

A Block Decomposition Algorithm for Sparse Optimization

Ganzhao Yuan¹, **Li Shen**², **Wei-Shi Zheng**³

¹ Peng Cheng Laboratory, China

² Tencent AI Lab, China ³ Sun Yat-sen University, China

July 15, 2020

Outline of This Talk

- Introduction
- The Proposed Algorithm
- Comparisons and Contributions
- Optimality Analysis
- Convergence Analysis
- Experimental Validation

Introduction

Optimization Problem:

$$\min_{\mathbf{x}} f(\mathbf{x}), \text{ s.t. } \|\mathbf{x}\|_0 \leq s \quad \text{or} \quad \min_{\mathbf{x}} f(\mathbf{x}) + \lambda \|\mathbf{x}\|_0,$$

- $f(\mathbf{x})$: smooth, convex, its gradient is L -Lipschitz continuous
- They can be rewritten as the following problem

$$\min_{\mathbf{x}} F(\mathbf{x}) \triangleq f(\mathbf{x}) + h(\mathbf{x}), \text{ with } h(\mathbf{x}) \triangleq h_{\text{cons}} \text{ or } h_{\text{regu}}.$$

$$h_{\text{cons}}(\mathbf{x}) \triangleq I_{\Psi}(\mathbf{x}), \quad \Psi \triangleq \{\mathbf{x} \mid \|\mathbf{x}\|_0 \leq s\}, \quad I_{\Psi}(\mathbf{x}) = \begin{cases} 0, & \mathbf{x} \in \Psi \\ \infty, & \mathbf{x} \notin \Psi \end{cases}$$

- These two problems are equivalent in the sense that $\lambda \propto \frac{1}{s}$.
- Applications: compressive sensing, sparse coding, subspace clustering

Application: Compressed Sensing

Signal recovery problem under noisy observations:

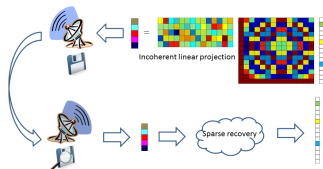
$$\min_{\mathbf{x} \in \mathbb{R}^n} \frac{1}{2} \|\Phi \mathbf{x} - \mathbf{y}\|_2^2, \text{ s.t. } \|\mathbf{x}\|_0 \leq s$$

Remarks.

- 1 Signal recovery problem:

$$\min_{\mathbf{x} \in \mathbb{R}^n} \|\mathbf{x}\|_0, \text{ s.t. } \Phi \mathbf{x} = \mathbf{y}$$

- 2 Many applications: magnetic resonance imaging, single-pixel camera



$$\begin{array}{l} M \times 1 \\ \text{measurements} \end{array} \mathbf{y} = \begin{array}{c} \Phi \\ M \times N \end{array} \begin{array}{l} N \times 1 \\ \text{sparse} \\ \text{signal} \\ \mathbf{x} \\ K \\ \text{nonzero} \\ \text{entries} \end{array}$$

$K < M \ll N$

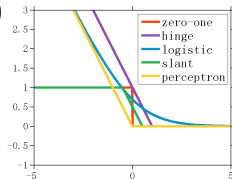
Application: Sparse Logistic Regression

Sparse logistic regression model:

$$\min_{\mathbf{x} \in \mathbb{R}^n} \sum_{i=1}^m \log(1 + \exp(-\mathbf{y}_i \langle \mathbf{s}_i, \mathbf{x} \rangle)), \quad \text{s.t. } \|\mathbf{x}\|_0 \leq s$$

Remarks.

- 1 Tackle both binary and multiclass classification problems
- 2 Other alternative loss functions $f(\mathbf{x}) = \sum_{i=1}^m l(\mathbf{x}, s_i, y_i)$
 - Zero-one loss: $I(f(s_i) = y_i)$ (I is the indicator function)
 - Slant loss: $\min(1, \min(0, 0.5 - f(s_i)y_i))$
 - Hinge loss: $\max(0, 1 - f(s_i)y_i)$
 - Perceptron loss: $\max(0, -f(s_i)y_i)$



The Proposed Algorithm

The Proposed Algorithm

Input: the size of the working set k , an initial feasible solution \mathbf{x}^0 .
Set $t = 0$.

while *not converge* **do**

S1 Employ some strategy to find a working set B of size k .

Denote $\bar{B} \triangleq \{1, \dots, n\} \setminus B$

S2 Solve the following subproblem globally using combinatorial search:

$$\mathbf{x}^{t+1} \leftarrow \arg \min_{\mathbf{z}} F(\mathbf{z}) + \frac{\theta}{2} \|\mathbf{z} - \mathbf{x}^t\|^2, \text{ s.t. } \mathbf{z}_{\bar{B}} = \mathbf{x}_{\bar{B}}^t$$

S3 Increment t by 1

end

Algorithm 1: A Block Decomposition Algorithm for Sparse Optimization

S1 Finding the working set

- **Random strategy.** Select one combination (which contains k coordinates) from the whole working set of size C_n^k uniformly.
- **Greedy strategy.** Pick top- k coordinates (as in \mathbf{c} and \mathbf{d} below) that lead to the greatest descent when one variable is changed (the rest variables are fixed) based on the current solution \mathbf{x}^t .

$$\forall i = 1, \dots, |Z|, \quad Z \triangleq \{i : \mathbf{x}_i^t = 0\}, \quad \mathbf{c}_i = \min_{\alpha} F(\mathbf{x}^t + \alpha \mathbf{e}_i) - F(\mathbf{x}^t).$$

$$\forall j = 1, \dots, |\bar{Z}|, \quad \bar{Z} \triangleq \{j : \mathbf{x}_j^t \neq 0\}, \quad \mathbf{d}_j = F(\mathbf{x}^t + \alpha \mathbf{e}_j) - F(\mathbf{x}^t), \quad \alpha = -\mathbf{x}_j^t.$$

S2 The main procedure:

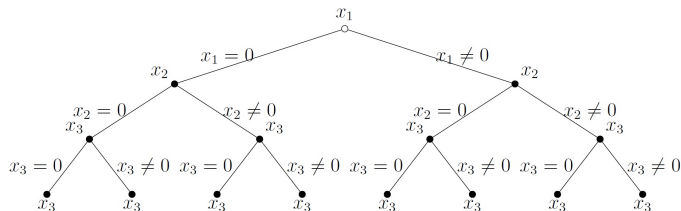
$$\mathbf{x}^{t+1} \leftarrow \arg \min_{\mathbf{z}} F(\mathbf{z}) + \frac{\theta}{2} \|\mathbf{z} - \mathbf{x}^t\|^2, \text{ s.t. } \mathbf{z}_{\bar{B}} = \mathbf{x}_{\bar{B}}^t$$

1 Two New Strategies

- a proximal point strategy: global convergence
- an exhaustive search function: minimize the original function

2 Solving the Subproblem Globally

- k unknown decision variables, a simple exhaustive search
- An example: $F(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \mathbf{Q} \mathbf{x} + \langle \mathbf{x}, \mathbf{p} \rangle + \lambda \|\mathbf{x}\|_0$
- reduce to solving $\sum_{i=0}^k C_k^i$ linear systems



Comparisons and Contributions

1 Relaxed approximation method

- convex: $f(\mathbf{x}) + \lambda\|\mathbf{x}\|_1$, $f(\mathbf{x}) + \lambda\|\mathbf{x}\|_{\text{tok}-k}$
- nonconvex: $f(\mathbf{x}) + \lambda\|\mathbf{x}\|_p$, reweighted ℓ_1 norm

⇒ our method directly controls the sparsity of the solution

2 Greedy pursuit method

- the solution MUST be initialized to zero
- $S = \emptyset$, $S = S \cup i_1$, $\min_{\mathbf{x}_S} f(\mathbf{x})$, $S = S \cup i_2$, $\min_{\mathbf{x}_S} f(\mathbf{x})$, ...

⇒ our method is a greedy coordinate descent algorithm without forcing the initial solution to zero

- ③ Combinatorial search method: global optimization methods
 - cutting plane methods
 - branch-and-cut methods

⇒ our method leverages the effectiveness of combinatorial search method methods

- ④ Proximal gradient method

- $\mathbf{x}^{k+1} = \text{Prox}_{\gamma h}(\mathbf{x}^k - \gamma \nabla f(\mathbf{x}^k))$
- with $\text{Prox}_{\bar{h}}(\mathbf{a}) = \arg \min_{\mathbf{x}} \frac{1}{2} \|\mathbf{x} - \mathbf{a}\|_2^2 + \bar{h}(\mathbf{x})$

⇒ our method significantly outperforms proximal gradient method

Contributions of this paper are three-fold:

- 1 **Algorithmically**: a new decomposition method, use combinatorial search and coordinate descent
- 2 **Theoretically**:
 - optimality analysis: optimality hierarchy, finds stronger stationary points (e.g. it finds stronger stationary points than Beck and Eldar's method ¹)
 - convergence analysis: convergence, convergence rate
- 3 **Empirically**: state-of-the-art performance (e.g. generally outperforms OMP method)

¹Sparsity Constrained Nonlinear Optimization: Optimality Conditions and Algorithms, SIAM Journal on Optimization, 2014.

Optimality Analysis

Basic Stationary Point

A solution $\check{\mathbf{x}}$ is called a basic stationary point if the following holds.

$$h \triangleq h_{\text{regu}} : \check{\mathbf{x}} = \arg \min_{\mathbf{y}} f(\mathbf{y}), \text{ s.t. } \mathbf{y}_Z = \mathbf{0};$$

$$h \triangleq h_{\text{cons}} : \check{\mathbf{x}} = \arg \min_{\mathbf{y}} f(\mathbf{y}), \text{ s.t. } |S| \leq k, \mathbf{y}_Z = \mathbf{0}.$$

Here, $S \triangleq \{i | \check{\mathbf{x}}_i \neq 0\}$, $Z \triangleq \{j | \check{\mathbf{x}}_j = 0\}$.

L-Stationary Point

A solution $\check{\mathbf{x}}$ is an L -stationary point if it holds that:

$$\check{\mathbf{x}} = \arg \min_{\mathbf{x}} g(\mathbf{x}, \check{\mathbf{x}}) + h(\mathbf{x}) \text{ with}$$

$$g(\mathbf{x}, \mathbf{z}) \triangleq f(\mathbf{z}) + \langle \nabla f(\mathbf{z}), \mathbf{x} - \mathbf{z} \rangle + \frac{L}{2} \|\mathbf{x} - \mathbf{z}\|_2^2.$$

Block- k Stationary Point

A solution $\bar{\mathbf{x}}$ is a block- k stationary point if it holds that:

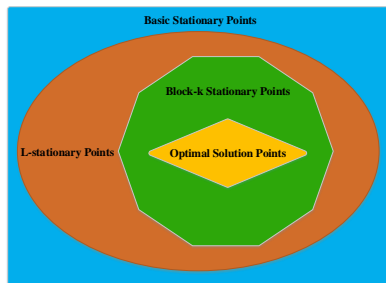
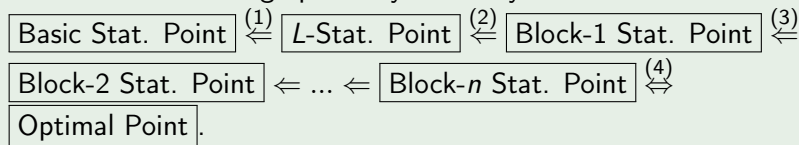
$$\forall |B| = k, \bar{\mathbf{x}} \in \arg \min_{\mathbf{z} \in \mathbb{R}^n} \mathcal{P}(\mathbf{z}; \bar{\mathbf{x}}, B) \triangleq \{F(\mathbf{z}), \text{ s.t. } \mathbf{z}_{\bar{B}} = \bar{\mathbf{x}}_{\bar{B}}\}$$

$$\bar{B} \triangleq \{1, \dots, n\} \setminus B.$$

Optimality Hierarchy

Relations between the three types of stationary point.

We have the following optimality hierarchy:



A Running Example

Optimization Problems:

$$\min_{\mathbf{x} \in \mathbb{R}^n} \frac{1}{2} \mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{x}^T \mathbf{p}, \quad s.t. \quad \|\mathbf{x}\|_0 \leq 4$$

$$\min_{\mathbf{x} \in \mathbb{R}^n} \frac{1}{2} \mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{x}^T \mathbf{p} + 0.01 \|\mathbf{x}\|_0$$

$$n = 6, \quad \mathbf{Q} = \mathbf{c} \mathbf{c}^T + \mathbf{I}, \quad \mathbf{p} = \mathbf{1}, \quad \mathbf{c} = [1 \ 2 \ 3 \ 4 \ 5 \ 6]^T.$$

Number of points satisfying optimality conditions.

	Basic-Stat.	L-Stat.	Block-1 Stat.	Block-2 Stat.	Block-3 Stat.	Block-4 Stat.	Block-5 Stat.	Block-6 Stat.
$h \triangleq h_{\text{cons.}}$	57	14	–	2	1	1	1	1
$h \triangleq h_{\text{regu.}}$	64	56	9	3	1	1	1	1

Convergence Analysis

Global Convergence.

Letting \mathbf{x}^t be the sequence generated by Algorithm 1, we have the following results.

(i) It holds that:

$$F(\mathbf{x}^{t+1}) - F(\mathbf{x}^t) \leq -\frac{\theta}{2} \|\mathbf{x}^{t+1} - \mathbf{x}^t\|^2, \quad \lim_{t \rightarrow \infty} \mathbb{E}[\|\mathbf{x}^{t+1} - \mathbf{x}^t\|] = 0.$$

(ii) As $t \rightarrow \infty$, \mathbf{x}^t converges to the block- k stationary point $\bar{\mathbf{x}}$ in expectation.

Optimization problem:

$$\min_{\mathbf{x}} f(\mathbf{x}), \text{ s.t. } \|\mathbf{x}\|_0 \leq k$$

Convergence Rate for Sparsity Constrained Optimization

Let $f(\cdot)$ be a σ -strongly convex function. We assume that $f(\cdot)$ is Lipschitz continuous such that $\forall t, \|\nabla f(\mathbf{x}^t)\|_2^2 \leq \kappa$ for some positive constant κ . Denote $\alpha \triangleq \frac{n\theta}{k\sigma} / (1 + \frac{n\theta}{k\sigma})$. We have the following results:

$$\mathbb{E}[F(\mathbf{x}^t) - F(\bar{\mathbf{x}})] \leq (F(\mathbf{x}^0) - F(\bar{\mathbf{x}}))\alpha^t + \frac{\kappa}{2\theta} \frac{\alpha}{1-\alpha},$$

and
$$\mathbb{E}[\frac{\sigma}{4} \|\mathbf{x}^{t+1} - \bar{\mathbf{x}}\|_2^2] \leq \frac{n2\theta}{k} (F(\mathbf{x}^0) - F(\bar{\mathbf{x}}))\alpha^t + \frac{n}{k} \frac{\kappa}{1-\alpha}.$$

Convergence Rate When $h \triangleq f_{\text{regu}}$ (a)

Optimization problem:

$$\min_{\mathbf{x}} f(\mathbf{x}) + \lambda \|\mathbf{x}\|_0$$

Convergence Rate for Sparse Regularized Optimization.

(a) It holds that $|x_i^t| \geq \delta$ for all i with $x_i^t \neq 0$. Whenever $\mathbf{x}^{t+1} \neq \mathbf{x}^t$, we have $\|\mathbf{x}^{t+1} - \mathbf{x}^t\|_2^2 \geq \frac{k\delta^2}{n}$ and the objective value is decreased at least by D . The solution changes at most \bar{J} times in expectation for finding a block- k stationary point $\bar{\mathbf{x}}$. Here δ , D , and \bar{J} are respectively defined as:

$$\delta \triangleq \min(\rho, \sqrt{2\lambda/(\theta + L)}, \min(|\mathbf{x}^0|)), \quad D \triangleq \frac{k\theta\delta^2}{2n}, \quad \bar{J} \triangleq \frac{F(\mathbf{x}^0) - F(\bar{\mathbf{x}})}{D}.$$

Convergence Rate When $h \triangleq f_{\text{regu}}$ (b)

Optimization problem:

$$\min_{\mathbf{x}} f(\mathbf{x}) + \lambda \|\mathbf{x}\|_0$$

Convergence Rate for Sparse Regularized Optimization.

(b) Assume that $f(\cdot)$ is generally convex, and it holds that $\forall t, \|\mathbf{x}^t\|_\infty \leq \rho$. If the support set of \mathbf{x}^t does not change for all $t = 0, 1, \dots, \infty$, Algorithm 1 takes at most V_1 iterations in expectation to converge to a stationary point $\bar{\mathbf{x}}$ satisfying $F(\mathbf{x}^t) - F(\bar{\mathbf{x}}) \leq \epsilon$. Moreover, Algorithm 1 takes at most $V_1 \times \bar{J}$ in expectation to converge to a stationary point $\bar{\mathbf{x}}$ satisfying $F(\mathbf{x}^t) - F(\bar{\mathbf{x}}) \leq \epsilon$. Here, V_1 is defined as:

$$V_1 = \max\left(\frac{4\nu^2}{\theta}, \sqrt{\frac{2\nu^2(F(\mathbf{x}^0) - F(\bar{\mathbf{x}}))}{\theta}}\right) / \epsilon, \text{ with } \nu \triangleq \frac{2n\rho\sqrt{k}\theta}{k}.$$

Convergence Rate When $h \triangleq f_{\text{regu}}(c)$

Optimization problem:

$$\min_{\mathbf{x}} f(\mathbf{x}) + \lambda \|\mathbf{x}\|_0$$

Convergence Rate Sparse Regularized Optimization.

(c) Assume that $f(\cdot)$ is σ -strongly convex. If the support set of \mathbf{x}^t does not change for all $t = 0, 1, \dots, \infty$, Algorithm 1 takes at most V_2 iterations in expectation to converge to a stationary point $\bar{\mathbf{x}}$ satisfying $F(\mathbf{x}^t) - F(\bar{\mathbf{x}}) \leq \epsilon$. Moreover, Algorithm 1 takes at most $V_2 \times \bar{J}$ in expectation to converge to a stationary point $\bar{\mathbf{x}}$ satisfying $F(\mathbf{x}^t) - F(\bar{\mathbf{x}}) \leq \epsilon$. Here, V_2 is defined as:

$$V_2 = \log_{\alpha}(\epsilon / (F(\mathbf{x}^0) - F(\bar{\mathbf{x}}))), \text{ with } \alpha \triangleq \frac{n\theta}{k\sigma} / (1 + \frac{n\theta}{k\sigma}).$$

Optimization problem:

$$\min_{\mathbf{x}} f(\mathbf{x}) + \lambda \|\mathbf{x}\|_0, \quad \text{s.t.} \quad \|\mathbf{x}\|_\infty \leq \rho$$

Improved Convergence Rate for Sparse Regularized Optimization.

(a) Assume that $f(\cdot)$ is generally convex, Algorithm 1 takes at most N_1 iterations in expectation to converge to a block- k stationary point $\bar{\mathbf{x}}$ satisfying $F(\mathbf{x}^t) - F(\bar{\mathbf{x}}) \leq \epsilon$, where

$$N_1 = \left(\frac{\bar{J}}{D} + \frac{1}{\epsilon} \right) \times \max\left(\frac{4\nu^2}{\theta}, \sqrt{\frac{2\nu^2(F(\mathbf{x}^0) - F(\bar{\mathbf{x}}) - D)}{\theta}} \right).$$

(b) Assume that $f(\cdot)$ is σ -strongly convex, Algorithm 1 takes at most N_2 iterations in expectation to converge to a block- k stationary point $\bar{\mathbf{x}}$ satisfying $F(\mathbf{x}^t) - F(\bar{\mathbf{x}}) \leq \epsilon$, where

$$N_2 = \bar{J} \log_\alpha\left(\frac{D}{F(\mathbf{x}^0) - F(\bar{\mathbf{x}})}\right) + \log_\alpha\left(\frac{\epsilon}{F(\mathbf{x}^0) - D - F(\bar{\mathbf{x}})}\right).$$

Experimental Validation


Sparsity Constrained Least Squares Problem

Optimization problems:

$$\min_{\mathbf{x} \in \mathbb{R}^n} \frac{1}{2} \|\mathbf{Ax} - \mathbf{b}\|_2^2, \text{ s.t. } \|\mathbf{x}\|_0 \leq s$$

Compared Methods:

- Proximal Gradient Method (PGM)
- Accelerated Proximal Gradient Method (APGM)
- Quadratic Penalty Method (QPM)
- Subspace Pursuit (SSP)
- Regularized Orthogonal Matching Pursuit (ROMP)
- Orthogonal Matching Pursuit (OMP)
- Compressive Sampling Matched Pursuit (CoSaMP)
- Convex ℓ_1 Approximation Method (CVX- ℓ_1)
- Proposed Decomposition Method (DEC-RiGj) ²

²Selecting i coordinates using Random strategy and j coordinates using Greedy strategy 

Sparsity Constrained Least Squares Problem

Conclusions:

- **DEC** is more effective than {PGM, APGM}. As k becomes larger, higher accuracy is achieved.
- **DEC-R0G2** converges quickly but it leads to worse solution quality than **DEC-R2G0**. A combined strategy is preferred.
- **DEC** generally takes less than 30 seconds to converge.

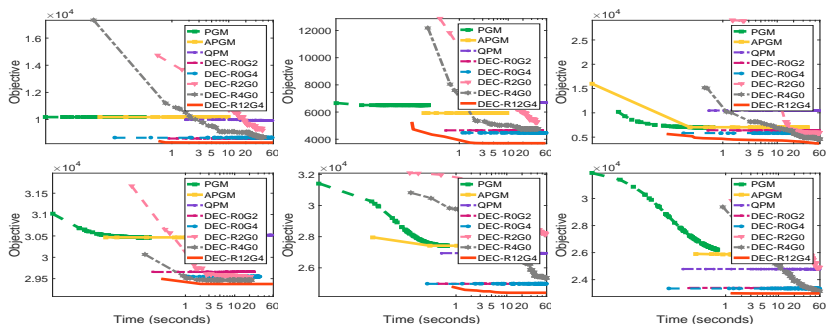


Figure: Convergence curve and computational efficiency for solving sparsity constrained least squares problems on different data sets with different s .

Sparsity Constrained Least Squares Problem

Conclusions:

- IHT methods {PGM, APGM, QPM} lead to bad performance.
- OMP and ROMP sometimes achieve bad accuracy.
- **DEC** significantly and consistently outperforms the greedy methods on many data sets.

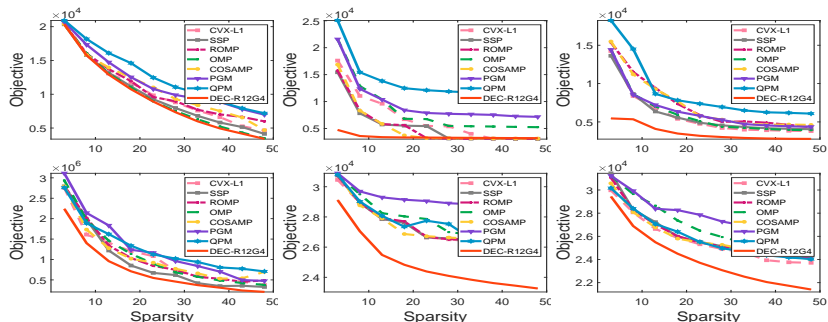


Figure: Experimental results on sparsity constrained least squares problems on different data sets with varying the sparsity of the solution.

Sparse Regularized Least Squares Problem

Optimization problems:

$$\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{Ax} - \mathbf{b}\|_2^2 + \lambda \|\mathbf{x}\|_0$$

Compared Methods:

- PGM- ℓ_0 : PGM for ℓ_0 Problem
- APGM- ℓ_0 : Accelerated PGM for ℓ_0 Problem
- PGM- ℓ_1 : PGM for ℓ_1 Problem
- PGM- ℓ_p : PGM for ℓ_p Problem ($p = \frac{1}{2}$)
- Proposed Decomposition Method (DEC-RiGj)

Sparse Regularized Least Squares Problem

Conclusions:

- PGM- ℓ_p achieves better performance than PGM- ℓ_1 .
- DEC generally outperforms the other methods in all data sets.

	PGM- ℓ_0	APGM- ℓ_0	PGM- ℓ_1	PGM- ℓ_p	DEC-R10G2
results on random-256-1024					
$\lambda = 10^0$	6.9e+2	2.4e+4	7.8e+2	4.0e+2	4.8e+2
$\lambda = 10^1$	2.3e+3	3.8e+4	3.3e+3	1.9e+3	2.2e+3
$\lambda = 10^2$	2.0e+4	1.3e+5	1.8e+4	1.1e+4	9.4e+3
$\lambda = 10^3$	2.5e+4	1.0e+6	2.4e+4	2.4e+4	2.4e+4
results on random-256-2048					
$\lambda = 10^0$	1.3e+3	2.7e+4	1.4e+3	6.0e+2	5.4e+2
$\lambda = 10^1$	2.9e+3	4.5e+4	4.9e+3	2.2e+3	2.2e+3
$\lambda = 10^2$	2.2e+4	2.3e+5	2.1e+4	1.1e+4	9.5e+3
$\lambda = 10^3$	2.7e+4	2.1e+6	2.6e+4	2.7e+4	2.6e+4
results on e2006-5000-1024					
$\lambda = 10^0$	8.5e+3	3.3e+4	1.1e+4	1.8e+4	7.3e+3
$\lambda = 10^1$	9.4e+3	4.2e+4	3.2e+4	3.2e+4	8.6e+3
$\lambda = 10^2$	3.2e+4	1.3e+5	3.2e+4	3.2e+4	1.3e+4
$\lambda = 10^3$	1.8e+4	1.1e+6	3.2e+4	3.2e+4	1.1e+4
results on e2006-5000-2048					
$\lambda = 10^0$	3.1e+3	3.4e+4	4.4e+3	1.4e+4	2.6e+3
$\lambda = 10^1$	5.2e+3	5.3e+4	1.2e+4	1.2e+4	4.5e+3
$\lambda = 10^2$	3.2e+4	2.4e+5	3.2e+4	3.2e+4	7.0e+3
$\lambda = 10^3$	1.8e+4	2.1e+6	3.2e+4	3.2e+4	1.3e+4

	PGM- ℓ_0	APGM- ℓ_0	PGM- ℓ_1	PGM- ℓ_p	DEC-R10G2
results on random-256-1024-C					
$\lambda = 10^0$	9.6e+2	5.7e+6	1.0e+3	1.0e+3	8.9e+2
$\lambda = 10^1$	8.1e+3	3.5e+6	1.0e+4	8.2e+3	7.3e+3
$\lambda = 10^2$	5.8e+4	6.2e+6	8.9e+4	5.4e+4	5.1e+4
$\lambda = 10^3$	2.5e+5	5.3e+6	3.7e+5	2.2e+5	2.0e+5
results on random-256-2048-C					
$\lambda = 10^0$	1.9e+3	5.7e+6	2.0e+3	1.9e+3	1.2e+3
$\lambda = 10^1$	1.7e+4	7.7e+6	2.0e+4	1.6e+4	9.2e+3
$\lambda = 10^2$	8.4e+4	4.2e+6	1.6e+5	6.4e+4	5.3e+4
$\lambda = 10^3$	2.5e+5	9.6e+6	6.3e+5	2.5e+5	2.4e+5
results on e2006-5000-1024-C					
$\lambda = 10^0$	3.0e+4	3.3e+4	2.8e+4	2.9e+4	2.2e+4
$\lambda = 10^1$	3.2e+4	4.2e+4	3.2e+4	3.2e+4	2.3e+4
$\lambda = 10^2$	3.2e+4	1.3e+5	3.2e+4	3.2e+4	2.9e+4
$\lambda = 10^3$	3.2e+4	1.1e+6	3.2e+4	3.2e+4	3.2e+4
results on e2006-5000-2048-C					
$\lambda = 10^0$	2.9e+4	3.4e+4	2.6e+4	2.7e+4	1.7e+4
$\lambda = 10^1$	3.2e+4	5.3e+4	3.2e+4	3.2e+4	2.1e+4
$\lambda = 10^2$	3.2e+4	2.4e+5	3.2e+4	3.2e+4	2.7e+4
$\lambda = 10^3$	3.2e+4	2.1e+6	3.2e+4	3.2e+4	3.2e+4

Table: Comparisons of objective values of all the methods for solving the sparse regularized least squares problem. The 1st, 2nd, and 3rd best results are colored with red, blue and green, respectively.

Sparse Regularized Least Squares Problem

Conclusions:

- **DEC** takes several times longer to converge. (70 seconds)
- The computational time is acceptable and pays off as **DEC** achieves significantly higher accuracy.
- The main bottleneck: small-sized subproblems ($\mathcal{O}(2^k)$).
- k is a parameter to balance the efficacy and efficiency.

	PGM- ℓ_0	APGM- ℓ_0	PGM- ℓ_1	PGM- ℓ_p	DEC-R10G2
r.-256-1024	12 \pm 3	13 \pm 3	5 \pm 3	15 \pm 3	36 \pm 3
r.-256-2048	11 \pm 3	11 \pm 3	9 \pm 3	16 \pm 3	66 \pm 7
e.-5000-1024	12 \pm 3	11 \pm 3	8 \pm 3	14 \pm 3	45 \pm 3
e.-5000-2048	12 \pm 3	10 \pm 3	12 \pm 3	5 \pm 3	65 \pm 8

Table: Comparisons of average times (in seconds) of all the methods on different data sets for solving the sparse regularized least squares problem.

Thank You!