A problem generator for big data optimization

Kimon Fountoulakis and Jacek Gondzio
University of Edinburgh, Edinburgh Research Group in Optimization

Need for Controlled Testing

The big-data era has sparked the development of new optimization methods. Frequently, the performance of the methods is tested on randomly generated problems. Randomly generated instances might be well-conditioned. Hence they may not reveal weaknesses or strengths of methods.
There is a need for the problem generator which allows control of crucial properties/parameters of the problem, such as the conditioning and the size of the problem.

Contribution

A problem generator for

$$\text{minimize } \tau\|x\|_1 + \frac{1}{2}\|Ax-b\|_2^2$$

$\tau > 0$, $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$

- The generator is inexpensive and has a
- low-memory-footprint

The generator allows control of the

- *dimensions m, n*
- *sparsity of A*
- *sparsity of $A^\intercal A$ (independently of A)*
- *singular value decomposition of A*
- *sparsity of the optimal solution x^**
- *values of the optimal solution*

MATLAB implementation

Download from:
`http://www.maths.ed.ac.uk/ERGO/trillion/`
or search on Google: ERGO trillion

A trillion variable example

n	processors	terabytes	seconds
2^{36}	4096	12.288	1970
2^{38}	16384	49.152	1990
2^{40}	65536	196.608	2006

All problems have been solved to a relative error of order 10^{-4} using a Newton-CG method

An example of inexpensive construction of matrix A

Given a singular value matrix $\Sigma \in \mathbb{R}^{m\times n}$ and the Givens rotation matrices $\tilde{G}$ and G we set the singular value decomposition of matrix A to

$$A = \tilde{G}\Sigma G^\intercal.$$

The matrix A does not have to be computed, matrix-vector products can be performed with A through its singular value decomposition.

How we use Givens rotation

Let $G(i,j,\theta) \in \mathbb{R}^{n\times n}$ be a Givens rotation matrix, which rotates plane i-j by an angle θ:

$$G(i,j,\theta) = \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & c & \cdots & -s & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & s & \cdots & c & \cdots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix},$$

where $i, j \in \{1, 2, \cdots, n\}$, $c = \cos\theta$ and $s = \sin\theta$.

We define the following composition of Givens rotations:

$$G = G(i_1, j_1, \theta)G(i_2, j_2, \theta)\cdots, G(i_k, j_k, \theta), \cdots, G(i_{n/2}, j_{n/2}, \theta)$$

where

$$i_k = 2k-1, \quad j_k = 2k \quad \text{for } k = 1, 2, 3, \cdots, n/2$$

and

$$\tilde{G} = \tilde{G}(l_1, p_1, \vartheta)\tilde{G}(l_2, p_2, \vartheta)\cdots, \tilde{G}(l_k, p_k, \vartheta), \cdots, \tilde{G}(l_{m/2}, p_{m/2}, \vartheta)$$

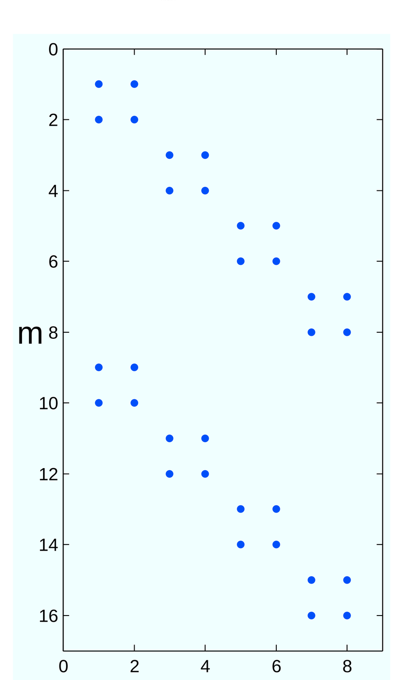
where

$$l_k = k, \quad p_k = m/2 + k \quad \text{for } k = 1, 2, 3, \cdots, m/2.$$

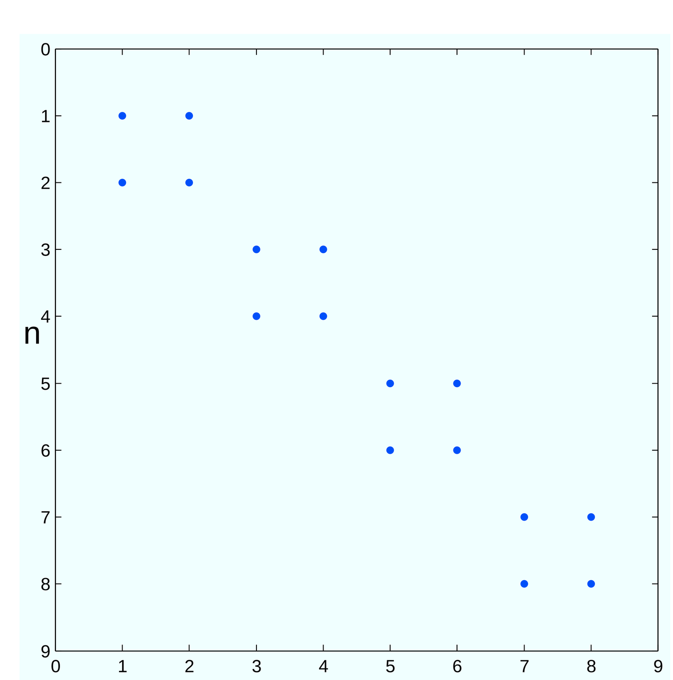
Memory requirements: *a 2×2 matrix, in total!*

Control of sparsity of matrices using Givens rotations

Single composition of Givens rotation

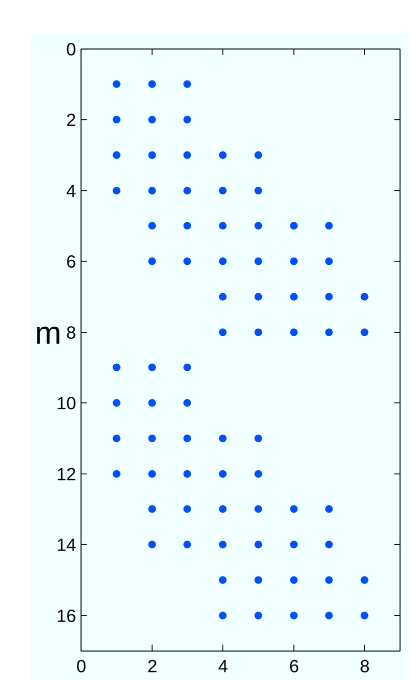


$A = \tilde{G}\Sigma G^\intercal$

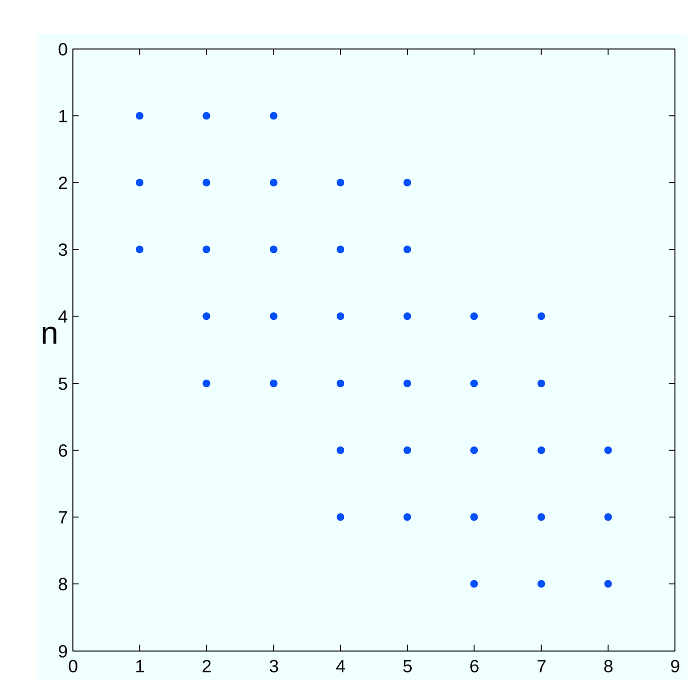


$A^\intercal A$

Double composition of Givens rotation



$A = \tilde{G}\tilde{G}_2\Sigma(GG_2)^\intercal$



$A^\intercal A$

Examples of construction of x^*

Simple example

1. Choose number of nonzeros $s \le \min(m, n)$
2. Choose a subset $S \subseteq \{1, 2, \cdots, n\}$, $|S| = s$.
3. $\forall i \in S$ choose x_i^* uniformly at random in $[-1, 1]$ and $\forall j \notin S$ set $x_j^* = 0$.

Other **non-trivial** examples that make the performance of the state-of-the-art methods degrade are studied in [1].

Construction of b

Initialize τ, x^* and A and generate b such that the optimality conditions are satisfied:

$$A^\intercal(b - Ax^*) \in \tau\partial\|x^*\|_1.$$

How to construct b

1. Initialize $\tau > 0$, $A \in \mathbb{R}^{m\times n}$ with $m \ge n$ and $\text{rank}(A) = n$, $x^* \in \mathbb{R}^n$
2. Pick a subgradient $g \in \partial\|x^*\|_1$:
$$g_i \in \begin{cases} 1, & \text{if } x_i^* > 0 \\ -1, & \text{if } x_i^* < 0 \\ [-1, 1], & \text{if } x_i^* = 0 \end{cases} \quad \forall i = 1, 2, \cdots, n$$
3. Set $e = \tau A(A^\intercal A)^{-1} g$
4. Return $b = Ax^* + e$

References

[1] K. Fountoulakis and J. Gondzio. Performance of First- and Second-Order Methods for Big Data Optimization. *Technical Report ERGO-15-005.*

[2] K. Fountoulakis and J. Gondzio. A Second-Order Method for Strongly Convex ℓ_1-Regularization Problems. *Mathematical Programming (accepted).* doi: 10.1007/s10107-015-0875-4.

Acknowledgements

The authors are grateful to Dr Kenton D' Mellow for providing guidance and useful suggestions regarding the solution of a large scale problem, i.e., 196 terabytes, using ARCHER supercomputer