Convex Relaxation Algorithms

Loris Cannelli
Soumendu Majee

ECE695: Sparse Representations and Signal Recovery
Purdue University

February 5, 2016
1 Problem Description and Motivations
   Introduction
   Convex Relaxation

2 Algorithms
   GPSR
   SPG
   TNIPM

3 Experimental Results

4 References
Introduction

• The $P_0$ problem that arises in different applications that require sparsity of the solution is given as:

$$\min_{x} \|x\|_0 \text{ subject to } y = Ax$$

(1)

with $A \in \mathbb{R}^{n \times m}, y \in \mathbb{R}^n$ and $x \in \mathbb{R}^m$

• The equality constraint $y = Ax$ is too strict because the vector $y$ might not be exactly represented by a few columns of $A$. So, we relax it as:

$$\min_{x} \|x\|_0 \text{ subject to } \|y - Ax\|_2^2 \leq \epsilon$$

(2)

with $\epsilon > 0$

• The problem can also be written as an unconstrained optimization problem as:

$$\min_{x} \frac{1}{2} \|y - Ax\|_2^2 + \tau \|x\|_0$$

(3)

for an appropriate choice of $\tau > 0$
Convex Relaxation

• The major issue with the $P_0$ problem:

$$\min_{x} \frac{1}{2} \| y - Ax \|_2^2 + \tau \| x \|_0 \tag{4}$$

is that, it is not convex.

• We can relax the $l_0$ norm to a $l_p$ norm where $0 < p < \infty$ as $P_p$:

$$\min_{x} \frac{1}{2} \| y - Ax \|_2^2 + \tau \| x \|_p \tag{5}$$

The closer $p$ is to 0, the closer the solution is to $P_0$.

• The smallest value of $p$ for which the problem $P_p$ is convex is 1. Thus the best convex relaxation here is the problem $P_1$:

$$\min_{x} \frac{1}{2} \| y - Ax \|_2^2 + \tau \| x \|_1 \tag{6}$$
Different forms of the $P_1$ problem

- There are several forms of the $P_1$ problem and all of them are equivalent for appropriate choices of the parameters $\tau, \epsilon, \delta > 0$
  
- The Unconstrained version:
  \[
  \min_x \frac{1}{2} \| \mathbf{y} - A\mathbf{x} \|^2_2 + \tau \| \mathbf{x} \|_1
  \] (7)

- Constraint on $\| \mathbf{y} - A\mathbf{x} \|^2_2$:
  \[
  \min_x \| \mathbf{x} \|_1 \text{ subject to } \| \mathbf{y} - A\mathbf{x} \|^2_2 \leq \epsilon
  \] (8)

- Constraint on $\| \mathbf{x} \|_1$:
  \[
  \min_x \| \mathbf{y} - A\mathbf{x} \|^2_2 \text{ subject to } \| \mathbf{x} \|_1 \leq \delta
  \] (9)

- In this presentation we will discuss and compare three methods that solve one of these variations of the $P_1$ problem.
Gradient Projection for Sparse Reconstruction (GPSR): Application to Compressed Sensing and Other Inverse Problems

- The paper by Figueiredo et al. [1] proposes a Gradient Projection algorithm to solve the Bound Constraint Quadratic Programming formulation for the unconstrained version of the $P_1$ problem:

$$\min_x \frac{1}{2} \|y - Ax\|_2^2 + \tau \|x\|_1$$  \hspace{1cm} (10)
Formulation as a Quadratic Program

• The original $P_1$ problem:

$$\min_x \frac{1}{2}\|y - Ax\|_2^2 + \tau\|x\|_1$$  (11)

• Split $x$ into two vectors:

$$x = u - v, \ u \geq 0, \ v \geq 0$$  (12)

• So, the $P_1$ problem can be rewritten as:

$$\min_{u,v} \frac{1}{2}\|y - A(u - v)\|_2^2 + \tau 1^T_m u + \tau 1^T_m v \ s.t. \ u \geq 0, \ v \geq 0$$  (13)
Formulation as a standard Bound Constraint Quadratic Program

- The problem can be written as a standard Bound Constraint Quadratic Program as:

\[
\min_{z} c^T z + \frac{1}{2} z^T B z \quad \text{subject to } z \geq 0
\]  

(14)

where, \( z = \begin{bmatrix} u \\ v \end{bmatrix} \), \( b = A^T y \), \( c = \tau 1_{2m} + \begin{bmatrix} -b \\ b \end{bmatrix} \), and

\[
B = \begin{bmatrix} A^T A & -A^T A \\ -A^T A & A^T A \end{bmatrix}
\]
Solving Using Gradient Projection Algorithm

- Define vector $g^k$ as:
  
  $$g^k_i = \begin{cases} 
  (\nabla F(z^k))_i, & \text{if } z^k_i > 0 \text{ or } (\nabla F(z^k))_i < 0 \\
  0, & \text{otherwise}
  \end{cases} \quad (15)$$

- Then the initial guess along the $g^k$ direction is calculated as:
  
  $$\alpha_0 = \arg\min_{\alpha \in \alpha} F(z^k - \alpha g^k) \quad (16)$$

  which can be computed explicitly as:

  $$\alpha_0 = \frac{(g^k)^T g^k}{(g^k)^T B g^k} \quad (17)$$

  To protect against too small or large values, $\alpha_0$ is restricted to the range $[\alpha_{\text{min}}, \alpha_{\text{max}}]$. This novel technique produces a better choice of $\alpha_0$. 
The complete GPSR algorithm

- **Step 0 (initialization):** Given $z^0$, choose parameters $\beta \in (0, 1)$, $\mu \in (0, 1/2)$ and $k = 0$.

- **Step 1:** Calculate $\alpha_0$ and replace it by $\text{mid}(\alpha_{\text{min}}, \alpha_0, \alpha_{\text{max}})$, where $\text{mid}(\cdot, \cdot, \cdot)$ is the middle value among of its three scalar arguments.

- **Step 2 (backtracking line search):** Choose $\alpha^k$ to be the first number in the sequence $\alpha_0, \beta \alpha_0, \beta^2 \alpha_0...$ such that

$$F((z^k - \alpha^k \nabla F(z^k))_+) \leq F(z^k) - \mu \nabla F(z^k)^T (z^k - (z^k - \alpha^k \nabla F(z^k))_+)$$

and then set $z^{k+1} = (z^k - \alpha^k \nabla F(z^k))_+$

- If stop criterion is met terminate, else go to **Step 1**. Stop criterion is given by:

$$\| \min(z, \nabla F(z)) \|_2 \leq \text{tol} \quad (18)$$
The paper by Berg et al [2] solves the basis pursuit denoise problem ($BP_\sigma$):

$$\min_x \|x\|_1 \text{ subject to } \|b - Ax\|_2 \leq \sigma$$  \hspace{1cm} (19)

They solve this problem by efficiently and approximately solving the following problem ($LS_\tau$) for a sequence of values of $\tau$:

$$\min_x \|b - Ax\|_2 \text{ subject to } \|x\|_1 \leq \tau$$  \hspace{1cm} (20)
Brief Approach

- The \((LS_\tau)\) problem is efficiently solved using the spectral projected gradient algorithm. The dual solution of \((LS_\tau)\) is then used to update \(\tau\) so that the next solution of \((LS_\tau)\) is closer to the solution of \((BP_\sigma)\).
- If \(x_\tau\) is the optimal solution of \((LS_\tau)\), then the function:
  \[
  \phi(\tau) = \|r_\tau\|_2 \quad \text{with} \quad r_\tau := b - Ax_\tau
  \]  
  gives the optimal value of \((LS_\tau)\) for each \(\tau \geq 0\)
- The derivative \(\phi'\) is given by \(-\lambda_\tau\), where \(\lambda_\tau \geq 0\) is the unique dual solution of \((LS_\tau)\) which can be found as a by-product of solving \((LS_\tau)\)
- Then Newton’s method is used to find a root of the nonlinear equation
  \[
  \phi(\tau) = \sigma
  \]  
  to get a sequence of regularization parameters \(\tau_k\) which converges to the optimal \(\tau_\sigma\) so that the solution of \((LS_\tau)\) and \((BP_\sigma)\) are same.
The Pareto Curve

- The Pareto curve is the optimal trade off between the $l_1$ norm of the solution and the $l_2$ norm of the residual in the $(LS_\tau)$ problem. It is the plot of the function $\phi(\tau)$ defined before.

Figure: A typical Pareto curve (solid line) showing two iterations of the Newton’s method
Newton’s Iteration

- The Newton’s method (or Newton-Raphson method) is an approach to iteratively find roots of equations. In this case is used to iteratively solve the equation:

\[ \phi(\tau) = \sigma \]  

(23)

- The update on \( \tau_k \) is given by:

\[ \tau_{k+1} = \tau_k + \frac{\sigma - \phi(\tau_k)}{\phi'(\tau_k)} \]  

(24)

- \( \phi(\tau_k) \) can be found from the solution \( x_{\tau_k} \) of \((LS_{\tau_k})\) as:

\[ \phi(\tau_k) = \|b - Ax_{\tau_k}\|_2 \]  

(25)

- The derivative \( \phi'(\tau_k) \) can be shown to be:

\[ \phi'(\tau_k) = -\lambda_{\tau_k} = - = \frac{\|A^T r_{\tau_k}\| + \infty}{\|r_{\tau_k}\|_2} \]  

(26)

where \( \lambda_{\tau_k} \) is the optimal dual variable for the problem: \((LS_{\tau_k})\) and \( r_{\tau_k} \) is its residual
Spectral Projected Gradient

• Define:

\[ P_\tau[c] = \{ \arg\min_x \|c - x\|_2 \text{ subject to } \|x\|_1 \leq 1 \} \quad (27) \]

• At each iteration the algorithm searches the projected gradient path \( P_\tau[x_l - \alpha g_l] \), where \( g_l \) is the current gradient of the function \( \|Ax - b\|_2^2 \)
Spectral Projected Gradient Algorithm

Input: $x$, $\tau$, $\delta$
Output: $x_\tau$, $r_\tau$

Set minimum and maximum step lengths $0 < \alpha_{\text{min}} < \alpha_{\text{max}}$.
Set initial step length $\alpha_0 \in [\alpha_{\text{min}}, \alpha_{\text{max}}]$ and sufficient descent parameter $\gamma \in (0, 1)$.
Set an integer linesearch history length $M \geq 1$.
Set initial iterates: $x_0 \leftarrow P[x], r_0 \leftarrow b - Ax_0, g_0 \leftarrow -A^Tr_0$.
Let $\ell \leftarrow 0$

\begin{algorithmic}
\State $\delta_\ell \leftarrow \|r_\ell\|_2 - (b^Tr_\ell - \tau \|g_\ell\|_\infty)/\|r_\ell\|_2$ \hfill [compute duality gap]
\State if $\delta_\ell < \delta$ then break \hfill [exit if converged]
\State $\alpha \leftarrow \alpha_\ell$ \hfill [initial step length]
\State \textbf{begin}
\State \hspace{1em} $\bar{x} \leftarrow P[x_\ell - \alpha g_\ell]$ \hfill [candidate linesearch iterate]
\State \hspace{1em} $\bar{r} \leftarrow b - A\bar{x}$ \hfill [update the corresponding residual]
\State \hspace{1em} if $\|\bar{r}\|_2^2 \leq \max_{j \in [0, \min\{k, M-1\}]} \|r_{\ell-j}\|_2^2 + \gamma(\bar{x} - x_\ell)^Tg_\ell$ then \hfill [exit linesearch]
\State \hspace{1em} \hspace{1em} break \hfill [exit linesearch]
\State \hspace{1em} else \hfill [decrease step length]
\State \hspace{1em} \hspace{1em} $\alpha \leftarrow \alpha/2$
\State \textbf{end}
\State $x_{\ell+1} \leftarrow \bar{x}$, $r_{\ell+1} \leftarrow \bar{r}$, $g_{\ell+1} \leftarrow -A^Tr_{\ell+1}$ \hfill [update iterates]
\State $\Delta x \leftarrow x_{\ell+1} - x_\ell$, $\Delta g \leftarrow g_{\ell+1} - g_\ell$
\State if $\Delta x^T\Delta g \leq 0$ then \hfill [Update the Barzilai-Borwein step length]
\State \hspace{1em} $\alpha_{\ell+1} \leftarrow \alpha_{\text{max}}$
\State else \hfill [initial step length]
\State \hspace{1em} $\alpha_{\ell+1} \leftarrow \min \{\alpha_{\text{max}}, \max \{\alpha_{\text{min}}, (\Delta x^T\Delta x)/(\Delta x^T\Delta g)\}\}$
\State $\ell \leftarrow \ell + 1$
\State \textbf{end}
\end{algorithmic}

return $x_\tau$, $r_\tau$
TNIPM: introduction

- The last approach we are going to describe is presented in "An Interior-Point Method for Large-Scale $l_1$-Regularized Least Squares" from Kim, Koh, Lustig, Boyd and Gorinevsk [3].
- This method is called Truncated Newton Interior-Point Method (TNIPM).
- The algorithm solves the aforementioned problem:

$$\min_{x \in \mathbb{R}^m} \|Ax - y\|_2^2 + \tau \|x\|_1$$  \hspace{1cm} (28)
The interior-point method is an approach for solving optimization problems of the following form:

\[
\begin{align*}
\min_{x \in \mathbb{R}^m} & \quad f_0(x) \\
\text{subject to} & \quad f_i(x) \leq 0, \quad i = 1, \ldots, p \\
& \quad Ax = b
\end{align*}
\]  

(29)

with each \( f_0, f_1, \ldots, f_p : \mathbb{R}^m \to \mathbb{R} \) convex and twice differentiable and \( A \in \mathbb{R}^{n \times m} \).

We further assume that:

- the problem has at least a solution \( x^* \) with optimal value \( f^* \)
- Slater’s constraint qualifications hold
Interior-point method: introduction

- We now rewrite problem (29) in an equivalent way, making the inequality constraints implicit in the objective:

\[
\min_{x \in \mathbb{R}^m} f_0(x) + \sum_{i=1}^{p} I_-(f_i(x))
\]

subject to \( Ax = b \)

- \( I_-(\cdot) : \mathbb{R} \to \mathbb{R} \) is the indicator function for the nonpositive reals:

\[
I_-(u) = \begin{cases} 
0, & \text{if } u \leq 0 \\
+\infty, & \text{if } u > 0 
\end{cases}
\]
We now rewrite problem (29) in an equivalent way, making the inequality constraints implicit in the objective:

\[
\min_{x \in \mathbb{R}^m} f_0(x) + \sum_{i=1}^{p} I_-(f_i(x))
\]

subject to \( Ax = b \)

\( I_-(\cdot) : \mathbb{R} \to \mathbb{R} \) is the indicator function for the nonpositive reals:

\[
I_-(u) = \begin{cases} 
0, & \text{if } u \leq 0 \\
+\infty, & \text{if } u > 0 
\end{cases}
\]

Note that the indicator function is convex, but not differentiable.
Interior-point method: main idea

- The key idea is to approximate the $I_-(\cdot)$ with the following function $\tilde{I}_-(\cdot): -\mathbb{R}^+ \rightarrow \mathbb{R}$:

$$\tilde{I}_-(z) = -(1/t)\log(-u),$$  \hspace{1cm} (32)

where $t > 0$ is a tunable parameter.
Interior-point method: main idea

- The key idea is to approximate the $l_\cdot(\cdot)$ with the following function $\tilde{l}_\cdot(\cdot): -\mathbb{R}_+^+ \to \mathbb{R}$:
  \[
  \tilde{l}_\cdot(z) = -(1/t) \log(-u),
  \]
  where $t > 0$ is a tunable parameter
- $\tilde{l}(\cdot)$ is convex and differentiable
**Interior-point method: main idea**

- The key idea is to approximate the $I_-(\cdot)$ with the following function $\tilde{I}_-(\cdot): -\mathbb{R}_{++} \rightarrow \mathbb{R}$:
  $$\tilde{I}_-(z) = -(1/t) \log(-u),$$
  \hspace{1cm} (32)

  where $t > 0$ is a tunable parameter
- $\tilde{I}(\cdot)$ is convex and differentiable

**Figure:** The dashed lines show the function $I_-(u)$ and the solid curves show $\tilde{I}_-(u)$ for $t = 0.5, 1, 2$. As $t$ increases, the approximation becomes more accurate [4].
Interior point method: main idea

- By using \( \tilde{I}_- (\cdot) \) we get the following approximation of the original problem:

\[
\min_{x \in \mathbb{R}^m} \quad tf_0(x) - \sum_{i=1}^{p} \log(-f_i(x))
\]

subject to \( Ax = b \)
Interior point method: main idea

- By using \( \tilde{f}(\cdot) \) we get the following approximation of the original problem:

\[
\min_{x \in \mathbb{R}^m} \quad tf_0(x) - \sum_{i=1}^{p} \log(-f_i(x))
\]

subject to \( Ax = b \) \hspace{1cm} (33)

- This is a convex and differentiable problem that can be solved with classical optimization approaches
Interior point method: main idea

• By using \( \tilde{I}_- (\cdot) \) we get the following approximation of the original problem:

\[
\min_{x \in \mathbb{R}^m} \quad tf_0(x) - \sum_{i=1}^{p} \log(-f_i(x))
\]

subject to \( Ax = b \) \hspace{1cm} (33)

• This is a convex and differentiable problem that can be solved with classical optimization approaches

• Through duality theory it can be proved [4] that a solution \( x^*(t) \) of (33) satisfies:

\[
f_0(x^*(t)) - f^* \leq p/t, \hspace{1cm} (34)\]

which means that \( x^*(t) \) is at least \( p/t \)-suboptimal and \( x^*(t) \) converges to an optimal point as \( t \) increases
Interior point method: main idea

• By using $\tilde{I}_-(\cdot)$ we get the following approximation of the original problem:

$$\min_{x \in \mathbb{R}^m} t f_0(x) - \sum_{i=1}^{p} \log(-f_i(x))$$

$$\text{subject to } Ax = b$$

(33)

• This is a convex and differentiable problem that can be solved with classical optimization approaches

• Through duality theory it can be proved [4] that a solution $x^*(t)$ of (33) satisfies:

$$f_0(x^*(t)) - f^* \leq \frac{p}{t},$$

(34)

which means that $x^*(t)$ is at least $p/t$-suboptimal and $x^*(t)$ converges to an optimal point as $t$ increases

• It has been noticed that it is hard to solve directly problem (33) for big values of $t$. Problems are ill-conditioned and solvers do not work well
Interior point method: main idea

- By using $\tilde{I}(-\cdot)$ we get the following approximation of the original problem:

  $$\min_{x \in \mathbb{R}^m} \quad tf_0(x) - \sum_{i=1}^{p} \log(-f_i(x))$$

  subject to $Ax = b$ \hspace{1cm} (33)

- This is a convex and differentiable problem that can be solved with classical optimization approaches

- Through duality theory it can be proved [4] that a solution $x^*(t)$ of (33) satisfies:

  $$f_0(x^*(t)) - f^* \leq \frac{p}{t},$$

  which means that $x^*(t)$ is at least $p/t$-suboptimal and $x^*(t)$ converges to an optimal point as $t$ increases

- It has been noticed that it is hard to solve directly problem (33) for big values of $t$. Problems are ill-conditioned and solvers do not work well

- **Key idea:** compute $x^*(t)$ for a sequence of increasing values of $t$ until $t \geq p/\epsilon$, where $\epsilon$ is the desired accuracy. The point obtained as a solution for a given $t$ will be used as a starting point for solving the following problem with a greater value of $t$. 
Alternative formulation

- Remember that we are interested in solving the following problem:

\[
\min_{x \in \mathbb{R}^m} \|Ax - y\|_2^2 + \tau \|x\|_1
\]
Alternative formulation

- Remember that we are interested in solving the following problem:

$$\min_{x \in \mathbb{R}^m} \| Ax - y \|_2^2 + \tau \| x \|_1$$

- It is easy to see that it can be reformulated as:

$$\min_{x, u \in \mathbb{R}^m} \| Ax - y \|_2^2 + \tau \sum_{i=1}^{m} u_i \quad \text{subject to} \quad -u_i \leq x_i \leq u_i, \quad i = 1, \ldots, m$$

(35)
Alternative formulation

• Remember that we are interested in solving the following problem:

\[
\min_{x \in \mathbb{R}^m} \|A x - y\|^2_2 + \tau \|x\|_1
\]

• It is easy to see that it can be reformulated as:

\[
\min_{x, u \in \mathbb{R}^m} \|A x - y\|^2_2 + \tau \sum_{i=1}^{m} u_i
\]

subject to \(- u_i \leq x_i \leq u_i, \quad i = 1, \ldots, m\) (35)

• The interior point approach can be applied to formulation (35): TNIPM solves a sequence of problem for increasing \(t\) of the form

\[
\min_{x, u \in \text{dom}\psi} \phi_t(x, u) = t\|A x - y\|^2_2 + t\lambda \sum_{i=1}^{m} u_i + \psi(x, u), \quad (36)
\]

with \(\psi(x, u) : = - \sum_{i=1}^{m} \log(u_i + x_i) - \sum_{i=1}^{m} \log(u_i - x_i)\)
TNIPM: step 1

- TNIPM repeats iteratively the following steps until some termination criterion is satisfied
- In the following we omit the iteration index $k$ for simplicity of notation

**Step 1**

- Compute a search direction $(\Delta x, \Delta u)$ according to the Newton’s method by solving the following system:

$$H \begin{bmatrix} \Delta x \\ \Delta u \end{bmatrix} = -g,$$  \hspace{1cm} (37)

where $H \in \mathbb{R}^{2m \times 2m}$ and $g \in \mathbb{R}^{2m}$ are respectively the Hessian and the gradient of $\phi_t(x, u)$

- (37) is solved in an approximate way by applying the Preconditioned Conjugate Gradient (PCG) algorithm [5]
TNIPM: step 1

- PCG is an algorithm that solves system (37) by approximating the Hessian $H$
TNIPM: step 1

- PCG is an algorithm that solves system (37) by approximating the Hessian $\mathbf{H}$
- TNIPM uses the following approximation

$$P = \begin{bmatrix} 2t \text{diag}(\mathbf{A}^T \mathbf{A}) & 0 \\ 0 & 0 \end{bmatrix} + \nabla^2 \psi(x, u) \quad (38)$$

- The diagonal entries of $\mathbf{A}^T \mathbf{A}$ must be computed only once
TNIPM: step 1

- PCG is an algorithm that solves system (37) by approximating the Hessian $H$
- TNIPM uses the following approximation

$$P = \begin{bmatrix} 2t \text{diag}(A^T A) & 0 \\ 0 & 0 \end{bmatrix} + \nabla^2 \Psi(x, u)$$  \hspace{1cm} (38)

- The diagonal entries of $A^T A$ must be computed only once
- **Stopping rule**: PCG stops and returns $(\Delta x, \Delta u)$ when a maximum number of iterations is exceeded or when the residual is less than:

$$\epsilon_{pcg} = \min \{0.1, 0.01 \eta/\|g\|_2\}. $$ \hspace{1cm} (39)

$\eta$ is the duality gap at the current iterate, which can be easily computed in closed form. Since $\eta$ decreases as TNIPM goes ahead, rule (39) becomes more and more restrictive.
TNIPM: step 1

- PCG is an algorithm that solves system (37) by approximating the Hessian $H$.
- TNIPM uses the following approximation:

$$P = \begin{bmatrix} 2t \text{diag}(A^T A) & 0 \\ 0 & 0 \end{bmatrix} + \nabla^2 \psi(x, u) \quad (38)$$

- The diagonal entries of $A^T A$ must be computed only once.
- **Stopping rule:** PCG stops and returns $(\Delta x, \Delta u)$ when a maximum number of iterations is exceeded or when the residual is less than:

$$\epsilon_{pcg} = \min \{ 0.1, 0.01 \eta / \| g \|_2 \} \quad (39)$$

$\eta$ is the duality gap at the current iterate, which can be easily computed in closed form. Since $\eta$ decreases as TNIPM goes ahead, rule (39) becomes more and more restrictive.

- **Initial point:** PCG is initialized with the search direction obtained at the previous run.
Step 2

- Compute the step size \( s \) by backtracking line search
TNIPM: steps 2-5

Step 2
- Compute the step size $s$ by backtracking line search

Step 3
- Update the iterate by $(x, u) = (x, u) + s(\Delta x, \Delta u)$
TNIPM: steps 2-5

**Step 2**
- Compute the step size $s$ by backtracking line search

**Step 3**
- Update the iterate by $(x, u) = (x, u) + s(\Delta x, \Delta u)$

**Step 4**
- Evaluate the duality gap $\eta$
Step 2
- Compute the step size $s$ by backtracking line search

Step 3
- Update the iterate by $(x, u) = (x, u) + s(\Delta x, \Delta u)$

Step 4
- Evaluate the duality gap $\eta$

Step 5
- Quit if $\eta/G(\nu)$ is less than a desired tolerance, where $G(\nu)$ is the dual function $G(\nu) = -(1/4)\nu^T\nu - \nu^Ty$ and $\nu$ is a dual feasible point that can be computed in closed form from the current iterate $(x, u)$
Step 6

- Update $t$ according to the following heuristic:

$$t = \begin{cases} 
\max\{2 \min\{2n/\eta, t\}, t\} & \text{if } s \geq 0.5 \\
\max\{t\} & \text{if } s < 0.5 
\end{cases}$$

(40)
TNIPM: step 6

Step 6

- Update $t$ according to the following heuristic:

$$t = \begin{cases} \max\{2 \min\{2n/\eta, t\}, t\} & \text{if } s \geq 0.5 \\ t & \text{if } s < 0.5 \end{cases}$$ \hspace{1cm} (40)

- Note that in the classical interior point method $t$ is updated only when $\phi_t(x, u)$ is minimized with a sufficient accuracy

- TNIPM has proved to work well by updating $t$ according to heuristic (40) that allows $t$ to change also at each iteration
Experimental results

- A randomly generated filled with independent samples from a standard Gaussian distribution; \( y \) contains in each run a different amount of randomly located \( \pm 1 \) spikes; \( \tau = 0.1\|A^T y\|_\infty \) [3]
- We first run TNIPM and then each of the other algorithms until each reaches the same value of the objective function reached by TNIPM
- Results averaged over 10 independent runs

Figure: Comparison in terms of iterations versus number of non-zeros in \( y \)

Figure: Comparison in terms of CPU time versus number of non-zeros in \( y \)
Experimental results

- $A \in \mathbb{R}^{0.1 m \times m}$ randomly generated filled with independent samples from a standard Gaussian distribution; $y$ contains in each run $m/4$ randomly located $\pm 1$ spikes; $\tau = 0.1 \|A^T y\|_{+\infty}$
- We try to assess the *scalability* of the algorithms with respect to the size of the problem. We assume the computational cost to be $\mathcal{O}(m^\alpha)$ and we try to estimate $\alpha$ from the simulations
- Results averaged over 5 independent runs

### Empirical asymptotic exponents $\mathcal{O}(m^\alpha)$

- GPSR ($\alpha = 0.649$)
- TNIPM ($\alpha = 0.989$)
- SPG ($\alpha = 0.834$)

**Figure:** Comparison in terms of CPU time versus problem size
Experimental results

- A randomly generated filled with independent samples from a standard Gaussian distribution; $y$ contains in each run 160 randomly located $\pm 1$ spikes
- $\beta = \frac{\|A^T y\|_\infty}{\tau}$
- Results averaged over 10 independent runs

**Figure:** Comparison in terms of CPU time versus $\beta$
Experimental results

- We compare the performances the algorithms in the reconstruction of a signal $f$ from the noisy observation $y = Af + n$ where $A \in \mathbb{R}^{1024 \times 4096}$ is filled with independent samples from a standard Gaussian distribution while $n$ is Gaussian noise with variance $\sigma^2 = 10^{-4}$.
- $f$ contains 160 randomly located $\pm 1$ spikes; $\tau = 0.1\|A^Ty\|_{+\infty}$

![Figure: Comparison in terms of objective function value versus iterations](image1.png)

![Figure: Comparison in terms of objective function value versus CPU time](image2.png)
Experimental results

- $\text{MSE} = \frac{1}{m} \| \hat{x} - x \|_2^2$, where $\hat{x}$ is an estimate of a vector $x \in \mathbb{R}^m$
- Minimum norm solution: $\hat{x} = A^T(AA^T)^{-1}y$

![Figures showing reconstruction of the signal via different approaches: Original, GPSR, TNIPM, SPG, Minimum norm solution. Each figure includes a plot with axes labeled and data points illustrating the reconstruction quality with associated MSE values.]
References


