A Derivative-free Adaptation of the Penalty Decomposition Method for Sparse Optimization

EUROPT 2021, Toulouse, 9th July 2021

Matteo Lapucci, Tommaso Levato, Marco Sciandrone

DINFO, Univerisity of Florence

Sparsity Constrained Optimization Problem

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \ & f(x) \\ \text{s.t.} \ & \|x\|_0 \le s \end{aligned} \tag{SCOP}$$

- $f : \mathbb{R}^n \to \mathbb{R}$ continuously differentiable;
- ℓ_0 pseudo-norm $\|x\|_0 = |\{i \mid x_i \neq 0\}|$;
- $s < n$.

Basic Definitions and Properties

- *support set:* $I_1(x) = \{i \mid x_i \neq 0\}$.

Basic Definitions and Properties

- *support set:* $I_1(x) = \{i \mid x_i \neq 0\}$.
- *super support set:* $J \subset \{1, \ldots, n\}$ s.t.

Basic Definitions and Properties

- *support set:* $I_1(x) = \{i \mid x_i \neq 0\}$.
- *super support set:* $J \subset \{1, \ldots, n\}$ s.t.
  - $|J| = s$,

Basic Definitions and Properties

- *support set:* $I_1(x) = \{i \mid x_i \neq 0\}$.
- *super support set:* $J \subset \{1, \ldots, n\}$ s.t.
  - $|J| = s$,
  - $I_1(x) \subseteq J$;

Basic Definitions and Properties

- *support set:* $I_1(x) = \{i \mid x_i \neq 0\}$.
- *super support set:* $J \subset \{1, \ldots, n\}$ s.t.
  - $|J| = s$,
  - $I_1(x) \subseteq J$;
- $x^\star$ satisfies *Lu-Zhang (LZ) conditions* if there exists a super support set J for $x^\star$ such that $\nabla_J f(x^\star) = 0$;

Basic Definitions and Properties

- *support set:* $I_1(x) = \{i \mid x_i \neq 0\}$.
- *super support set:* $J \subset \{1, \ldots, n\}$ s.t.
  - $|J| = s$,
  - $I_1(x) \subseteq J$;
- $x^\star$ satisfies *Lu-Zhang (LZ) conditions* if there exists a super support set J for $x^\star$ such that $\nabla_J f(x^\star) = 0$;
  - $x^\star$ local optimizer $\implies$ $x^\star$ satisfies LZ conditions;

Penalty Decomposition Approach (Lu and Zhang, 2013)

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{s.t.} \quad \|x\|_0 \leq s. \qquad \Leftrightarrow \qquad \begin{aligned} \min_{x,y \in \mathbb{R}^n} \; & f(x) \\ \text{s.t.} \; & \|y\|_0 \leq s, \\ & x - y = 0. \end{aligned}$$

Penalty Decomposition Approach (Lu and Zhang, 2013)

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \; & f(x) \\ \text{s.t.} \; & \|x\|_0 \le s. \end{aligned} \quad \Leftrightarrow \quad \begin{aligned} \min_{x,y \in \mathbb{R}^n} \; & f(x) \\ \text{s.t.} \; & \|y\|_0 \le s, \\ & x - y = 0. \end{aligned}$$

- Penalty Function:

$$q_\tau(x, y) = f(x) + \tau \|x - y\|^2$$

Penalty Decomposition Approach (Lu and Zhang, 2013)

Algorithm 1: `PenaltyDecomposition`

Input: $\tau_0 > 0$, $\theta > 1$, $x^0 = y^0 \in \mathbb{R}^n$ s.t. $\|x^0\|_0 \leq s$, $\{\varepsilon_k\}$ s.t. $\varepsilon_k \to 0$.

for $k = 0, 1, \ldots$ **do**

  $\ell = 0$, $u^0, v^0 = x^k, y^k$

  while $\|\nabla_x q_{\tau_k}(u^\ell, v^\ell)\| > \varepsilon_k$ **do**

    $u^{\ell+1} \in \arg\min_u q_{\tau_k}(u, v^\ell)$; `// x update step`

    $v^{\ell+1} \in \arg\min_{\|y\|_0 \leq s} q_{\tau_k}(u^{\ell+1}, v)$; `// y update step`

    $\ell = \ell + 1$

  $\tau_{k+1} = \theta \tau_k$; `// penalty parameter update`

  $x^{k+1}, y^{k+1} = u^\ell, v^\ell$

Output: The sequence $\{x^k\}$.

PD: Convergence Properties

Theorem

Let $\{x^k, y^k\}$ be the sequence generated by the PD Algorithm. Then $\{x^k, y^k\}$ admits cluster points and every cluster point $(\bar{x}, \bar{y})$ is such that

- $\bar{x} = \bar{y}$,
- $\|\bar{x}\|_0 \leq s$,
- $\bar{x}$ *satisfies the Lu-Zhang conditions for problem (SCOP).*

Nonconvex Case

$$u^{\ell+1} \in \arg\min_{u} q_{\tau_k}(u, v^{\ell})$$

Nonconvex Case

$$u^{\ell+1} \in \arg\min_{u} q_{\tau_k}(u, v^{\ell})$$

$$\downarrow$$

$$u^{\ell+1} = u^{\ell} - \alpha_{\ell} \nabla_x q_{\tau_k}(u^{\ell}, v^{\ell})$$

α_{ℓ} computed by an Armijo line search

Nonconvex Case

$$u^{\ell+1} \in \arg\min_{u} q_{\tau_k}(u, v^{\ell})$$

$$\downarrow$$

$$u^{\ell+1} = u^{\ell} - \alpha_{\ell} \nabla_x q_{\tau_k}(u^{\ell}, v^{\ell})$$

α_{ℓ} computed by an Armijo line search

Proposition

The $\mathit{inexact}$ PD Algorithm enjoys the same convergence properties as the original approach.

The Derivative-Free Setting

- (SCOP);
- ∇f not available;
- no tailored approach from the literature...

The Derivative-Free Setting

- (SCOP);
- ∇f not available;
- no tailored approach from the literature...

Idea: **Adapting the Penalty Decomposition scheme.**

A Derivative-Free Line Search

- Compute step length along direction d;

A Derivative-Free Line Search

- Compute step length along direction d;
- Initial step size α_0;

A Derivative-Free Line Search

- Compute step length along direction d;
- Initial step size α_0;
- If $f(x + \alpha_0 d) \le f(x) - \gamma \alpha_0^2 \|d\|^2$ (sufficient decrease at α_0)

A Derivative-Free Line Search

- Compute step length along direction d;
- Initial step size α_0;
- If $f(x + \alpha_0 d) \leq f(x) - \gamma \alpha_0^2 \|d\|^2$ (sufficient decrease at α_0)
  set:

$$\alpha^* = \min_{t=1,2\dots} \{\sigma^{t-1}\alpha_0 \mid f(x + \sigma^t \alpha_0 d) \leq f(x) - \gamma(\sigma^t \alpha_0)^2 \|d\|^2\}$$

(extrapolation step, $\sigma > 1$)

A Derivative-Free Line Search

- Compute step length along direction d;
- Initial step size α_0;
- If $f(x + \alpha_0 d) \leq f(x) - \gamma \alpha_0^2 \|d\|^2$ (sufficient decrease at α_0)
  set:

$$\alpha^* = \min_{t=1,2\ldots} \{\sigma^{t-1}\alpha_0 \mid f(x + \sigma^t \alpha_0 d) \leq f(x) - \gamma(\sigma^t \alpha_0)^2 \|d\|^2\}$$

(extrapolation step, $\sigma > 1$)

- Otherwise set $\alpha^* = 0$ (failure)

Local Optimization by Coordinate Search

- $\mathcal{D} = \{e_1, \ldots, e_n, -e_1, \ldots, -e_n\}$;

Local Optimization by Coordinate Search

- $\mathcal{D} = \{e_1, \ldots, e_n, -e_1, \ldots, -e_n\}$;
- $\tilde{\alpha}_i \sim d_i \in \mathcal{D}$;

Local Optimization by Coordinate Search

- $\mathcal{D} = \{e_1, \ldots, e_n, -e_1, \ldots, -e_n\}$;
- $\tilde{\alpha}_i \sim d_i \in \mathcal{D}$;

Algorithm 4: `Coordinate Search`

Input: $\tilde{\alpha} \in \mathbb{R}^{2n}$, $\varepsilon_k > 0$, (u^ℓ, v^ℓ), $\gamma, \delta < 1$, $\sigma > 1$.
for $i = 1, \ldots, 2n$ **do**
    $\alpha_i^\ell = \texttt{LineSearch}(q_{\tau_k}(u, v^\ell), d_i, \tilde{\alpha}_i^\ell, u^\ell, \gamma, \sigma)$; `// DF line search`
    if $\alpha_i^\ell = 0$ **then**
        $\tilde{\alpha}_i = \delta \tilde{\alpha}_i$; `// failure: decrease base steplength for` d_i
    else
        $\tilde{\alpha}_i = \alpha_i$; `// success: update base steplength for` d_i
    if $\alpha_i^\ell > \varepsilon_k$ **then**
        $u^\ell = u^\ell + \alpha_i^\ell d_i$; `// perform step if stepsize not too small`

Output: u^ℓ.

Derivative-Free Penalty Decomposition

- x-**update step:**

$$u^{\ell+1} \in \arg\min_{u} q_{\tau_k}(u, v^{\ell})$$

Derivative-Free Penalty Decomposition

- x-**update step:**

$$u^{\ell+1} \in \arg\min_{u} q_{\tau_k}(u, v^{\ell})$$

$$\downarrow$$

$$u^{\ell+1} = \texttt{CoordinateSearch}(q_{\tau_k}(\cdot, v^{\ell}), \mathcal{D}, \tilde{\alpha}^{\ell})$$

Derivative-Free Penalty Decomposition

- ***x*-update step:**

$$u^{\ell+1} \in \arg\min_{u} q_{\tau_k}(u, v^{\ell})$$

$$\downarrow$$

$$u^{\ell+1} = \texttt{CoordinateSearch}(q_{\tau_k}(\cdot, v^{\ell}), \mathcal{D}, \tilde{\alpha}^{\ell})$$

- **Inner loop stopping criterion:**

$$\|\nabla_x q_{\tau_k}(u^{\ell}, v^{\ell})\| \leq \varepsilon_k$$

Derivative-Free Penalty Decomposition

- **x-update step:**

$$u^{\ell+1} \in \arg\min_{u} q_{\tau_k}(u, v^{\ell})$$

$$\downarrow$$

$$u^{\ell+1} = \texttt{CoordinateSearch}(q_{\tau_k}(\cdot, v^{\ell}), \mathcal{D}, \tilde{\alpha}^{\ell})$$

- **Inner loop stopping criterion:**

$$\|\nabla_x q_{\tau_k}(u^{\ell}, v^{\ell})\| \leq \varepsilon_k$$

$$\downarrow$$

$$\max_{i=1,\dots,2n} \{\tilde{\alpha}_i^{\ell}\} \leq \varepsilon_k$$

Derivative-Free Penalty Decomposition

Algorithm 5: `DerivativeFreeInexactPenaltyDecomposition`

```
for k = 0, 1, ... do
    ℓ = 0, α̃^0 = e ∈ ℝ^{2n}, u^0, v^0 = x^k, y^k
    while max_{i=1,...,2n} {α̃_i^ℓ} > ε_k do
        for i = 1, ..., 2n do
            α_i^ℓ = LineSearch(q_{τ_k}(u, v^ℓ), d_i, α̃_i^ℓ, u^ℓ, γ, σ)
            if α_i^ℓ = 0 then
                α̃_i^{ℓ+1} = δ α̃_i^ℓ
            else
                α̃_i^{ℓ+1} = α_i^ℓ
            if α_i^ℓ > ε_k then
                u^ℓ = u^ℓ + α_i^ℓ d_i^ℓ
        u^{ℓ+1} = u^ℓ
        v^{ℓ+1} ∈ arg min_{||v||_0 ≤ s} q_{τ_k}(u^{ℓ+1}, v)
        ℓ = ℓ + 1
    τ_{k+1} = θ τ_k
    x^{k+1}, y^{k+1} = u^ℓ, v^ℓ
```

Output: The sequence $\{x^k\}$.

Derivative-Free PD: Convergence

Theorem

Let $\{x^k, y^k\}$ be the sequence generated by the `derivative-free Penalty Decomposition` *Algorithm with $\varepsilon_k \to 0$. Then $\{x^k, y^k\}$ admits cluster points and every cluster point $(\bar{x}, \bar{y})$ is such that*

- $\bar{x} = \bar{y}$,
- $\|\bar{x}\|_0 \leq s$,
- $\bar{x}$ *satisfies the Lu-Zhang conditions for problem (SCOP).*

Derivative-Free PD: Convergence

Theorem

Let $\{x^k, y^k\}$ be the sequence generated by the `derivative-free Penalty Decomposition` *Algorithm with $\varepsilon_k \to 0$. Then $\{x^k, y^k\}$ admits cluster points and every cluster point $(\bar{x}, \bar{y})$ is such that*

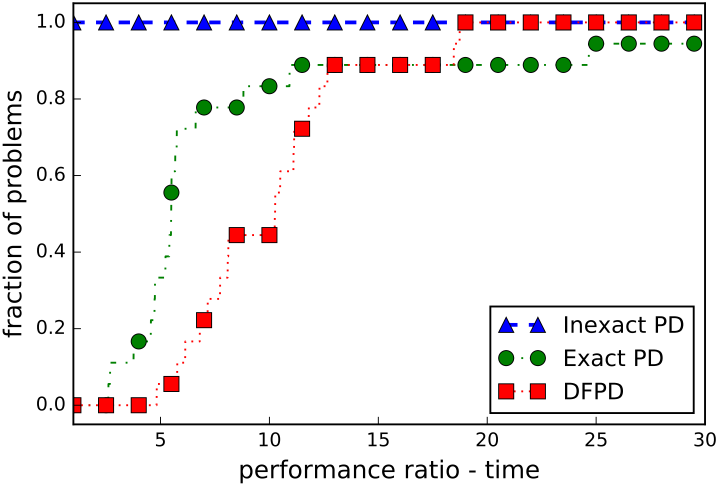
- $\bar{x} = \bar{y}$,
- $\|\bar{x}\|_0 \leq s$,
- $\bar{x}$ *satisfies the Lu-Zhang conditions for problem (SCOP).*

Same exact result as the original PD algorithm!

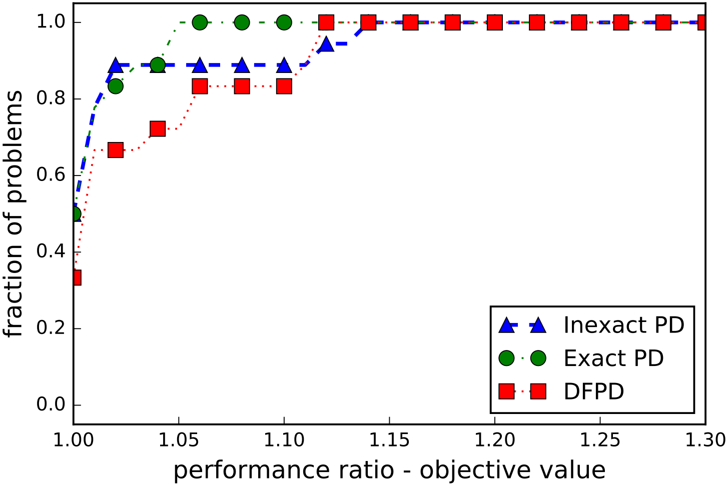
Preliminary Experiments: Setup

- **Sparse Logistic Regression Problem**:
  - Dataset: $\{(z_i, l_i) \mid z_i \in \mathbb{R}^n, l_i \in \mathbb{R}, i = 1, \ldots, N\}$
  - Loss (objective):
    $$f(w) = \sum_{i=1}^{N} \log\left(1 + \exp\left(-l_i(w^T z^i)\right)\right)$$
  - Feature subset selection:
    $$\|w\|_0 \leq s$$
  - 6 datasets from UCI ML Repository:
    - `heart` ($n = 25$)
    - `breast` ($n = 33$)
    - `biodeg` ($n = 41$)
    - `spectf` ($n = 44$)
    - `spam` ($n = 57$)
    - `a2a` ($n = 123$)
  - $s = \frac{1}{4}n, \ \frac{1}{2}n, \ \frac{3}{4}n.$

Preliminary Experiments: Results - Time



Preliminary Experiments: Results - Objective Value



Conclusions

- Penalty Decomposition can be adapted to be used in DF setting;
- Theoretically equivalent to its derivative-based counterpart;
- Preliminary experiments show practical reliability of the approach.

A Derivative-free Adaptation of the Penalty Decomposition Method for Sparse Optimization

EUROPT 2021, Toulouse, 9th July 2021

Matteo Lapucci, Tommaso Levato, Marco Sciandrone

DINFO, Univerisity of Florence