Lecture 9: Composite minimization II

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

EE-556 (Fall 2019)
License Information for Mathematics of Data Slides

- This work is released under a Creative Commons License with the following terms:
  - **Attribution**
    - The licensor permits others to copy, distribute, display, and perform the work. In return, licensees must give the original authors credit.
  - **Non-Commercial**
    - The licensor permits others to copy, distribute, display, and perform the work. In return, licensees may not use the work for commercial purposes – unless they get the licensor's permission.
  - **Share Alike**
    - The licensor permits others to distribute derivative works only under a license identical to the one that governs the licensor’s work.
  - [Full Text of the License](#)

---

Mathematics of Data: From Theory to Computation | Prof. Volkan Cevher, volkan.cevher@epfl.ch

[EPFL]
Outline

▶ Today
  1. Proximal gradient method - Nonconvex case
  2. Proximal-Newton method

▶ Next week
Recommended reading material

Motivation

Data analytics problems in various disciplines can often be simplified to nonsmooth composite 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 nonconvex minimization**

We study the following composite nonconvex minimization problem,

\[
F^* := \min_{\mathbf{x} \in \mathbb{R}^p} \{ F(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x}) \}, \tag{CM}
\]

where

- \( g: \mathbb{R}^p \to \mathbb{R} \cup \{\infty\} \) is proper, closed, convex, and (possibly) nonsmooth.
- \( f: \mathbb{R}^p \to \mathbb{R} \) is proper, closed, nonconvex, \( \text{dom}(f) \) is convex, and \( f \in \mathcal{F}_{L_f}^{1,1} \).

- Example: Neural network pruning with \( \ell_1 \)-norm regularization
Recall: Proximal-gradient algorithm

<table>
<thead>
<tr>
<th>Basic proximal-gradient scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Choose $x^0 \in \text{dom}(F)$ arbitrarily as a starting point.</td>
</tr>
<tr>
<td>2. For $k = 0, 1, \cdots$, generate a sequence ${x^k}_{k \geq 0}$ as:</td>
</tr>
<tr>
<td>$x^{k+1} := \text{prox}_{\alpha g} \left( x^k - \alpha \nabla f(x^k) \right)$,</td>
</tr>
</tbody>
</table>
Recall: Proximal-gradient algorithm

<table>
<thead>
<tr>
<th>Basic proximal-gradient scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1.</strong> Choose $x^0 \in \text{dom}(F)$ arbitrarily as a starting point.</td>
</tr>
<tr>
<td><strong>2.</strong> For $k = 0, 1, \cdots$, generate a sequence ${x^k}_{k \geq 0}$ as:</td>
</tr>
</tbody>
</table>

$$x^{k+1} := \text{prox}_{\alpha g} \left( x^k - \alpha \nabla f(x^k) \right),$$

Next:

- We analyze the same algorithm as in lecture 8, but with nonconvex objective.
- Guarantees: To the first order stationary point of the composite problem.
- Metric for the convergence: Gradient Mapping.
Sufficient decrease property

\[ x^{k+1} := \text{prox}_{\alpha g} \left( x^k - \alpha \nabla f(x^k) \right) \]

Assumption

\( f \) is smooth with parameter \( L \), i.e., \( f \in \mathcal{F}_{L}^{1,1}(\mathbb{R}^p) \).

\( g : \mathbb{R}^p \rightarrow \mathbb{R} \cup \{+\infty\} \) is proper, closed, convex, and (possibly) nonsmooth. \( g \) is proximally tractable.

Lemma (Sufficient decrease)

\[
F(x^{k+1}) \leq F(x^k) - \alpha \left( 1 - \frac{\alpha L}{2} \right) \left\| \frac{1}{\alpha} (x^k - x^{k+1}) \right\|^2_2.
\] (1)

\( \triangleright \) Note: \( \frac{1}{\alpha} (x^k - x^{k+1}) = \frac{1}{\alpha} \left( x^k - \text{prox}_{\alpha g} \left( x^k - \alpha \nabla f(x^k) \right) \right) \)

Corollary

\[
F(x^{k+1}) \leq F(x^k) - \frac{1}{2L} \left\| Lf(x^k - x^{k+1}) \right\|^2_2, \quad \text{for } \alpha = 1/L
\]
A metric for 1st order stationarity: Gradient mapping

Definition (Gradient Mapping)

\[ G_\alpha(x^k) := \frac{1}{\alpha} (x^k - x^{k+1}) = \frac{1}{\alpha} \left( x^k - \text{prox}_{\alpha g} \left( x^k - \alpha \nabla f(x^k) \right) \right) \]
A metric for 1st order stationarity: Gradient mapping

**Definition (Gradient Mapping)**

\[
G_{\alpha}(x^k) := \frac{1}{\alpha}(x^k - x^{k+1}) = \frac{1}{\alpha}\left(x^k - \text{prox}_{\alpha g}\left(x^k - \alpha \nabla f(x^k)\right)\right)
\]

Remarks:
- \(G_{\alpha}(x) = 0 \iff x \text{ is a stationary point.}\)
- If \(g = 0\) \(\Rightarrow G_{\alpha}(x) = \nabla f(x) \Rightarrow \text{Reduces to same metric for gradient descent.}\)
A metric for 1st order stationarity: **Gradient mapping**

**Definition (Gradient Mapping)**

\[ G_\alpha(x^k) := \frac{1}{\alpha}(x^k - x^{k+1}) = \frac{1}{\alpha}\left(x^k - \text{prox}_{\alpha g}(x^k - \alpha \nabla f(x^k))\right) \]

**Remarks:**

- \( G_\alpha(x) = 0 \iff x \) is a stationary point.
- If \( g = 0 \) \( \Rightarrow G_\alpha(x) = \nabla f(x) \Rightarrow \) Reduces to same metric for gradient descent.

**Lipschitz continuity**

\[
\|G_\alpha(x) - G_\alpha(y)\| \leq \left(\frac{2}{\alpha} + L\right)\|x - y\|, \quad \forall x, y \in \text{dom}(F)
\]

\[
\left\|G_{\frac{1}{L}}(x) - G_{\frac{1}{L}}(y)\right\| \leq 3L\|x - y\|, \quad \forall x, y \in \text{dom}(F)
\]

**Monotonicity**

\( \alpha_1 \geq \alpha_2 > 0, \)

\[ \|G_{\alpha_1}(x)\| \geq \|G_{\alpha_2}(x)\|, \quad \forall x \in \text{dom}(F) \]
Choosing the step-size

**Step size selection**

**Constant.** \( \alpha_k = \alpha \in \left(0, \frac{2}{L}\right) \)

**Backtracking line search for estimating** \( L_f \): Find the smallest integer \( i_k \) which satisfies the sufficient decrease property for the set of parameters \((\gamma, \eta)\), where \( \gamma \in (0, 1) \), and \( \eta > 1 \), i.e.,

\[
F(x^{k+1}) \leq F(x^k) - \frac{\gamma}{L_k} \|G_{\alpha_k}(x^k)\|_2^2,
\]

where \( \alpha_k = \frac{1}{L_k} = \frac{1}{L_{k-1}\eta^{i_k}} \).

**Remarks**

**Line search** procedure terminates in finite step under the given assumptions. See (1).

If \( \hat{L} \geq \frac{L}{2(1 - \gamma)} \) \( \Rightarrow \) \( \frac{\hat{L} - \frac{L}{2}}{\hat{L}} \geq \gamma \) \( \Rightarrow \)

\[
F(x^{k+1}) \leq F(x^k) - \frac{\gamma}{L} \|G_{\alpha_k}(x^k)\|_2^2.
\]

This inequality implies that linesearch must stop when \( L_k \geq \frac{L}{2(1 - \gamma)} \).

\( L_k \) is upper bounded: \( L_k \leq \max \left\{ L_{k-1}, \frac{\eta L}{2(1 - \gamma)} \right\} \) \( \Rightarrow \alpha_k \) is upper bounded.
Nonconvex case: Convergence

Sufficient decrease property

\[ \{x^k\}_{k \geq 0}: \text{sequence generated by the proximal gradient method.} \]

- **Step size:**
  - Constant step size: \( \alpha \).
  - Step size with backtracking linesearch with parameters \((\gamma, \eta)\), \( \alpha_k = 1/L_k \).

- **Sufficient decrease property:**
  - **Constant step size:**
    \[
    F(x^{k+1}) \leq F(x^k) - \alpha \left(1 - \frac{\alpha L}{2}\right) \|G_{\alpha}(x^k)\|_2^2.
    \]
  
- **Line search:**
    \[
    F(x^{k+1}) \leq F(x^k) - \gamma \min \left\{ \alpha_{k-1}, \frac{2(1 - \gamma)}{\eta L} \right\} \|G_{\alpha_{k-1}}(x^k)\|_2^2.
    \]
Nonconvex case: Convergence

**Theorem (Convergence of proximal-gradient method: Nonconvex [2])**

Let \( \{x^k\} \) be generated by proximal-gradient method. Then:

\[
\min_{0,1,\ldots,k} \|G_\alpha(x^k)\|_2 \leq \frac{\sqrt{F(x^0) - F(x^*)}}{\sqrt{M(k + 1)}}
\]

The worst-case complexity to reach \( \min_{0,1,\ldots,k} \|G_\alpha(x^k)\|_2 \leq \varepsilon, \varepsilon\text{-first order stationary of the composite problem} \) is \( \mathcal{O}\left(\frac{1}{\varepsilon^2}\right) \).

**Remark.**

Same convergence rate is preserved as in the unconstrained nonconvex setting.
Recall: Composite convex minimization

Problem (Unconstrained composite convex minimization)

\[
F^* := \min_{x \in \mathbb{R}^p} \{ F(x) := f(x) + g(x) \}
\]

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

Proximal gradient method (ISTA) vs. fast proximal gradient method (FISTA)

Assumptions, step sizes and convergence rates

Proximal gradient method:

\[ f \in \mathcal{F}^{1,1}_L, \quad \alpha = \frac{1}{L} \]

\[ F(x^k) - F(x^*) \leq \epsilon, \quad O\left(\frac{1}{\epsilon}\right). \]

Fast proximal gradient method:

\[ f \in \mathcal{F}^{1,1}_L, \quad \alpha = \frac{1}{L} \]

\[ F(x^k) - F(x^*) \leq \epsilon, \quad O\left(\frac{1}{\sqrt{\epsilon}}\right). \]
Recall: Composite convex minimization guarantees

Proximal gradient method (ISTA) vs. fast proximal gradient method (FISTA)

Assumptions, step sizes and convergence rates

Proximal gradient method:

\[
\begin{align*}
&f \in \mathcal{F}^{1,1}_L, \quad \alpha = \frac{1}{L} \\
&F(x^k) - F(x^*) \leq \epsilon, \quad O\left(\frac{1}{\epsilon}\right).
\end{align*}
\]

Fast proximal gradient method:

\[
\begin{align*}
&f \in \mathcal{F}^{1,1}_L, \quad \alpha = \frac{1}{L} \\
&F(x^k) - F(x^*) \leq \epsilon, \quad O\left(\frac{1}{\sqrt{\epsilon}}\right).
\end{align*}
\]

- We require \(\alpha_k\) to be a function of \(L\).
- It may not be possible to know exactly the Lipschitz constant. Line-search?
- Adaptation to local geometry \(\rightarrow\) may lead to larger steps.
How can we better adapt to the local geometry?

Non-adaptive:

\[ f(x^k) \]

Global quadratic upper bound

\[ Q_L(x, x^k) \]

\[ \bullet x^{k+1} = \arg \min_x \left\{ f(x^k) + \langle \nabla f(x^k), x - x^k \rangle + \frac{L}{2} \| x - x^k \|_2^2 \right\} \]

\[ \| \nabla f(x) - \nabla f(y) \| \leq L \| y - x \| \]

\[ L \text{ is a global worst-case constant} \]

\[ f(x) \leq f(x^k) + \nabla f(x^k)^T(x - x^k) + \frac{L}{2} \| x - x^k \|_2^2 \]
How can we better adapt to the local geometry?

Line-search:

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

Local quadratic upper bound

\[ Q_{L_k}(x, x^k) \]

\[ L \text{ is a global worst-case constant} \]

\[ \| \nabla f(x) - \nabla f(y) \| \leq L \| y - x \| \]

Appplies only locally

\[ f(x) \leq f(x^k) + \nabla f(x^k)^T (x - x^k) + \frac{L_k}{2} \| x - x^k \|_2^2 \]
How can we better adapt to the local geometry?

Variable metric:

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

\[ \| \nabla f(x) - \nabla f(y) \| \leq L \| y - x \| \]

L is a global worst-case constant
*UniXGrad: A first order universal adaptive algorithm.

\[
\min_{x \in \mathcal{K}} f(x)
\]

Motivation behind UniXGrad

Is it possible to get optimal rates for constrained convex optimization for \( f \in F_{L}^{2,1} \), without knowing the Lipschitz constant?

### UniXGrad [3]

1. Set \( y^0 \in \mathcal{K} \), diameter \( D \), weight \( \alpha^k \), learning rate \( \{\eta_k\} \in [T] \).
2. Define \( d_{\phi}(x, y) = \phi(x) - \phi(y) - \langle \nabla \phi(y), x - y \rangle \).
3. For \( k = 0, 1, \ldots \), iterate

\[
\begin{align*}
    x^{k+1} &= \arg \min_{x \in \mathcal{K}} \alpha^k \langle x, M^k \rangle + \frac{1}{\eta_k} d_{\phi}(x, y^k) \\
    y^{k+1} &= \arg \min_{y \in \mathcal{K}} \alpha^k \langle y, g^k \rangle + \frac{1}{\eta_k} d_{\phi}(y, y^k)
\end{align*}
\]

\( M^k = \nabla f(\tilde{z}^k) \)

\( g^k = \nabla f(\tilde{x}^k) \)

• This is essentially the **Mirror-Prox** scheme [4], with an adaptive step size!
UniXGrad - Properties and convergence

**The notion of averaging**

Define $\alpha^k = k$ and following quantities:

\[
\tilde{x}^k = \frac{\alpha^k x^k + \sum_{i=1}^{k-1} \alpha^i x^i}{\sum_{i=1}^{k} \alpha^i}, \quad \tilde{z}^k = \frac{\alpha^k y^k + \sum_{i=1}^{k-1} \alpha^i x^i}{\sum_{i=1}^{k} \alpha^i}
\]

**Learning rate**

UniXGrad uses the following weights and learning rate:

\[
\eta_k = \frac{2D}{\sqrt{1 + \sum_{\tau=0}^{k} \alpha^{i\tau} \|g^i - M^i\|^2}}
\]

where $D^2 = \sup_{x,y \in \mathcal{K}} d_\phi(x,y)$ is diameter of the compact set $\mathcal{K}$ w.r.t. Bregman divergences.
Convergence rate of AcceleGrad

Assume that $f$ is convex and $f \in F_{L}^{1,1}$. Let $K$ be a convex set with bounded diameter $D$, and assume $x^* \in K$. Define $\tilde{x}^T = (\sum_{k=0}^{T-1} \alpha_k y^{k+1}) / (\sum_{k=0}^{T-1} \alpha_k)$. Then,

$$f(\tilde{x}^T) - \min_{x \in \mathbb{R}^d} f(x) \leq O \left( \frac{DG + LD^2}{T^2} \right)$$

If $f$ is only convex and $G$-Lipschitz, then

$$f(\bar{y}^T) - \min_{x \in \mathbb{R}^d} f(x) \leq O \left( GD / \sqrt{T} \right)$$
Assumptions A.2

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

Proximal-Newton update

Similar to classical newton, proximal-newton suggests the following update scheme using second order Taylor series expansion near \( x_k \).

\[
x^{k+1} := \arg \min_{x \in \mathbb{R}^p} \left\{ \frac{1}{2} (x - x^k)^T \nabla^2 f(x^k) (x - x^k) + \nabla f(x^k)^T (x - x^k) + g(x) \right\}.
\]

(3)
The proximal-Newton-type algorithm

<table>
<thead>
<tr>
<th>Proximal-Newton algorithm (PNA)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1.</strong> Given $x^0 \in \mathbb{R}^p$ as a starting point.</td>
</tr>
<tr>
<td><strong>2.</strong> For $k = 0, 1, \cdots$, perform the following steps:</td>
</tr>
<tr>
<td><strong>2.1.</strong> Evaluate an SDP matrix $H_k \approx \nabla^2 f(x^k)$ and $\nabla f(x^k)$.</td>
</tr>
<tr>
<td><strong>2.2.</strong> Compute $d^k := \text{prox}_{H_k^{-1}g} \left( x^k - H_k^{-1} \nabla f(x^k) \right) - x^k$.</td>
</tr>
<tr>
<td><strong>2.3.</strong> Update $x^{k+1} := x^k + \alpha_k d^k$.</td>
</tr>
</tbody>
</table>
The proximal-Newton-type algorithm

<table>
<thead>
<tr>
<th>Proximal-Newton algorithm (PNA)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1.</strong> Given $x^0 \in \mathbb{R}^p$ as a starting point.</td>
</tr>
<tr>
<td><strong>2.</strong> For $k = 0, 1, \cdots$, perform the following steps:</td>
</tr>
<tr>
<td><strong>2.1.</strong> Evaluate an SDP matrix $H_k \approx \nabla^2 f(x^k)$ and $\nabla f(x^k)$.</td>
</tr>
<tr>
<td><strong>2.2.</strong> Compute $d^k := \text{prox}_{H_k^{-1}g} \left( x^k - H_k^{-1} \nabla f(x^k) \right) - x^k$.</td>
</tr>
<tr>
<td><strong>2.3.</strong> Update $x^{k+1} := x^k + \alpha_k d^k$.</td>
</tr>
</tbody>
</table>

**Remark**

- $H_k \equiv \nabla^2 f(x^k) \implies$ proximal-Newton algorithm.
- $H_k \approx \nabla^2 f(x^k) \implies$ proximal-quasi-Newton algorithm.
- A generalized prox-operator: $\text{prox}_{H_k^{-1}g} \left( x^k + H_k^{-1} \nabla f(x^k) \right)$. 
Convergence analysis

Theorem (Global convergence [6])

Assume generalized-prox subproblem is solved exactly for the algorithm and there exists $\mu > 0$ such that $H_k \succeq \mu I$ for all $k \geq 0$. Then;

$$\{x^k\}_{k \geq 0} \text{ globally converges to a solution } x^* \text{ of (2).}$$
Convergence analysis

**Theorem (Global convergence [6])**

Assume generalized-prox subproblem is solved exactly for the algorithm and there exists $\mu > 0$ such that $H_k \succeq \mu I$ for all $k \geq 0$. Then;

\[
\{x^k\}_{k \geq 0} \text{ globally converges to a solution } x^* \text{ of (2)}.
\]

**Theorem (Local convergence [6])**

Assume generalized-prox subproblem is solved exactly for the algorithm and there exists $0 < \mu \leq L_2 < +\infty$ such that $\mu I \preceq H_k \preceq L_2 I$ for all sufficiently large $k$. Then;

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

\[
\lim_{k \to +\infty} \frac{\| (H_k - \nabla^2 f(x^*)) (x^{k+1} - x^k) \|}{\| x^{k+1} - x^k \|} = 0,
\]

then $\{x^k\}$ locally converges to $x^*$ at a super linear rate.
*How to compute the approximation $H_k$?

How to update $H_k$?

Matrix $H_k$ can be updated by using low-rank updates.

- **BFGS update**: maintain the Dennis-Moré condition and $H_k \succ 0$.

  $$H_{k+1} := H_k + \frac{y_k y_k^T}{s_k^T y_k} - \frac{H_k s_k s_k^T H_k}{s_k^T H_k s_k}, \quad H_0 := \gamma I, \quad (\gamma > 0).$$

  where $y_k := \nabla f(x^{k+1}) - \nabla f(x^k)$ and $s_k := x^{k+1} - x^k$.

- **Diagonal+Rank-1 [2]**: computing PN direction $d^k$ is in polynomial time, but it does not maintain the Dennis-Moré condition:

  $$H_k := D_k + u_k u_k^T, \quad u_k := (s_k - H_0 y_k)/\sqrt{(s_k - H_0 y_k)^T y_k},$$

  where $D_k$ is a positive diagonal matrix.
Pros and cons

Pros

▶ Fast local convergence rate (super-linear or quadratic)
▶ Numerical robustness under the inexactness/noise ([6]).
▶ Well-suited for problems with many data points but few parameters. For example,

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

where \( \ell_j \) is twice continuously differentiable and convex, \( g \in \mathcal{F}_{\text{prox}} \), \( p \ll n \).
Pros and cons

Pros

▶ **Fast local convergence rate** (super-linear or quadratic)
▶ **Numerical robustness** under the inexactness/noise ([6]).
▶ Well-suited for problems with *many* data points but *few* parameters. For example,

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

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

Cons

▶ **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**.
▶ Requires strict conditions for global/local convergence analysis.
Example 1: Sparse logistic regression

**Problem (Sparse logistic regression)**

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

\[
P(b|a, x, \mu) = \frac{1}{1 + e^{-b(x^T a + \mu)}},
\]

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

**Optimization formulation**

\[
\min_{x \in \mathbb{R}^p} \left\{ \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(b_i(x_i^T x + \mu)) + \rho \|x\|_1 \right\},
\]

where \( a_i \) is the \( i \)-th row of data matrix \( A \) in \( \mathbb{R}^{n \times p} \), \( \rho > 0 \) is a regularization parameter, and \( \ell \) 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

Parameters

- Tolerance $10^{-6}$.
- L-BFGS memory $m = 50$.
- Ground truth: Get a high accuracy approximation of $x^*$ and $f^*$ by TFOCS with tolerance $10^{-12}$. 
Example: Sparse logistic regression - Numerical results

\[ \frac{F(x^k) - F^*}{F^*} \text{ in log scale} \]

- Pure Newton
- Quasi-Newton with BFGS
- Quasi-Newton with L-BFGS
- Accelerated gradient method
- Line Search AGD with adaptive restart

Number of iterations

Time (s)

\[ \frac{F(x^k) - F^*}{F^*} \text{ in log scale} \]

- Pure Newton
- Quasi-Newton with BFGS
- Quasi-Newton with L-BFGS
- Accelerated gradient method
- Line Search AGD with adaptive restart

Time (s)
**Example 2: \(\ell_1\)-regularized least squares**

**Problem (\(\ell_1\)-regularized least squares)**

Given \(A \in \mathbb{R}^{n \times p}\) and \(b \in \mathbb{R}^n\), solve:

\[
F^* := \min_{x \in \mathbb{R}^p} \left\{ F(x) := \frac{1}{2} \|Ax - b\|_2^2 + \rho \|x\|_1 \right\},
\]

where \(\rho > 0\) is a regularization parameter.

**Complexity per iterations**

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

**Synthetic data generation**

- \(A := \text{randn}(n, p)\) - standard Gaussian \(\mathcal{N}(0, I)\).
- \(x^*\) is a \(s\)-sparse vector generated randomly.
- \(b := Ax^* + \mathcal{N}(0, 10^{-3})\).
Example 2: $\ell_1$-regularized least squares - Numerical results - Trial 1

**Parameters:** $n = 750, p = 2000, s = 200, \rho = 1$
Example 2: $\ell_1$-regularized least squares - Numerical results - Trial 2

**Parameters:** $n = 750, p = 2000, s = 200, \rho = 1$
Stochastic convex composite minimization

Problem (Mathematical formulation)

Consider the following constrained convex minimization problem:

\[ F^* = \min_{x \in \mathbb{R}^p} \left\{ F(x) := \mathbb{E}[h(x, \theta)] + g(x) \right\} \]

- \( \theta \) is a random vector whose probability distribution is supported on set \( \Theta \).
- The solution set \( S^* := \{ x^* \in \text{dom}(F) : F(x^*) = F^* \} \) is nonempty.
- \( h(x, \theta) \in F_{L, \theta}^{1,1}(\mathbb{R}^p) \) and \( f = \mathbb{E}[h(x, \theta)] \in F_{L}^{1,1}(\mathbb{R}^p) \).
- \( g \in \mathcal{F}_{\text{prox}}(\mathbb{R}^p) \).

- The goal is to extend the methods to proximal case where \( g \neq 0 \).
Stochastic proximal gradient method

**Stochastic proximal gradient method (SPG)**

1. Choose $x^0 \in \mathbb{R}^p$ and $(\gamma_k)_{k \in \mathbb{N}} \in ]0, +\infty[^{\mathbb{N}}$.
2. For $k = 0, 1, \ldots$ perform:

   $$x^{k+1} = \text{prox}_{\gamma_k g}(x^k - \gamma_k G(x^k, \theta_k)).$$

- $\{\theta_k\}$: jointly independent random variables.
- $G(x^k, \theta_k)$: an unbiased estimate of the full gradient:

  $$\mathbb{E}[G(x^k, \theta_k)] = \nabla f(x^k).$$
Stochastic proximal gradient method

<table>
<thead>
<tr>
<th>Stochastic proximal gradient method (SPG)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Choose $\mathbf{x}^0 \in \mathbb{R}^p$ and $(\gamma_k)_{k \in \mathbb{N}} \in ]0, +\infty[^{\mathbb{N}}$.</td>
</tr>
<tr>
<td>2. For $k = 0, 1, \ldots$ perform:</td>
</tr>
<tr>
<td>$\mathbf{x}^{k+1} = \text{prox}_{\gamma_k g} (\mathbf{x}^k - \gamma_k G(\mathbf{x}^k, \theta_k))$.</td>
</tr>
</tbody>
</table>

- $\{\theta_k\}$: jointly independent random variables.
- $G(\mathbf{x}^k, \theta_k)$: an unbiased estimate of the full gradient:
  $$\mathbb{E}[G(\mathbf{x}^k, \theta_k)] = \nabla f(\mathbf{x}^k).$$

Remark
- Cost of computing $G(\mathbf{x}^k, \theta_k)$ is usually much cheaper than $\nabla f(\mathbf{x}^k)$. 
Convergence analysis

Assumption A4.

(i) Bounded variance: \( \mathbb{E}_\theta[\|G(x, \theta) - \nabla f(x)\|^2] \leq \sigma^2 \)

(ii) The step size \((\gamma_k)_{k \in \mathbb{N}} \in \ell^2(\mathbb{N}) \setminus \ell^1(\mathbb{N})\), i.e.,

\[
\sum_{k=0}^{\infty} \gamma_k = \infty \text{ and } \sum_{k=0}^{\infty} \gamma_k^2 < +\infty.
\]

Theorem (Ergodic convergence)

Assumptions:

- The sequence \( \{x^k\}_{k \geq 0} \) is generated by SPG.
- Assumption A4. is satisfied and the set of solutions is non-empty.

Conclusion:

- Define \( \hat{x}^s = \left( \sum_{k=0}^{s} \gamma_k x^k \right) / \sum_{k=0}^{s} \gamma_k \), then

\[
\mathbb{E} F(\hat{x}^s) - F(x^*) \leq \left( 0.5 \|x^0 - x^*\|^2 + \sigma^2 \sum_{k=0}^{\infty} \gamma_k^2 \right) / \sum_{k=0}^{s} \gamma_k.
\]
Convergence analysis

Assumption A4.

(i) **Bounded variance:** \( \mathbb{E}_\theta [\| G(x, \theta) - \nabla f(x) \|^2] \leq \sigma^2 \)

(ii) The step size \( (\gamma_k)_{k \in \mathbb{N}} \in \ell^2(\mathbb{N}) \setminus \ell^1(\mathbb{N}) \), i.e.,

\[
\sum_{k=0}^\infty \gamma_k = \infty \quad \text{and} \quad \sum_{k=0}^\infty \gamma_k^2 < +\infty.
\]

Theorem (Non-ergodic convergence [14])

**Assumptions:**

- The sequence \( \{x^k\}_{k \geq 0} \) is generated by SPG.
- Assumption A4(i). is satisfied and \( \gamma_k \sim 1/(k + 1) \).
- \( f \) is \( \mu \)-strongly convex.

**Conclusion:**

- 1/k rate is obtained:

  \[
  \mathbb{E}\|x^k - x^*\|^2 = \mathcal{O}(1/k).
  \]

- If \( F \) is \( R \)-smooth, i.e. \( F(x) - F(x^*) \leq R\|x - x^*\|^2 \), then

  \[
  \mathbb{E}F(x^k) - F(x^*) = \mathcal{O}(1/k).
  \]
Composite optimization with finite sums

\[ F^* := \min_{x \in \text{dom}(F)} \left\{ F(x) := \frac{1}{m} \sum_{k=1}^{m} f_k(x) + g(x) \right\}, \]  
\hspace{1cm} (7)

- \( f_k \in \mathcal{F}_{L_k}^{1,1}(\mathbb{R}^p) \) and \( f = \frac{1}{k} \sum_{k=1}^{m} f_k \in \mathcal{F}_{L}^{1,1}(\mathbb{R}^p) \).
- \( g \in \mathcal{F}_{\text{prox}}(\mathbb{R}^p) \).

Why is stochastic minimization?

- \( f(x) = \mathbb{E}_{j} f_j(x) \) where \( \mathbb{P}(j = k) = 1/m \).
- Computation \( \nabla f(x) = \frac{1}{m} \sum_{k=1}^{m} \nabla f_k(x) \) is expensive when \( m \) is large.
- Examples: Portfolio optimization, SVM.
**Large scale problems**

**Definition (Recall)**

\(\nabla f_j\) is called the **stochastic gradient** of \(f\), and it is **unbiased** estimate, i.e.

\[
\mathbb{E}_j \nabla f_j(x) = \sum_{i=1}^{m} P(j = k) \nabla f_k(x) = \frac{1}{m} \sum_{i=1}^{m} \nabla f_k(x) = \nabla f(x)
\]

**Example**

Define \(f(x) = \frac{1}{2m} \|Ax - b\|^2\) with \(b \in \mathbb{R}^m\). To find **stochastic gradient**, observe:

\[
f(x) = \frac{1}{2m} \sum_{k=1}^{m} |a_k^T x - b_k|^2
\]

Thus,

\[
f_j(x) = \frac{1}{2} |a_j^T x - b_j|^2 \quad \text{with} \quad \nabla f_j(x) = (a_j^T x - b_j)a_j
\]
Large scale problems

Definition (Recall)

- $\nabla f_j$ is called the stochastic gradient of $f$, and it is unbiased estimate, i.e.

$$
\mathbb{E}_j \nabla f_j(x) = \sum_{i=1}^{m} P(j = k) \nabla f_k(x) = \frac{1}{m} \sum_{i=1}^{m} \nabla f_k(x) = \nabla f(x)
$$

Example

Similarly, one can find stochastic gradient of

1. $f(x) = \frac{1}{m} \sum_{i=1}^{m} \log(1 + \exp(-b_i a_i^T x))$ where $a_i \in \mathbb{R}^p$, $b_i = \pm 1$.
2. $f(x) = \frac{1}{2} x^T Q x$ where $Q$ is positive semidefinite matrix.
3. $f(x) = \frac{1}{m} \sum_{i=1}^{m} (a_i^T x - \bar{b})^2$ where $a_i \in \mathbb{R}^p$, $\bar{b} \in \mathbb{R}$. 
**Stochastic proximal gradient algorithm for the finite sum problem**

<table>
<thead>
<tr>
<th>Stochastic proximal gradient algorithm (SPG) [4, 14]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Choose $x^0 \in \mathbb{R}^p$ as a starting point and $\gamma_0 &gt; 0$.</td>
</tr>
<tr>
<td>2. For $k = 0, 1, \ldots$, perform:</td>
</tr>
</tbody>
</table>
| \[
| \begin{align*}
| \text{Pick } i_k & \in \{1, \ldots, m\} \text{ uniformly at random} \\
| x^{k+1} & := \text{prox}_{\gamma_k g} \left( x^k - \gamma_k \nabla f_{i_k} (x^k) \right), \\
| \end{align*}
| \] |
| where $\gamma_k \in (0, 1/L]$ is a given step size aka **learning rate** |

**Properties**

- Cheap per iteration complexity (one component gradient and one prox).
- Decreasing stepsize such that $(\gamma_k)_{k \in \mathbb{N}} \in \ell^2(\mathbb{N}) \setminus \ell^1(\mathbb{N})$. 
Convergence analysis

Assumption A4.

(i) The variance is bounded: \( \mathbb{E}_j [||\nabla f_j(x) - \nabla f(x)||^2] \leq \sigma^2 \)

(ii) The step size \((\gamma_k)_{k \in \mathbb{N}} \in \ell^2(\mathbb{N}) \setminus \ell^1(\mathbb{N})\), i.e.,

\[
\sum_{k=0}^{\infty} \gamma_k = \infty \quad \text{and} \quad \sum_{k=0}^{\infty} \gamma_k^2 < +\infty.
\]

Theorem (Ergodic convergence)

Assumptions:

- The sequence \( \{x^k\}_{k \geq 0} \) is generated by SPG.
- Assumption A4. is satisfied and the set of solutions is non-empty.

Conclusion:

- Define \( \hat{x}^s = \left( \frac{\sum_{k=0}^{s} \gamma_k x^k}{\sum_{k=0}^{\infty} \gamma_k} \right) \), then

\[
\mathbb{E} F(\hat{x}^s) - F(x^*) \leq \left( 0.5 ||x^0 - x^*||^2 + \sigma^2 \sum_{k=0}^{\infty} \gamma_k^2 \right) / \sum_{k=0}^{\infty} \gamma_k.
\]
Convergence analysis

Assumption A4.

(i) The variance is bounded: \( \mathbb{E}_j [\| \nabla f_j(x) - \nabla f(x) \|^2] \leq \sigma^2 \)

(ii) The step size \((\gamma_k)_{k \in \mathbb{N}} \in \ell^2(\mathbb{N}) \setminus \ell^1(\mathbb{N})\), i.e.,

\[
\sum_{k=0}^{\infty} \gamma_k = \infty \quad \text{and} \quad \sum_{k=0}^{\infty} \gamma_k^2 < +\infty.
\]

Theorem (Non-ergodic convergence [14])

Assumptions:
- \( f \) is \( \mu \)-strongly convex and \( \gamma_k \sim 1/(k + 1) \).

Conclusion:
- 1\( \big/ k \) rate is obtained:
  \[
  \mathbb{E} \| x^k - x^* \|^2 = \mathcal{O}(1/k).
  \]
- If \( F \) is \( R \)-smooth, i.e. \( F(x) - F(x^*) \leq R \| x - x^* \|^2 \), then
  \[
  \mathbb{E} F(x^k) - F(x^*) = \mathcal{O}(1/k).
  \]
### Comparisons

▶ **SPG vs PG**

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>$\gamma_k \to 0$</th>
<th>Strong convexity</th>
<th>Convergence</th>
<th>Rate</th>
<th>#\ #grad/Iter.</th>
</tr>
</thead>
<tbody>
<tr>
<td>PG</td>
<td>No</td>
<td>No</td>
<td>Ergodic</td>
<td>$1/k$</td>
<td>$m$</td>
</tr>
<tr>
<td>SPG</td>
<td>Yes</td>
<td>No</td>
<td>Ergodic</td>
<td>$1/\sum_{j=0}^{k-1} \gamma_j$</td>
<td>$1$</td>
</tr>
<tr>
<td>PG</td>
<td>No</td>
<td>Yes</td>
<td>Non-Ergodic</td>
<td>Linear</td>
<td>$m$</td>
</tr>
<tr>
<td>SPG</td>
<td>Yes</td>
<td>Yes</td>
<td>Non-Ergodic</td>
<td>$1/k$</td>
<td>$1$</td>
</tr>
</tbody>
</table>

▶ PG = Proximal gradient, aka Forward-backward.
▶ SPG = Stochastic Proximal gradient.

▶ **Cheap per iteration complexity + slow convergence rate.**

> SPG $\rightarrow$ **Large scale problems + low accurate of solution**

▶ Decreasing learning rate due to variance, hence slower convergence.

Solution: Variance reduction technique (see Prox-SVRG)
Example: $\ell_1$-regularized least squares revisited

**Problem ($\ell_1$-regularized least squares)**

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

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

where $\rho > 0$ is a regularization parameter. \hspace{1cm} (9)

**Complexity per iterations**

- Evaluating $\nabla f_j(x^k) = (a_j^T x^k - b)a_j$ requires one $a_j^T x$ and one $\lambda a_j$.
- One soft-thresholding operator $\text{prox}_{\lambda g}(x) = \text{sign}(x) \otimes \max\{|x| - \kappa, 0\}$.
- **Optional**: Evaluating $L_{\max} = \max_{1 \leq k \leq m} \|a_k\|^2$ - via iterations.

**Synthetic data generation**

- $A := \text{randn}(m, p)$ - standard Gaussian $\mathcal{N}(0, I)$.
- $x^\star$ is a sparse vector generated randomly.
- $b := Ax^\star + \mathcal{N}(0, 10^{-3})$. 
Example: $\ell_1$-regularized least squares revisited - Numerical test

Ergodic convergence in function value

Deterministic: $\gamma_t = 0.25/L$

Stochastic: $\gamma_t = 100/(100 + t)$
*Accelerated SPG I*

**Accelerated SPG (AccSPG)**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>0.</strong></td>
<td>$0 \leq \mu$-strong convexity of $F$.</td>
</tr>
<tr>
<td><strong>1.</strong></td>
<td>Choose $y^0 = z^0 = 0$, $(\gamma_k)<em>{k \in \mathbb{N}}, (\alpha_k)</em>{k \in \mathbb{N}} \in \mathbb{R}^+ \cup {0}, \alpha_0 = 1, \gamma_0 = L + \mu$.</td>
</tr>
<tr>
<td><strong>2.</strong></td>
<td>For $k = 0, 1, \ldots$ perform:</td>
</tr>
<tr>
<td><strong>2a.</strong></td>
<td>$x^{k+1} = (1 - \alpha_k)y^k + \alpha_kz^k$.</td>
</tr>
<tr>
<td><strong>2b.</strong></td>
<td>$y^{k+1} = \text{prox}_{g/\gamma_k} \left( x^{k+1} - \frac{1}{\gamma_k} G(x^{k+1}, \theta_k) \right)$.</td>
</tr>
<tr>
<td><strong>2c.</strong></td>
<td>$z^{k+1} = z^k - \frac{1}{\gamma_k \alpha_k + \mu} \left( \gamma_k (x^{k+1} - y^{k+1}) + \mu (z^k - x^{k+1}) \right)$.</td>
</tr>
</tbody>
</table>
Accelerated SPG I

Theorem (Convergence of AccSPG with strong convexity [15])

Define $\lambda_k = \prod_{j=1}^{k} (1 - \alpha_j)$ and $\lambda_0 = 1$. Let:

1. $f$ is $\mu$-strongly convex,
2. $\mathbb{E}[\|z^k - x^*\|^2] \leq D^2$,
3. $\mathbb{E}[\|G(x^k, \theta_k) - \nabla f(x^k)\|^2] \leq M^2$.
4. $\gamma_k = L + \frac{\mu}{\lambda_{k-1}}$ and $\alpha_k = \sqrt{\lambda_{k-1} + \frac{\lambda_{k-1}^2}{4}} - \frac{\lambda_{k-1}}{2}$.

Then,

$$\mathbb{E}[f(y^{k+1}) - f(x^*)] \leq \frac{2(L + \mu)D^2}{k^2} + \frac{6M^2}{\mu k}.$$ 

The accelerated technique can be used to reduce the error term related to $\mathbb{E}[\|z^k - x^*\|^2]$. 
### Accelerated SPG (AccSPG)

1. Choose $y^0 = z^0 = 0$, $(\gamma_k)_{k \in \mathbb{N}}$, $(\alpha_k)_{k \in \mathbb{N}} \in ]0, +\infty[^\mathbb{N}$, $\alpha_0 = 1$, $\gamma_0 = L$.
2. For $k = 0, 1, \ldots$ perform:
   2a. $x^{k+1} = (1 - \alpha_k)y^k + \alpha_kz^k$.
   2b. $y^{k+1} = \text{prox}_{g/\gamma_k}(x^{k+1} - \frac{1}{\gamma_k}G(x^{k+1}, \theta_k))$.
   2c. $z^{k+1} = z^k - \frac{1}{\alpha_k}(x^{k+1} - y^{k+1})$.

---

**Theorem (Convergence of AccSPG without strong convexity [15])**

Let:
1. $\mathbb{E}[\|z^k - x^*\|^2] \leq D^2$,
2. $\mathbb{E}[\|G(x^k, \theta_k) - \nabla f(x^k)\|^2] \leq M^2$,
3. $\gamma_k = c(k + 1)^{3/2} + L$ for a fixed $c > 0$, and $\alpha_k = 2/(k + 2)$.

Then,

$$\mathbb{E}[f(y^{k+1}) - f(x^*)] \leq \frac{3D^2L}{k^2} + \left(3D^2c + \frac{5M^2}{3c}\right)\frac{1}{\sqrt{k}}.$$
Stochastic proximal gradient algorithm with variance reduction (Prox-SVRG) [13]

Recall: Variance reduction techniques.

- Select the stochastic gradient $\nabla f_{i_k}$, and compute a gradient estimate

$$ r_k = \nabla f_{i_k}(x^k) - \nabla f_{i_k}(\tilde{x}) + \nabla f(\tilde{x}), $$

where $\tilde{x}$ is a good approximation of $x^*$.

- As $\tilde{x} \to x^*$ and $x^k \to x^*$,

$$ \nabla f_{i_k}(x^k) - \nabla f_{i_k}(\tilde{x}) + \nabla f(\tilde{x}) \to 0. $$

- Therefore,

$$ \mathbb{E}\left[ \left\| \nabla f_{i_k}(x^k) - \nabla f_{i_k}(\tilde{x}) + \nabla f(\tilde{x}) \right\|^2 \right] \to 0. $$
Stochastic proximal gradient algorithm with variance reduction (Prox-SVRG) [13]

1. Choose $\tilde{x}^0 \in \mathbb{R}^p$ as a starting point and $\gamma > 0$ and $n \in \mathbb{N}_+$.
2. For $s = 0, 1, 2 \cdots$, perform:
   2a. $\tilde{x} = \tilde{x}^s$, $\tilde{v} = \nabla f(\tilde{x})$, $x_0 = \tilde{x}$.
   2b. For $k = 0, 1, \cdots n - 1$, perform:
      \[
      \begin{align*}
      \{ & \text{Pick } i_k \in \{1, \ldots, m\} \text{ uniformly at random} \\
      & r_k = \nabla f_{i_k}(x_k) - \nabla f_{i_k}(\tilde{x}) + \tilde{v} \\
      & x^{k+1} := \text{prox}_{\gamma g}(x^k - \gamma r_k),\end{align*}
      \]
   2c. Update $\tilde{x}^s = \frac{1}{n} \sum_{j=0}^{n-1} x^j$.

Properties

- A multistage scheme to reduce the variance of the stochastic gradient.
- Possibility of constant learning rate.
- $m + 2n$ component gradient evaluations at each iteration.
Convergence analysis

Assumption A5.

(i) \( F = f + g \) is \( \mu \)-strongly convex
(ii) The learning rate \( 0 < \gamma < 1/(4L_{\text{max}}) \).
(iii) \( n \) is large enough such that

\[
\kappa = \frac{1}{\mu \gamma (1 - 4\gamma L_{\text{max}}) n} + \frac{4\gamma L_{\text{max}}(n + 1)}{(1 - 4\gamma L_{\text{max}}) n} < 1.
\]

Theorem

Assumptions:

- The sequence \( \{\tilde{x}^s\}_{k \geq 0} \) is generated by Prox-SVRG.
- Assumption A5. is satisfied.

Conclusion: Linear convergence is obtained:

\[
\mathbb{E}F(\tilde{x}^s) - F(x^*) \leq \kappa^s (F(\tilde{x}^0) - F(x^*)).
\]
Choice of $\gamma$ and $n$, and complexity

Chose $\gamma$ and $n$ such that $\kappa \in (0, 1)$:

For example

$$\gamma = 0.1/L_{\text{max}}, \quad n = 100(L_{\text{max}}/\mu) \implies \kappa \approx 5/6.$$ 

Complexity

$$\mathbb{E}F(\tilde{x}^{s}) - F(x^{*}) \leq \varepsilon \text{ when } s \geq \log(\kappa^{-1}) \log((F(\tilde{x}^{0}) - F(x^{*}))/\varepsilon)$$

Since at each stage needs $m + 2n$ component gradient evaluations, with $n = O(L_{\text{max}}/\mu)$, we get the overall complexity is

$$O\left((m + L_{\text{max}}/\mu) \log(1/\varepsilon)\right).$$
Example: $\ell_1$-regularized least squares revisited - Numerical test

SPG: $\gamma_t = \frac{100}{100 + t}$ and Prox-SVRG: $\gamma = \frac{0.1}{L_{\text{max}}}$
SVRG++ for nonstrongly convex objectives

Strong convexity assumption rules out many important applications.

Examples: Lasso, ridge regression, $\ell_1$ regularized logistic regression.

Idea: Increase the number of inner iterations linearly.

SVRG++ [1]

1. Choose $\tilde{x}_0 \in \mathbb{R}^p$ as a starting point, $m_0 \in \mathbb{N}^+$ as the inner loop parameter, $\gamma > 0$ and $n \in \mathbb{N}^+$.

2. For $s = 0, 1, 2, \ldots, n$, perform:

   2a. $\tilde{x}_s = \tilde{x}_s$, $\tilde{v}_s = \nabla f(\tilde{x}_s)$, $m_s = 2s m_0$.

   2b. For $k = 0, 1, \ldots, m_s - 1$, perform:

      Pick $i_k \in \{1, \ldots, m\}$ uniformly at random
      $r_k = \nabla f_{i_k}(x_{s+1}) - \nabla f_{i_k}(\tilde{x}) + \tilde{v}_s x_{k+1} = \text{prox}_{\gamma g}(x_{s+1} - \gamma r_k)$

   2c. Update $\tilde{x}_s = \frac{1}{m_s} \sum_{j=1}^{m_s} x_j$, $x_{s+1} = x_{m_s}$.

Properties

- Learning rate $\gamma < 1/(7L_{\max})$.
- Gradient complexity $O(S \cdot n + 2S \cdot m_0)$.
- Linear convergence as in the strongly convex case [1].
SVRG++ for nonstrongly convex objectives

**Strong convexity assumption** rules out many important applications.

▷ Examples: *Lasso, ridge regression, ℓ₁ regularized logistic regression.*

**Idea:** Increase the number of inner iterations linearly.
SVRG++ for nonstrongly convex objectives

Strong convexity assumption rules out many important applications.
▷ Examples: Lasso, ridge regression, ℓ₁ regularized logistic regression.

Idea: Increase the number of inner iterations linearly.

<table>
<thead>
<tr>
<th>SVRG++ [1]</th>
</tr>
</thead>
</table>
| 1. Choose \( \tilde{x}^0 \in \mathbb{R}^p \) as a starting point, \( m_0 \in \mathbb{N}_+ \) as the inner loop parameter, \( \gamma > 0 \) and \( n \in \mathbb{N}_+ \).
| 2. For \( s = 0, 1, 2 \cdots n \), perform:
| \( 2a. \) \( \tilde{x} = \tilde{x}^s, \quad \tilde{v} = \nabla f (\tilde{x}), \quad m_s = 2^s m_0. \)
| \( 2b. \) For \( k = 0, 1, \cdots m_s - 1 \), perform:
| \[ \begin{align*}
| \quad & \text{Pick } i_k \in \{1, \ldots, m\} \text{ uniformly at random} \\
| \quad & r_k = \nabla f_{i_k} (x_k^s) - \nabla f_{i_k} (\tilde{x}) + \tilde{v} \\
| \quad & x_{s+1} = \text{prox}_{\gamma g} \left( x_s^k - \gamma r_k \right), \\
| & \text{Update } \tilde{x}^s = \frac{1}{m_s} \sum_{j=1}^{m_s} x_j^s, \quad x_{s+1}^0 = x_{s+1}^{m_s}. 
| \end{align*} \] (11)
SVRG++ for nonstrongly convex objectives

**Strong convexity assumption** rules out many important applications.

Examples: *Lasso, ridge regression, \(\ell_1\) regularized logistic regression.*

**Idea:** Increase the number of inner iterations linearly.

**SVRG++ [1]**

<table>
<thead>
<tr>
<th>1. Choose (\tilde{x}^0 \in \mathbb{R}^p) as a starting point, (m_0 \in \mathbb{N}<em>+) as the inner loop parameter, (\gamma &gt; 0) and (n \in \mathbb{N}</em>+).</th>
</tr>
</thead>
<tbody>
<tr>
<td>2. For (s = 0, 1, 2 \cdots n), perform:</td>
</tr>
<tr>
<td>2a. (\tilde{x} = \tilde{x}^s, \quad \tilde{v} = \nabla f(\tilde{x}), \quad m_s = 2^s m_0).</td>
</tr>
<tr>
<td>2b. For (k = 0, 1, \cdots m_s - 1), perform:</td>
</tr>
</tbody>
</table>
| \[
\begin{align*}
    r_k &= \nabla f_{i_k}(x_k^s) - \nabla f_{i_k}(\tilde{x}) + \tilde{v} \\
    x_{s+1}^{k+1} &= \text{prox}_{\gamma g}(x_s^k - \gamma r_k),
\end{align*}
\]

(11)

2c. Update \(\tilde{x}^s = \frac{1}{m_s} \sum_{j=1}^{m_s} x_j^s\), \(x_{s+1}^0 = x_{s+1}^{m_s}\).

**Properties**

- **Learning rate** \(\gamma < 1/(7L_{\text{max}})\).
- **Gradient complexity** \(O(S \cdot n + 2^S \cdot m_0)\).
- **Linear convergence** as in the strongly convex case [1].
Improved svrg for non-strongly-convex or sum-of-non-convex objectives.  

*First-order methods in optimization*, volume 25.  

Unixgrad: A universal, adaptive algorithm with optimal guarantees for constrained optimization.  

Prox-method with rate of convergence o (1/t) for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems.  

A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems.  
References II

A quasi-Newton proximal splitting method.

Signal recovery by proximal forward-backward splitting.
Springer-Verlag, 2011.

[4] J, Duchi and Y. Singer,
Efficient Online and Batch Learning Using Forward Backward Splitting

On the convergence of the proximal point algorithm for convex minimization.

Proximal newton-type methods for convex optimization.
References

   *Introductory lectures on convex optimization: a basic course*, volume 87 of *Applied Optimization*.

   *Interior-point Polynomial Algorithms in Convex Programming*.

   Proximal algorithms.

    *Convex Analysis*, volume 28 of *Princeton Mathematics Series*.

    Monotone operators and the proximal point algorithm.

    Composite self-concordant minimization.
References IV

[13] L. Xiao and T. Zhang,  
A Proximal Stochastic Gradient Method with Progressive Variance Reduction,  

[14] L. Rosasco, S. Villa, B. C. Vũ,  
Stochastic Forward-Backward Splitting for Monotone Inclusions,  

Accelerated gradient methods for stochastic optimization and online learning. 

SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. 