F) : F(\mathbf{x}) \leq F(\mathbf{x}^0)\}$ *is bounded from below, then* $\{\mathbf{x}^k\}$ *generated by PGA converges to* $\mathbf{x}^\star$*.*
- *Let*

$$\bar{\mathbf{x}}^k := S_k^{-1} \sum_{j=0}^{k} \alpha_k \mathbf{x}^j, \quad \text{where } S_k := \sum_{j=0}^{k} \alpha_j > 0.$$

Then $\boxed{F(\bar{\mathbf{x}}^k) - F^\star \leq \frac{\bar{L}_k}{2S_k} \|\mathbf{x}^0 - \mathbf{x}^\star\|_2^2}$, *where* $\bar{L}_k := \max_{0 \leq j \leq k} L_j$.

Theorem (Local convergence [11])

Assumptions:

- *Let* $\mathbf{x}^\star$ *be the unique solution of* (1) *such that* $\nabla^2 f(\mathbf{x}^\star) \succ 0$*.*
- *For* k *sufficiently large, if* $\mathbf{D}_k := L_k \mathbb{I}$ *and* $\max\{|1 - \frac{L_k}{\sigma^*_{\min}}|, |1 - \frac{L_k}{\sigma^*_{\max}}|\} < \frac{1}{2}$.

Conclusion: $\{\mathbf{x}^k\}_{k \geq 0}$ *generated by PGA converges to* $\mathbf{x}^\star$ *at a linear rate.*

Example 1: Graphical model selection

Graphical model selection

$$\min_{\Theta \succ 0} \left\{ \underbrace{\mathrm{tr}(\Sigma\Theta) - \log\det(\Theta)}_{f(\Theta)} + \underbrace{\rho\|\mathrm{vec}(\Theta)\|_1}_{g(\Theta)} \right\}.$$

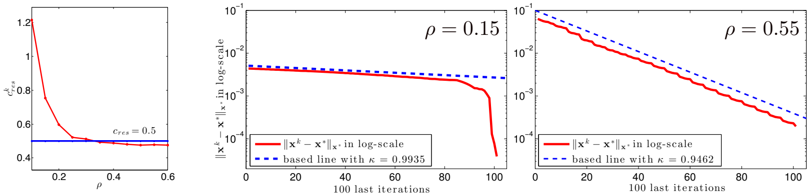
Example 1: Graphical model selection

Graphical model selection

$$\min_{\Theta \succ 0} \Big\{ \underbrace{\operatorname{tr}(\Sigma\Theta) - \log\det(\Theta)}_{f(\Theta)} + \underbrace{\rho\|\operatorname{vec}(\Theta)\|_1}_{g(\Theta)} \Big\}.$$

Linear convergence of PGA

Graph learning: Lymph [p = 587]



Improvement - greedy proximal gradient variant

Mathematical observation

Let us define

- $\mathbf{s}_g^k := \mathbf{x}^k + \mathbf{d}^k$
- $\hat{\mathbf{x}}^k = (1-\alpha_k)\mathbf{x}^k + \alpha_k \mathbf{s}^k$ for $\alpha_k \in (0,1]$.

If $F(\mathbf{s}_g^k) \leq F(\mathbf{x}^k)$, then by convexity of F:

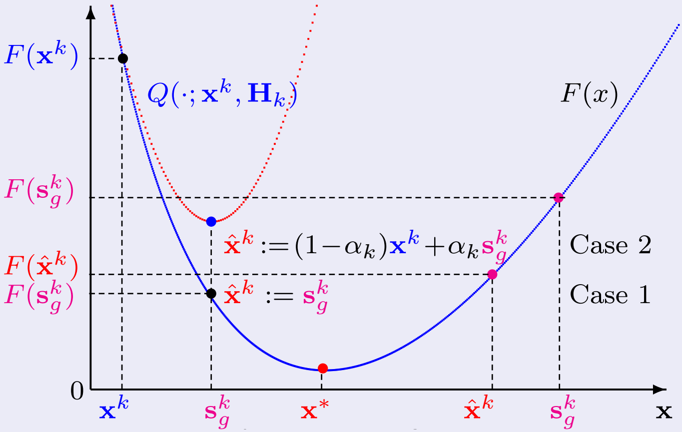
$$F(\hat{\mathbf{x}}^k) = F((1-\alpha_k)\mathbf{x}^k + \alpha_k) \leq (1-\alpha_k)F(\mathbf{x}^k) + \alpha_k F(\mathbf{s}_g^k) \overset{F(\mathbf{s}_g^k)\leq F(\mathbf{x}^k)}{\leq} F(\mathbf{x}^k)$$

By comparing $F(\mathbf{x}^k)$, $F(\mathbf{s}_g^k)$ and $F(\hat{\mathbf{x}}^k)$ we can pick $\mathbf{x}^{k+1}$ as

$$\mathbf{x}^{k+1} = \begin{cases} \mathbf{s}_g^k & \text{if } \mathbf{s}_g^k \in \mathrm{dom}(F) \text{ and } F(\mathbf{s}_g^k) \leq F(\mathbf{x}^k), \\ \hat{\mathbf{x}}^k & \text{otherwise.} \end{cases}$$

Improvement - greedy proximal gradient variant

Visualization of the idea



Example 2: Poisson imaging reconstruction

Optimization problem with TV-norm

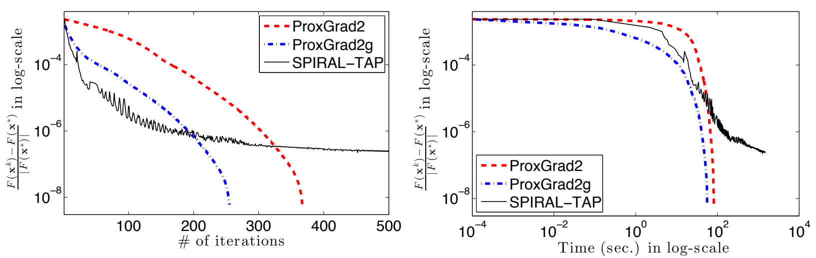
$$\min_{\mathbf{x}\in\mathbb{R}^{n\times p}} \Big\{ \underbrace{\sum_{i=1}^{n}(\mathbf{K}\mathbf{x})_i - \sum_{i=1}^{n} y_i \log((\mathbf{K}\mathbf{x})_i)}_{f(\mathbf{x})} + \underbrace{\rho\|\mathbf{x}\|_{\mathrm{TV}}}_{g(\mathbf{x})} \Big\}$$

Example 2: Poisson imaging reconstruction

Optimization problem with TV-norm

$$\min_{\mathbf{x}\in\mathbb{R}^{n\times p}} \Big\{ \underbrace{\sum_{i=1}^{n} (\mathbf{K}\mathbf{x})_i - \sum_{i=1}^{n} y_i \log((\mathbf{K}\mathbf{x})_i)}_{f(\mathbf{x})} + \underbrace{\rho\|\mathbf{x}\|_{\mathrm{TV}}}_{g(\mathbf{x})} \Big\}$$

Convergence of PGA, greedy PGA and SPIRAL-TAP



Example 2: Poisson imaging reconstruction - cont.

Visualization of the outcome - cameraman

Overview of algorithms/complexity

Assumption	Algorithm	Convergence rate (ε)	Complexity per iteration
$f, g \in \mathcal{F}(\mathbb{R}^p)$	Subgradient	$\mathcal{O}(1/\sqrt{k})$	1 sub-gradient of f, g
	Bundle method	$\mathcal{O}(1/\sqrt{k})$	1 sub-gradient of f, g
	Mirror-descent	$\mathcal{O}(1/\sqrt{k})$	1 sub-gradient of f, g
$f \in \mathcal{F}^{1,1}_{L,\mu}(\mathbb{R}^p)$, $g \in \mathcal{F}_{\text{prox}}(\mathbb{R}^n)$	Proximal-gradient	$\mathcal{O}(1/k)$ $(\mu = 0)$, linear $(\mu > 0)$	1 gradient, 1 prox operator
	Accelerated proximal-gradient	$\mathcal{O}(1/k^2)$ $(\mu = 0)$, linear $(\mu > 0)$	1 gradient, 1 or 2 prox operator(s)
	Proximal quasi-Newton	locally superlinear, globally sublinear	One gradient, rank-2 update
	Proximal Newton	locally quadratic, locally sublinear $\mathcal{O}(1/k^s)$, $1 \leq s \leq 3$	One gradient, one Hessian inverse
$f, g \in \mathcal{F}_{\text{prox}}(\mathbb{R}^n)$	Peaceman-Douglas	$\mathcal{O}(1/k)$-ergodic	≥ 1 prox operator(s) f, g
	Douglas-Rachford	$\mathcal{O}(1/k)$-ergodic	≥ 1 prox operator(s) f, g
	ALM	$\mathcal{O}(1/k^2)$	≥ 1 prox operator(s) f, g
	ADMM	$\mathcal{O}(1/k)$	≥ 1 prox operator(s) f, g

- ALM = augmented Lagrangian method, ADMM = alternating direction method of multiplier.
- $\mathcal{F}$ = class of proper, closed convex functions.
- $\mathcal{F}^{1,1}_{L,\mu}$ = class of strongly convex functions with Lipschitz gradient.
- $\mathcal{F}_{\text{prox}}$ = class of convex functions with tractable prox-operator.

Overview of algorithms/complexity

Assumption	Algorithm	Convergence rate	Complexity per iteration
$f, g \in \mathcal{F}(\mathbb{R}^p)$	Subgradient	$\mathcal{O}(1/\sqrt{k})$	1 sub-gradient of f, g
	Bundle method	$\mathcal{O}(1/\sqrt{k})$	1 sub-gradient of f, g
	Mirror-descent	$\mathcal{O}(1/\sqrt{k})$	1 sub-gradient of f, g
$f \in \mathcal{F}^{1,1}_{L,\mu}(\mathbb{R}^p)$, $g \in \mathcal{F}_{\text{prox}}(\mathbb{R}^n)$	Proximal-gradient	$\mathcal{O}(1/k)$ ($\mu = 0$), linear ($\mu > 0$)	1 gradient, 1 prox operator
	Accelerated proximal-gradient	$\mathcal{O}(1/k^2)$ ($\mu = 0$), linear ($\mu > 0$)	1 gradient, 1 or 2 prox operator(s)
	Proximal quasi-Newton	locally superlinear, globally sublinear	One gradient, rank-2 update
	Proximal Newton	locally quadratic, locally sublinear $\mathcal{O}(1/k^s)$, $1 \leq s \leq 3$	One gradient, one Hessian inverse
$f, g \in \mathcal{F}_{\text{prox}}(\mathbb{R}^n)$	Peaceman-Douglas	$\mathcal{O}(1/k)$-ergodic	≥ 1 prox operator(s) f, g
	Douglas-Rachford	$\mathcal{O}(1/k)$-ergodic	≥ 1 prox operator(s) f, g
	ALM	$\mathcal{O}(1/k^2)$	≥ 1 prox operator(s) f, g
	ADMM	$\mathcal{O}(1/k)$	≥ 1 prox operator(s) f, g

- ALM = augmented Lagrangian method, ADMM = alternating direction method of multiplier.
- $\mathcal{F}$ = class of proper, closed convex functions.
- $\mathcal{F}^{1,1}_{L,\mu}$ = class of strongly convex functions with Lipschitz gradient.
- $\mathcal{F}_{\text{prox}}$ = class of convex functions with tractable prox-operator.

References I

[1] A. Beck and M. Teboulle.
A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems.
SIAM J. Imaging Sciences, 2(1):183–202, 2009.

[2] S. Becker and M.J. Fadili.
A quasi-Newton proximal splitting method.
In *Proceedings of Neutral Information Processing Systems Foundation*, 2012.

[3] P. Combettes and Pesquet J.-C.
Signal recovery by proximal forward-backward splitting.
In *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, pages 185–212. Springer-Verlag, 2011.

[4] O. Güler.
On the convergence of the proximal point algorithm for convex minimization.
SIAM J. Control Optim., 29(2):403–419, 1991.

[5] J.D. Lee, Y. Sun, and M.A. Saunders.
Proximal newton-type methods for convex optimization.
Tech. Report., pages 1–25, 2012.

References II

[6] Y. Nesterov.
Introductory lectures on convex optimization: a basic course, volume 87 of *Applied Optimization*.
Kluwer Academic Publishers, 2004.

[7] Y. Nesterov and A. Nemirovski.
Interior-point Polynomial Algorithms in Convex Programming.
Society for Industrial Mathematics, 1994.

[8] N. Parikh and S. Boyd.
Proximal algorithms.
Foundations and Trends in Optimization, 1(3):123–231, 2013.

[9] R. T. Rockafellar.
Convex Analysis, volume 28 of *Princeton Mathematics Series*.
Princeton University Press, 1970.

[10] R.T. Rockafellar.
Monotone operators and the proximal point algorithm.
SIAM Journal on Control and Optimization, 14:877–898, 1976.

[11] Q. Tran-Dinh, A. Kyrillidis, and V. Cevher.
Composite self-concordant minimization.
Tech. report, Lab. for Information and Inference Systems (LIONS), EPFL, Switzerland, CH-1015 Lausanne, Switzerland, January 2013.