Y.C. Huang 12/07/2021

Gradient Projection for Sparse Reconstruction

Application to Compressed Sensing and Other Inverse Problems

Outline

- Introduction
  - Compressed Sensing
  - Previous Algorithms
- Proposed Formulation
- Gradient Projection Algorithms
- Experiment Results
- Reference

Introduction

- Try to estimate $\mathbf{x}$ from observations

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}$$

where $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{y} \in \mathbb{R}^k$, $\mathbf{A}$ is an $k \times n$ matrix, $k < n$, $\mathbf{n}$ is white Gaussian noise.

Introduction

- Consider the unconstrained optimization problem

$$\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \tau\|\mathbf{x}\|_1$$

where τ is a nonnegative parameter.

- Other related convex optimization problems:
  - (QP) $\min_{\mathbf{x}} \|\mathbf{x}\|_1$ subject to $\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 \leq \epsilon$, where ϵ is a nonnegative parameter
  - (LP) $\min_{\mathbf{x}} \|\mathbf{x}\|_1$ subject to $\mathbf{y} = \mathbf{A}\mathbf{x}$

Compressed Sensing

- $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}$, where $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{y} \in \mathbb{R}^k$, $\mathbf{A}$ is an $k \times n$ matrix
  - $k \ll n$
  - Signal is sparse or approximately sparse in some orthonormal basis
- Signal can be recovered from far fewer samples
  - A powerful alternative to Shannon-Nyquist sampling

Compressed Sensing

- Matrix $\mathbf{A}$ has the form $\mathbf{A} = \mathbf{RW}$
- $\mathbf{R}$ is a low-ranked randomized sensing matrix
- $\mathbf{W}$ is the representation basis
    - e.g. a wavelet basis or Fourier basis

Previous Algorithms

- Most algorithms need to explicitly store $\mathbf{A} = \mathbf{RW}$, which is not suitable for high-dimensional cases.
- Iterative shrinkage/thresholding (IST) are not suitable if the nonzero components of $\mathbf{x}$ is significant
- Matching pursuit (MP) or Orthogonal MP (greedy method) is not designed for above problems, but is used to reconstruct $\mathbf{x}$ from $\mathbf{y} = \mathbf{Ax}$

Proposed Formulation

- The unconstrained optimization problem

$$\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \tau\|\mathbf{x}\|_1$$

where τ is a nonnegative parameter.

Proposed Formulation

Formulate as a Quadratic Program

- $\mathbf{x} = \mathbf{u} - \mathbf{v}, \mathbf{u} \geq \mathbf{0}, \mathbf{v} \geq \mathbf{0}$, where $u_i = (x_i)_+$ and $v_i = (-x_i)_+$
  - $(a)_+ = \max\{0, a\}$
- Bound constrained quadratic program (BCQP)

$$\begin{aligned} \min_{\mathbf{u},\mathbf{v}} \quad & \frac{1}{2}\|\mathbf{y} - \mathbf{A}(\mathbf{u} - \mathbf{v})\|_2^2 + \tau \mathbf{1}_n^T \mathbf{u} + \tau \mathbf{1}_n^T \mathbf{v} \\ \text{subject to} \quad & \mathbf{u} \geq \mathbf{0}, \mathbf{v} \geq \mathbf{0} \end{aligned}$$

Proposed Formulation

- In more standard BCQP form

$$\min_{\mathbf{z}} \quad \mathbf{c}^T\mathbf{z} + \frac{1}{2}\mathbf{z}^T\mathbf{B}\mathbf{z} \equiv F(\mathbf{z})$$

$$\text{subject to} \quad \mathbf{z} \geq \mathbf{0}$$

where $\mathbf{z} = \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix}, \mathbf{b} = \mathbf{A}^T\mathbf{y}, \mathbf{c} = \tau\mathbf{1}_{2n} + \begin{bmatrix} -\mathbf{b} \\ \mathbf{b} \end{bmatrix}$

and $\mathbf{B} = \begin{bmatrix} \mathbf{A}^T\mathbf{A} & -\mathbf{A}^T\mathbf{A} \\ -\mathbf{A}^T\mathbf{A} & \mathbf{A}^T\mathbf{A} \end{bmatrix}$

Gradient Projection Algorithms

In each iteration,

- First, choose some scalar parameter $\alpha^{(k)} > 0$ and set
  - $\mathbf{w}^{(k)} = \left(\mathbf{z}^{(k)} - \alpha^{(k)} \nabla F\left(\mathbf{z}^{(k)}\right)\right)_+$
- Then, choose a second parameter $\lambda^{(k)} \in [0,1]$ and set
  - $\mathbf{z}^{(k+1)} = \left(1 - \lambda^{(k)}\right)\mathbf{z}^{(k)} + \lambda^{(k)}\mathbf{w}^{(k)} = \mathbf{z}^{(k)} + \lambda^{(k)}\left(\mathbf{w}^{(k)} - \mathbf{z}^{(k)}\right)$

The GPSR-Basic Algorithm

- Define vector $\mathbf{g}^{(k)}$ by
  - $$\mathbf{g}_i^{(k)} = \begin{cases} \left(\nabla F\left(\mathbf{z}^{(k)}\right)\right)_i, & \text{if } \mathbf{z}_i^{(k)} > 0 \text{ or } \left(\nabla F\left(\mathbf{z}^{(k)}\right)\right)_i < 0 \\ 0, & \text{otherwise} . \end{cases}$$
- Initial guess
  - $$\alpha_0 = \frac{(\mathbf{g}^{(k)})^T \mathbf{g}^{(k)}}{(\mathbf{g}^{(k)})^T \mathbf{B} \mathbf{g}^{(k)}} \text{ (by letting } \nabla_\alpha F(\mathbf{z}^{(k)} - \alpha \mathbf{g}^{(k)}) = 0)$$
  - Confine α_0 to the interval $[\alpha_{\min}, \alpha_{\max}]$

The GPSR-Basic Algorithm

- Step 0 (initialization): Given $\mathbf{z}^{(0)}$, choose parameters $\beta \in (0,1)$ and $\mu \in (0,1/2)$; set $k = 0$.
- Step 1: Compute α_0 like last slide, and replace α_0 by $mid(\alpha_{\min}, \alpha_0, \alpha_{\max})$.
- Step 2 (backtracking line search): Choose $\alpha^{(k)}$ to be the first number in sequence $\alpha_0, \beta\alpha_0, \beta^2\alpha_0, \ldots,$ such that

$$
\begin{aligned}
&F\left(\left(\mathbf{z}^{(k)} - \alpha^{(k)}\nabla F\left(\mathbf{z}^{(k)}\right)\right)_+\right) \\
&\leq F\left(\mathbf{z}^{(k)}\right) - \mu \nabla F\left(\mathbf{z}^{(k)}\right)^T\left(\mathbf{z}^{(k)} - \left(\mathbf{z}^{(k)} - \alpha^{(k)}\nabla F\left(\mathbf{z}^{(k)}\right)\right)_+\right)
\end{aligned}
$$

  and set $\mathbf{z}^{(k+1)} = (\mathbf{z}^{(k)} - \alpha^{(k)}\nabla F(\mathbf{z}^{(k)}))_+$
- Step 3: Check termination or $k \leftarrow k+1$ and go to Step 1

The GPSR-BB Algorithm

- BB: Barzilai-Borwein
- Second order method, approximate Hessian of F at iteration k by $\eta^{(k)}I$
  - $\nabla F(\mathbf{z}^{(k)}) - \nabla F(\mathbf{z}^{(k-1)}) \approx \eta^{(k)}\left[\mathbf{z}^{(k)} - \mathbf{z}^{(k-1)}\right]$
  - $\mathbf{z}^{(k+1)} = \mathbf{z}^{(k)} - \left(\eta^{(k)}\right)^{-1} \nabla F\left(\mathbf{z}^{(k)}\right) = \mathbf{z}^{(k)} - \alpha^{(k)} \nabla F\left(\mathbf{z}^{(k)}\right)$
- In our problem,
  - $\nabla F(\mathbf{z}^{(k)}) - \nabla F(\mathbf{z}^{(k-1)}) = \mathbf{B}\left(\mathbf{z}^{(k)} - \mathbf{z}^{(k-1)}\right)$

The GPSR-BB Algorithm

- Step 0 (initialization): Given $\mathbf{z}^{(0)}$, choose parameters $\alpha_{\min}, \alpha_{\max}, \alpha^{(0)}$, and set $k = 0$.
- Step 1: Compute $\delta^{(k)} = \left(\mathbf{z}^{(k)} - \alpha^{(k)} \nabla F\left(\mathbf{z}^{(k)}\right)\right)_{+} - \mathbf{z}^{(k)}$
- Step 2: $\lambda^{(k)} = \operatorname{mid}\left\{0, \dfrac{\left(\delta^{(k)}\right)^{T} \nabla F(\mathbf{z}^{(k)})}{\left(\delta^{(k)}\right)^{T} \mathbf{B} \delta^{(k)}}, 1\right\}$ (by $\nabla_{\delta} F(\mathbf{z}^{(k)} + \lambda^{(k)}\delta) = 0$) and set $\mathbf{z}^{(k+1)} = \mathbf{z}^{(k)} + \lambda^{(k)} \delta^{(k)}$.
- Step 3 (update α): compute $\gamma^{(k)} = \left(\delta^{(k)}\right)^{T} \mathbf{B} \delta^{(k)}$;
- $$\alpha_i^{(k+1)} = \begin{cases} \left(\nabla F\left(\mathbf{z}^{(k)}\right)\right)_i, & \text{if } \gamma^{(k)} = 0 \\ \operatorname{mid}\left\{\alpha_{\min}, \dfrac{\|\delta^{(k)}\|_2^2}{\gamma^{(k)}}, \alpha_{\max}\right\}, & \text{otherwise}. \end{cases}$$
- Step 4: Check termination or $k \leftarrow k + 1$ and go to Step 1

Termination

- Choose some threshold *th* and for each iteration we define
  - $\mathcal{I}_k = \{i \,|\, z_i^{(k)} \neq 0\}$
  - $$\begin{aligned} \mathcal{C}_k = \{i \,|\, &(i \in \mathcal{I}_k \text{ and } i \notin \mathcal{I}_{k-1}) \\ &\text{or } (i \notin \mathcal{I}_k \text{ and } i \in \mathcal{I}_{k-1})\} \end{aligned}$$
- The algorithm terminates if
  - $|\mathcal{C}_k| / |\mathcal{I}_k| \leq th$
- This criteria takes account of
  - the nonzero indices of $\mathbf{z}$
  - how much these changed in recent iterations

Experiment Results

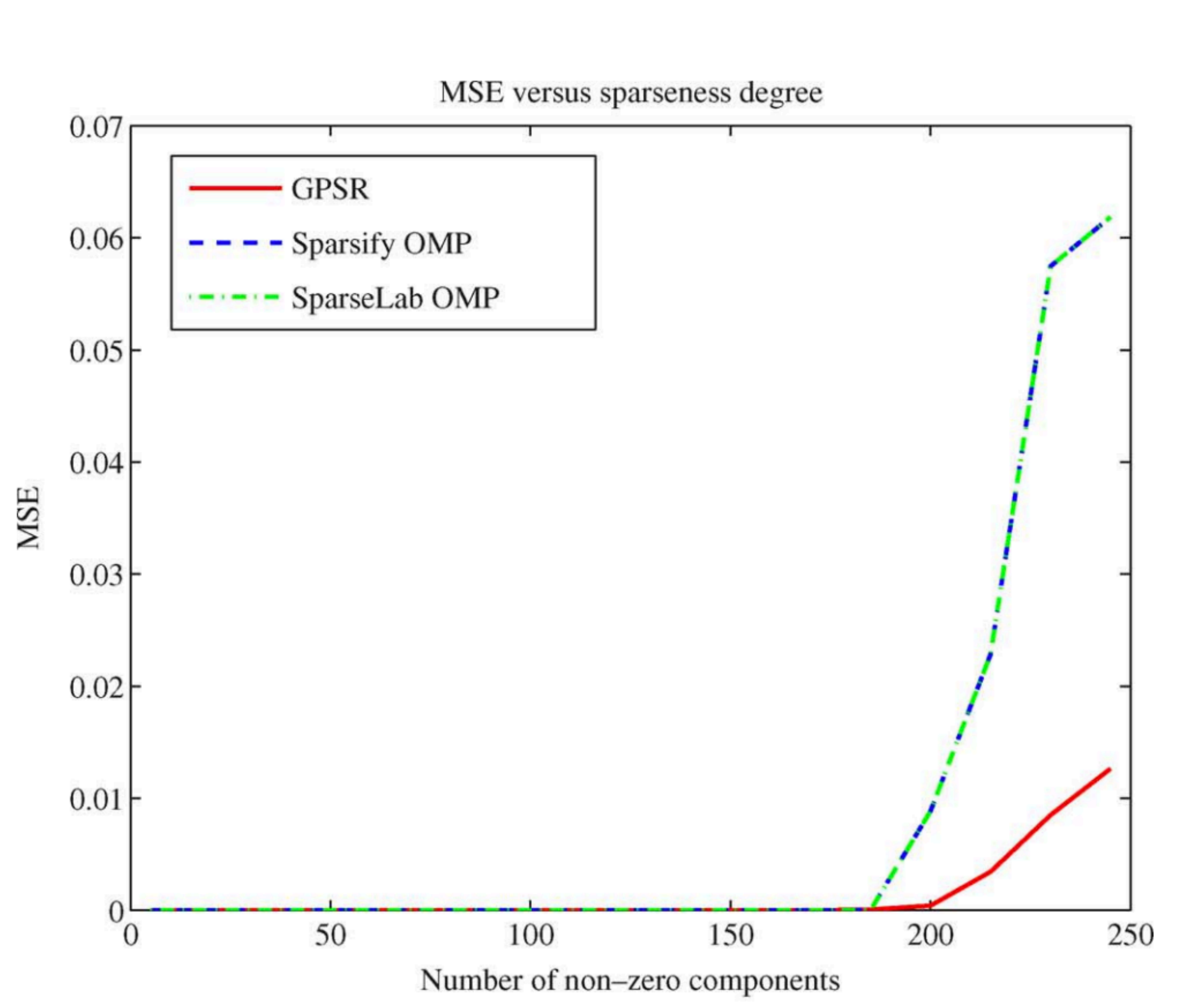
- Compressed Sensing
  - $n = 4096, k = 1024$, the original signal $\mathbf{x}$ contains 160 randomly placed nonzero components
  - white Gaussian noise with variance $\sigma^2 = 10^{-4}$
  - $\tau = 0.1 \|\mathbf{A}^T \mathbf{y}\|_\infty$

Experiment Results

TABLE I
CPU TIMES (AVERAGE OVER TEN RUNS) OF SEVERAL ALGORITHMS ON THE EXPERIMENT OF FIG. 1

Algorithm	CPU time (seconds)
GPSR-BB monotone	0.59
GPSR-BB nonmonotone	0.51
GPSR-Basic	0.69
GPSR-BB monotone + debias	0.89
GPSR-BB nonmonotone + debias	0.82
GPSR-Basic + debias	0.98
l1_ls	6.56
IST	2.76

Experiment Results



MSE $= (1/n)\|\hat{\mathbf{x}} - \mathbf{x}\|_2^2$, where $\hat{\mathbf{x}}$ is the estimator of $\mathbf{x}$

Conclusion

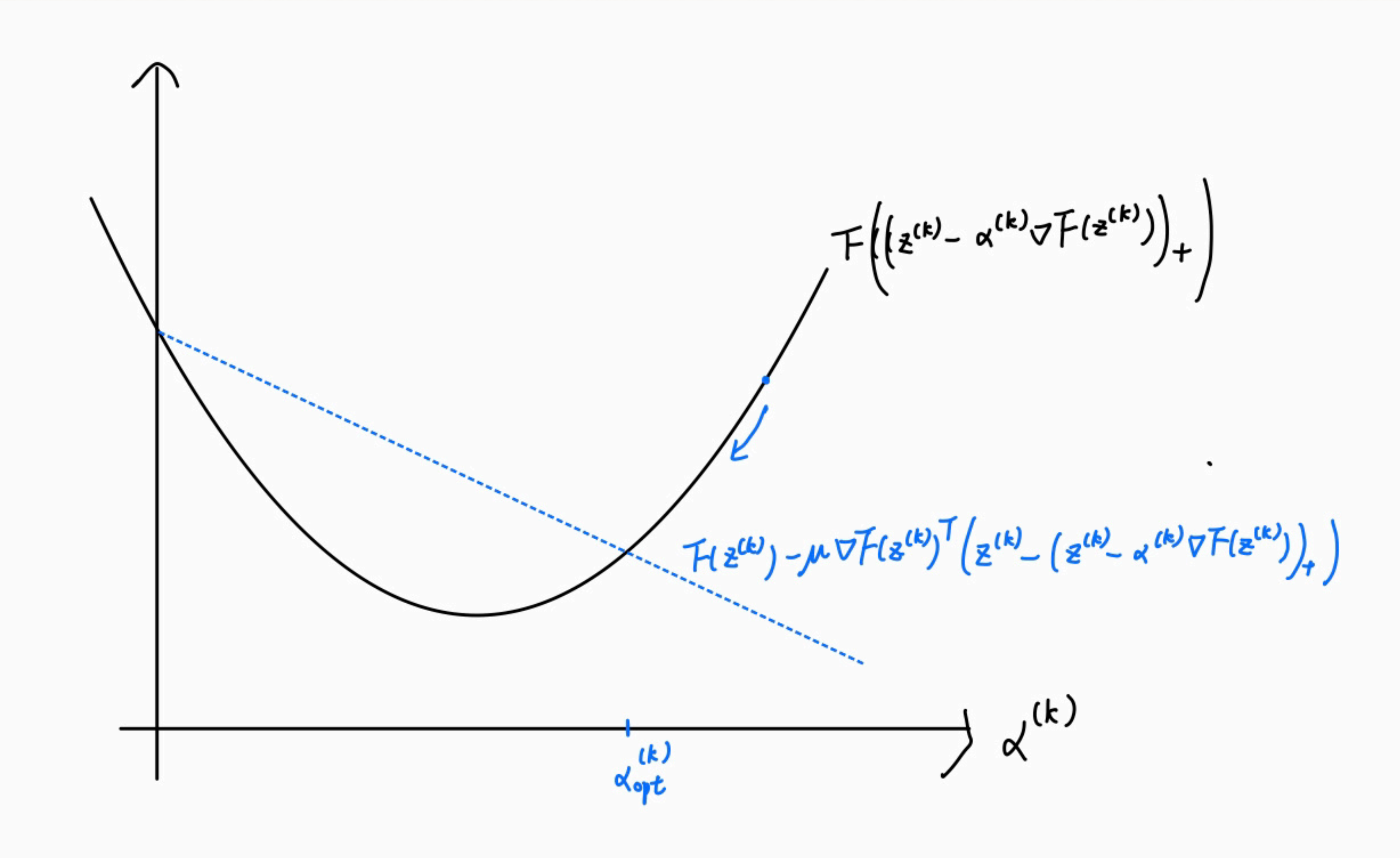
- Proposed algorithms for solving a quadratic programming of a class of convex nonsmooth unconstrained optimization
- It can be efficiently applied to CS problem, image reconstruction, and other inverse problems

Reference

- M. A. T. Figueiredo, R. D. Nowak and S. J. Wright, "Gradient Projection for Sparse Reconstruction: Application to Compressed Sensing and Other Inverse Problems," in IEEE Journal of Selected Topics in Signal Processing, vol. 1, no. 4, pp. 586-597, Dec. 2007, doi: 10.1109/JSTSP.2007.910281.
- DONOHO, David L. Compressed sensing. IEEE Transactions on information theory, 2006, 52.4: 1289-1306.
- Serafini, Thomas, Gaetano Zanghirati, and Luca Zanni. "Gradient projection methods for quadratic programs and applications in training support vector machines." *Optimization Methods and Software* 20.2-3 (2005): 353-378.

Debiasing

- **Fix** the zero components of previous result $\mathbf{x}_{GP}$
- Minimize $\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2$ using a CG algorithm
- The algorithm is terminated when
  - $\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 \leq \mathrm{tolD}\|\mathbf{y} - \mathbf{A}\mathbf{x}_{\mathbf{GP}}\|_2^2$



$F\left(\left(z^{(k)} - \alpha^{(k)} \nabla F(z^{(k)})\right)_+\right)$
$F(z^{(k)}) - \mu \nabla F(z^{(k)})^T \left(z^{(k)} - \left(z^{(k)} - \alpha^{(k)} \nabla F(z^{(k)})\right)_+\right)$
$\alpha^{(k)}$
$\alpha_{opt}^{(k)}$