School of Mathematics

Applications of IPMs: From Sparse Approximations to Discrete Optimal Transport

Jacek Gondzio
Email: J.Gondzio@ed.ac.uk
URL: http://www.maths.ed.ac.uk/~gondzio

Outline

- Motivation: **sparsity**, a desired feature
  $\longrightarrow$ for example, $\boldsymbol{\ell_1}$-regularized least squares (LASSO)
- 1st-order vs 2nd-order methods
- Inexact Newton method
  - How much of Hessian information is needed?
  - Iterative methods with suitable **preconditioners**
    $\longrightarrow$ Newton Conjugte Gradients
    $\longrightarrow$ (Inexact) Interior Point Methods
- Applications
- Conclusions

Sparse Approximations

- Statistics: Estimate x from observations
- Machine Learning: Classifications, SVMs, etc
- Inverse Problems
- Wavelet-based signal/image reconstruction & restoration
- Compressed Sensing (Signal Processing)

Such problems lead to some *dense*, often *structured*, possibly *very large* optimization instances (LP, QP or NLP):

$$\begin{array}{ll} \min_x & f(x) + \tau_1 \|x\|_1 + \tau_2 \|Lx\|_1 \\ \text{s.t.} & Ax \;=\; b. \end{array}$$

Cutting-edge optimization techniques are needed!
Plethora of highly **specialised 1st-order methods** exist.
Work of **Yu. Nesterov, S. Wright** and an army of followers.

1st-order methods vs 2nd-order methods

The 2nd-order methods are sometimes criticised as unsuitable: "computing/using the 2nd-order information is too expensive".

An **unfounded criticism** based on an **unfair comparison**:
specialised 1st-order methods compared with
general (of-the-shelf) 2nd-order methods.

The 1st-order methods have clear drawbacks:

- they struggle with accuracy, and
- they work only for trivial, well conditioned problems.

The **specialised 2nd-order methods**
overcome these drawbacks and are very competitive.

This talk will demonstrate why.

ℓ_1-regularization

$$\min_x \quad f(x) + \tau \|x\|_1.$$

Think of LASSO:

$$\min_x \; \|Ax - b\|_2^2 + \tau \|x\|_1.$$

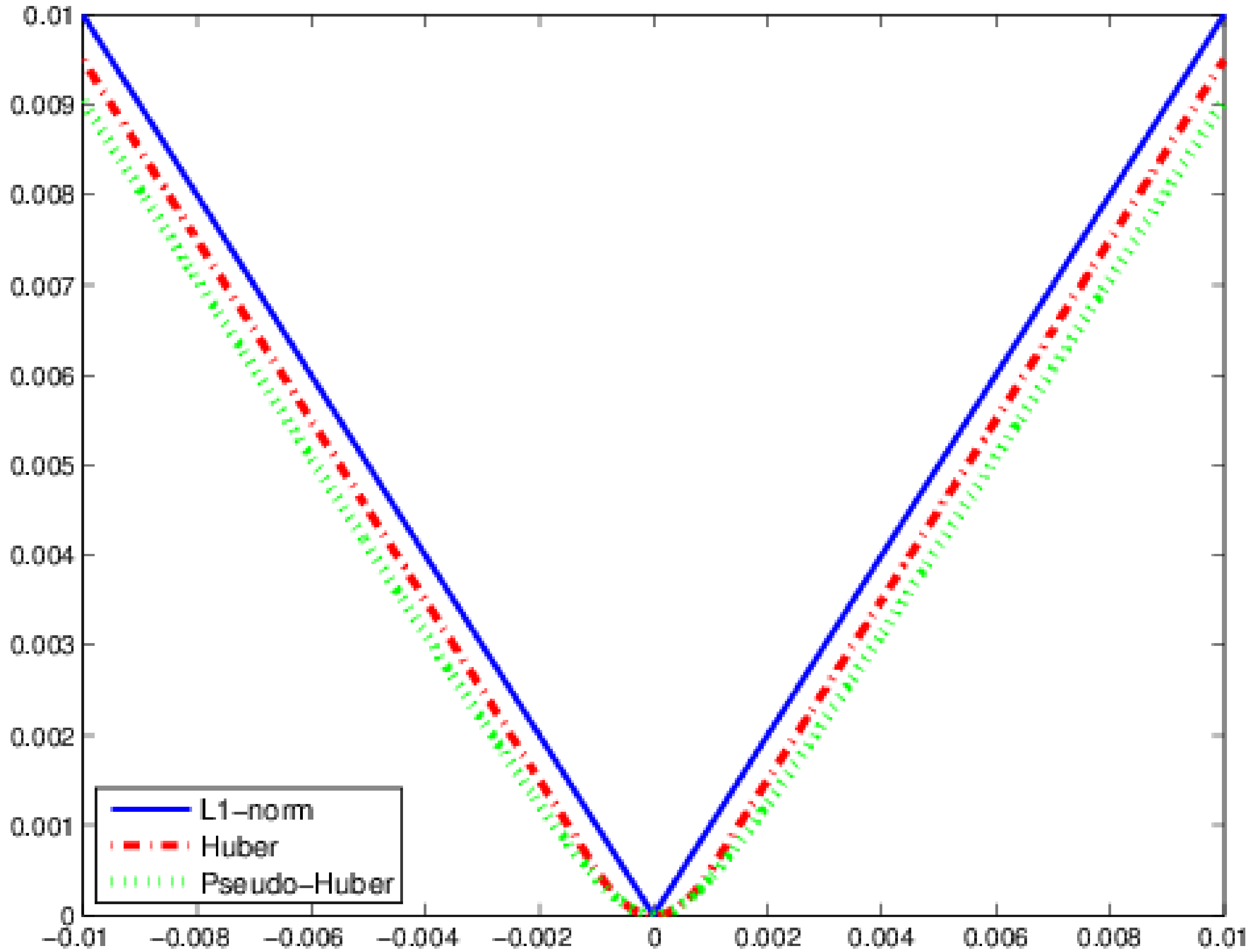
Unconstrained optimization $\Rightarrow$ easy

Serious Issue: nondifferentiability of $\|.\|_1$

Two possible tricks:

- Splitting $x = u - v$ with $u, v \geq 0$
- Smoothing with pseudo-Huber approximation replaces $\|x\|_1$ with $\psi_\mu(x) = \sum_{i=1}^n (\sqrt{\mu^2 + x_i^2} - \mu)$

Huber:



0.01
0.009
0.008
0.007
0.006
0.005
0.004
0.003
0.002
0.001
0
−0.01
−0.008
−0.006
−0.004
−0.002
0
0.002
0.004
0.006
0.008
0.01
L1−norm
Huber
Pseudo−Huber

Continuation

Embed inexact Newton Method into a *homotopy* approach:

- Inequalities $u \geq 0,\ v \geq 0$ $\longrightarrow$ use **IPM**

  replace $z \geq 0$ with $-\mu\, log z$ and drive μ to zero.

- Pseudo-Huber regression $\longrightarrow$ use **continuation**

  replace $|x_i|$ with $\mu(\sqrt{1+\frac{x_i^2}{\mu^2}}-1)$ and drive μ to zero.

Questions:

- How?
- Theory?
- Practice?

How: Use approximate Hessian

Use 2nd-order information (Newton direction).

But, do not waste time on computing *exact* direction.

Use Inexact Newton Method

Dembo, Eisenstat and Steihaug,
Inexact Newton Methods,
SIAM J. on Numerical Analysis 19 (1982) 400–408.

Bellavia, Inexact Interior Point Method,
Journal of Optimization Theory and Appls 96 (1998) 109–121.

Inexact Newton Method

Replace an *exact* Newton direction

$$\nabla^2 f(x)\Delta x = -\nabla f(x)$$

with an *inexact* one:

$$\nabla^2 f(x)\Delta x = -\nabla f(x) + \boldsymbol{r},$$

where the *error* $\boldsymbol{r}$ is small: $\|\boldsymbol{r}\| \leq \boldsymbol{\eta}\|\nabla f(x)\|$, $\boldsymbol{\eta} \in (0,1)$.

Use iterative methods of linear algebra:

- Continuation $\rightarrow$ Newton CG
- IPMs $\rightarrow$ Inexact IPM $\rightarrow$ Iterative schemes for KKT systems

IMPs: Theorem: Suppose the feasible IPM for QP is used.

If the method operates in the *small* neighbourhood

$$\mathcal{N}_2(\theta) := \{(x,y,s) \in \mathcal{F}^0 : \|XSe - \mu e\|_2 \leq \theta\mu\}$$

and uses the *inexact* Newton direction with $\boldsymbol{\eta} = 0.3$, then it converges in at most

$$K = \mathcal{O}(\sqrt{\boldsymbol{n}}\ \ln(1/\epsilon)) \quad \text{iterations.}$$

If the method operates in the *symmetric* neighbourhood

$$\mathcal{N}_S(\gamma) := \{(x,y,s) \in \mathcal{F}^0 : \gamma\mu \leq x_i s_i \leq (1/\gamma)\mu\}$$

and uses the *inexact* Newton direction with $\boldsymbol{\eta} = 0.05$, then it converges in at most

$$K = \mathcal{O}(\boldsymbol{n}\ \ln(1/\epsilon)) \quad \text{iterations.}$$

Gondzio, Convergence Analysis of an Inexact Feasible IPM for Convex Quadratic Programming, *SIAM Journal on Optimization* 23 (2013) No 3, pp. 1510-1527.

Continuation: Compressed Sensing Case

Replace $\min_x \; f(x) = \tau \|W^T x\|_1 + \frac{1}{2}\|Ax - b\|_2^2, \quad \longrightarrow \boldsymbol{x_\tau}$

with $\min_x \; f_\mu(x) = \tau \psi_\mu(W^T x) + \frac{1}{2}\|Ax - b\|_2^2, \quad \longrightarrow \boldsymbol{x_{\tau,\mu}}$

Solve approximately a family of problems for a (short) decreasing sequence of μ's: $\boldsymbol{\mu_0 > \mu_1 > \mu_2 \cdots}$

Theorem (Brief description)

There exists a $\tilde{\mu}$ such that $\forall \mu \leq \tilde{\mu}$ the difference of the two solutions satisfies

$$\|x_{\tau,\mu} - x_\tau\|_2 = \mathcal{O}(\mu^{1/2}) \quad \forall \, \tau, \mu.$$

Primal-Dual Newton Conjugate Gradient Method:

Fountoulakis and Gondzio, A Second-order Method for Strongly Convex ℓ_1-regularization Problems, *Mathematical Programming*, 156 (2016) 189–219.

Dassios, Fountoulakis and Gondzio, A Preconditioner for a Primal-Dual Newton Conjugate Gradient Method for Compressed Sensing Problems, *SIAM J on Scientific Computing*, 37 (2015) A2783–A2812.

Examples

Examples of ℓ_1-regularization

- Compressed Sensing
  with **K. Fountoulakis and P. Zhlobich**

$$\min_x \; \tau \|x\|_1 + \frac{1}{2}\|Ax - b\|_2^2, \quad A \in \mathcal{R}^{m \times n}$$

- Compressed Sensing (Coherent and Redundant Dict.)
  with **I. Dassios and K. Fountoulakis**

$$\min_x \; \tau \|W^* x\|_1 + \frac{1}{2}\|Ax - b\|_2^2, \quad W \in \mathcal{C}^{n \times l}, \, A \in \mathcal{R}^{m \times n}$$

  think of Total Variation

- Big Data optimization (Machine Learning), LASSO
  with **K. Fountoulakis**

Example 1: Compressed Sensing

with **K. Fountoulakis and P. Zhlobich**

Large dense quadratic optimization problem:

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

where $A \in \mathcal{R}^{m \times n}$ is a **very special matrix**.

Fountoulakis, Gondzio, Zhlobich
Matrix-free IPM for Compressed Sensing Problems,
Mathematical Programming Computation 6 (2014), pp. 1–31.

Dassios, Fountoulakis, Gondzio
A Preconditioner for a Primal-Dual Newton Conjugate Gradient Method for Compressed Sensing Problems,
SIAM J on Scientific Computing 37 (2015) A2783–A2812.

Software available at `http://www.maths.ed.ac.uk/ERGO/`

Restricted Isometry Property (RIP)

- *rows* of A are orthogonal to each other (A is built of a subset of rows of an othonormal matrix $U \in \mathcal{R}^{n \times n}$)

$$AA^T = I_m.$$

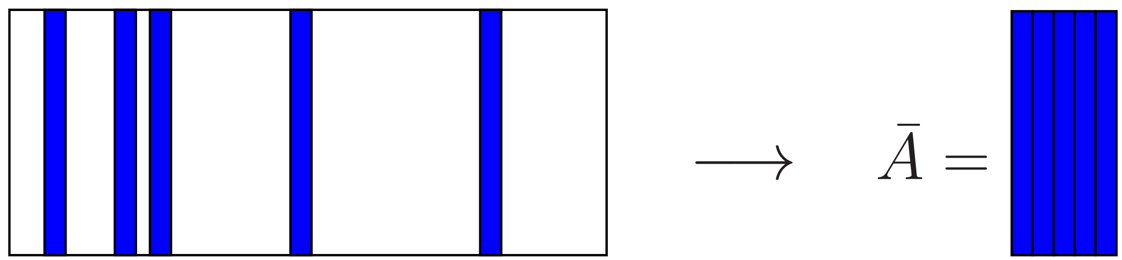
- small subsets of *columns* of A are nearly-orthogonal to each other: *Restricted Isometry Property (RIP)*

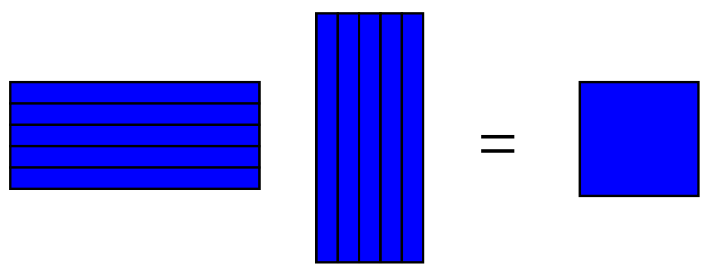
$$\|\bar{A}^T \bar{A} - \frac{m}{n} I_k\| \leq \delta_k \in (0,1).$$

Candès, Romberg and Tao, Stable Signal Recovery from Incomplete and Inaccurate Measurements, *Comm on Pure and Applied Mathematics* 59 (2006) 1207-1233.

Restricted Isometry Property

Matrix $\bar{A} \in \mathcal{R}^{m \times k}$ $(k \ll n)$ is built of a subset of columns of $A \in \mathcal{R}^{m \times n}$.

$A =$  $\bar{A} =$

$\bar{A}^T \bar{A} =$  $\approx \frac{m}{n} I_k.$

This yields a very well conditioned optimization problem.

Problem Reformulation

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

Replace $x = x^+ - x^-$ to be able to use $|x| = x^+ + x^-$.
Use $|x_i| = z_i + z_{i+n}$ to replace $\|x\|_1$ with $\|x\|_1 = 1_{2n}^T z$.
(Increases problem dimension from n to $2n$.)

$$\min_{z \geq 0} \ c^T z + \frac{1}{2} z^T Q z,$$

where

$$Q = \begin{bmatrix} A^T \\ -A^T \end{bmatrix} \begin{bmatrix} A & -A \end{bmatrix} = \begin{bmatrix} A^T A & -A^T A \\ -A^T A & A^T A \end{bmatrix} \in \mathcal{R}^{2n \times 2n}$$

Preconditioner

Approximate

$$\mathcal{M} = \begin{bmatrix} A^T A & -A^T A \\ -A^T A & A^T A \end{bmatrix} + \begin{bmatrix} \Theta_1^{-1} & \\ & \Theta_2^{-1} \end{bmatrix}$$

with

$$\mathcal{P} = \frac{m}{n} \begin{bmatrix} I_n & -I_n \\ -I_n & I_n \end{bmatrix} + \begin{bmatrix} \Theta_1^{-1} & \\ & \Theta_2^{-1} \end{bmatrix}.$$

We expect (*optimal partition*):

- k entries of $\Theta^{-1} \rightarrow 0, \quad k \ll 2n,$
- $2n - k$ entries of $\Theta^{-1} \rightarrow \infty.$

Spectral Properties of $\mathcal{P}^{-1}\mathcal{M}$

Theorem

- Exactly n eigenvalues of $\mathcal{P}^{-1}\mathcal{M}$ are 1.
- The remaining n eigenvalues satisfy

$$|\lambda(\mathcal{P}^{-1}\mathcal{M}) - 1| \leq \delta_k + \frac{n}{m\delta_k L},$$

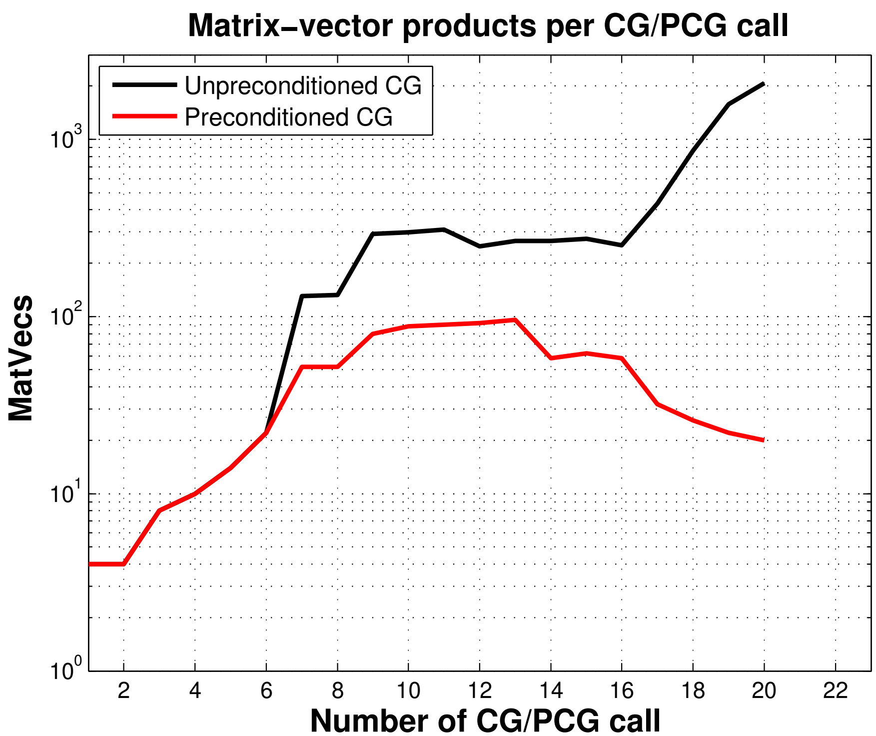
where δ_k is the RIP-constant, and
L is a threshold of "large" $(\Theta_1 + \Theta_2)^{-1}$.

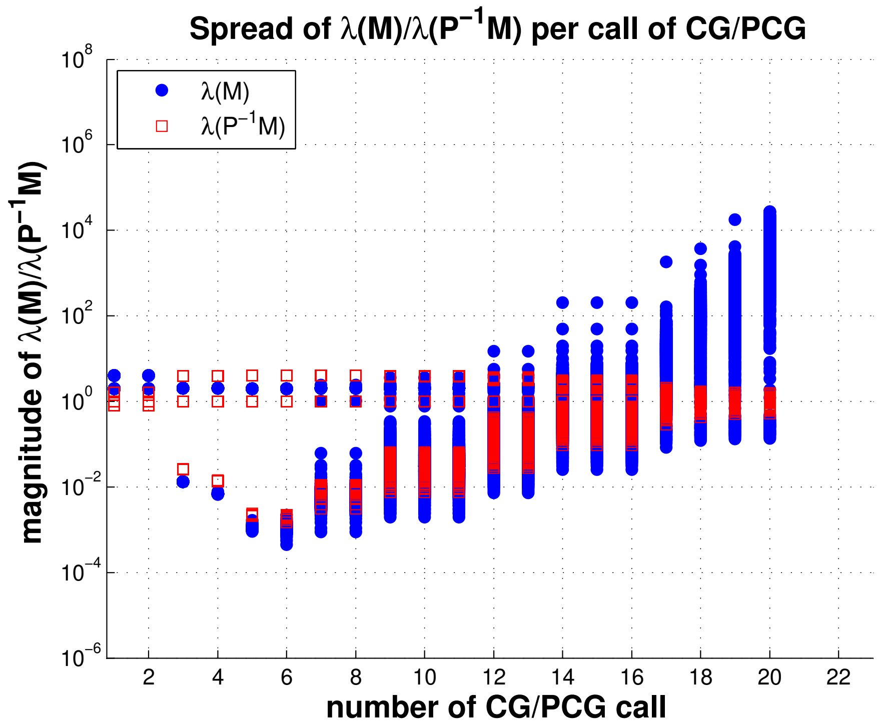
Fountoulakis, Gondzio, Zhlobich

Matrix-free IPM for Compressed Sensing Problems,

Mathematical Programming Computation 6 (2014), pp. 1–31.

Preconditioning





$\longrightarrow$ good clustering of eigenvalues

`mf-IPM` compares favourably with `NestA` on easy probs
(`NestA`: Becker, Bobin and Candés).

Example 2: Simple test for ℓ_1-regularization

$$\min_x \ \tau \|x\|_1 + \|Ax - b\|_2^2$$

Special matrix given in SVD form $A = U \Sigma V^T$, where U and V are products of Givens rotations. The user controls:

- the condition number $\kappa(A)$,
- the sparsity of matrix A.

Matlab generator:
`https://www.maths.ed.ac.uk/ERGO/trillion/`

Fountoulakis and Gondzio
Performance of First- and Second-Order Methods for ℓ_1-regularized Least Squares Problems, *Computational Optimization and Applications* 65 (2016) 605–635.

Excessive Computational Tests (4 mths of CPU)

- FISTA (Fast Iterative Shrinkage-Thresholding Algorithm)
- PCDM (Parallel Coordinate Descent Method)
- PSSgb (Projected Scaled Subgradient, Gafni-Bertsekas)
- pdNCG (primal-dual Newton Conjugate Gradient)

The **1st order** methods:

- work well if the condition number $\kappa(A) \leq 10^2$,
- struggle when $\kappa(A) \geq 10^3$,
- stall when $\kappa(A) \geq 10^4$.

The **2nd order** method (pdNCG, diagonal preconditioner):

- works well if the condition number $\kappa(A) \leq 10^6$.

Let us go big: a trillion ($2^{40} \approx 10^{12}$) variables

n (billions)	Processors	Memory (TB)	time (s)
1	64	0.192	1923
4	256	0.768	1968
16	1024	3.072	1986
64	4096	12.288	1970
256	16384	49.152	1990
1,024	65536	196.608	2006

ARCHER (ranked 25 on `top500.com`, 11 March 2015)

Linpack Performance (Rmax) 1,642.54 TFlop/s
Theoretical Peak (Rpeak) 2,550.53 TFlop/s

Fountoulakis and Gondzio
Performance of First- and Second-Order Methods for ℓ_1-regularized Least Squares Problems,
Computational Optimization and Applications 65 (2016) 605–635.

More Examples of Sparse Approximations

- Sparse Approximations with IPMs
  → ℓ_1-regularized problems,
  work with **V. De Simone, D. di Serafino, S. Pougkakiotis, M. Viola**

- Discrete Optimal Transport with IPMs
  → large, but highly structured,
  work with **F. Zanetti**

More Sparse Approximations: Use IPMs

Problems of the form

$$\begin{array}{ll} \min & f(x) + \tau_1 \|x\|_1 + \tau_2 \|Lx\|_1 \\ \text{s.t.} & Ax = b. \end{array}$$

- Sparse portfolio selection
  comparison with Split Bregman method
- Classification models for funct'l Magnetic Resonance Imaging
  comparison with FISTA and ADMM
- TV-based Poisson Image Restoration
  comparison with PDAL
- Linear Classification via Regularized Logistic Regression
  comparison with newGLMNET and ADMM

De Simone, di Serafino, Gondzio, Pougkakiotis, Viola,
Sparse Approximations with Interior Point Methods,
SIAM Review 64 (2022) pp. 954–988. https://arxiv.org/abs/2102.13608

Example 3: Binary Classification of fMRI Data

$$\min_{w} \; \frac{1}{2s} \|Dw - \hat{y}\|^2 + \tau_1 \|w\|_1 + \tau_2 \|Lw\|_1$$

where: $\tau_1, \tau_2 > 0$, $\|Lw\|_1$ is a discrete anisotropic TV of w,

and $L = [L_x^T \;\; L_y^T \;\; L_z^T]^T \in \mathcal{R}^{l \times q}$ are the first-order forward finite differences in x, y, z.

Baldassarre, Pontil & Mouraõ-Miranda,

Sparsity Is Better with Stability: Combining Accuracy and Stability for Model Selection in Brain Decoding,

Frontiers of Neuroscience 2017. https://doi.org/10.3389/fnins.2017.00062

Classification models for fMRI

Comparison of IPM, FISTA and ADMM (opt tol 10^{-5}). We report:

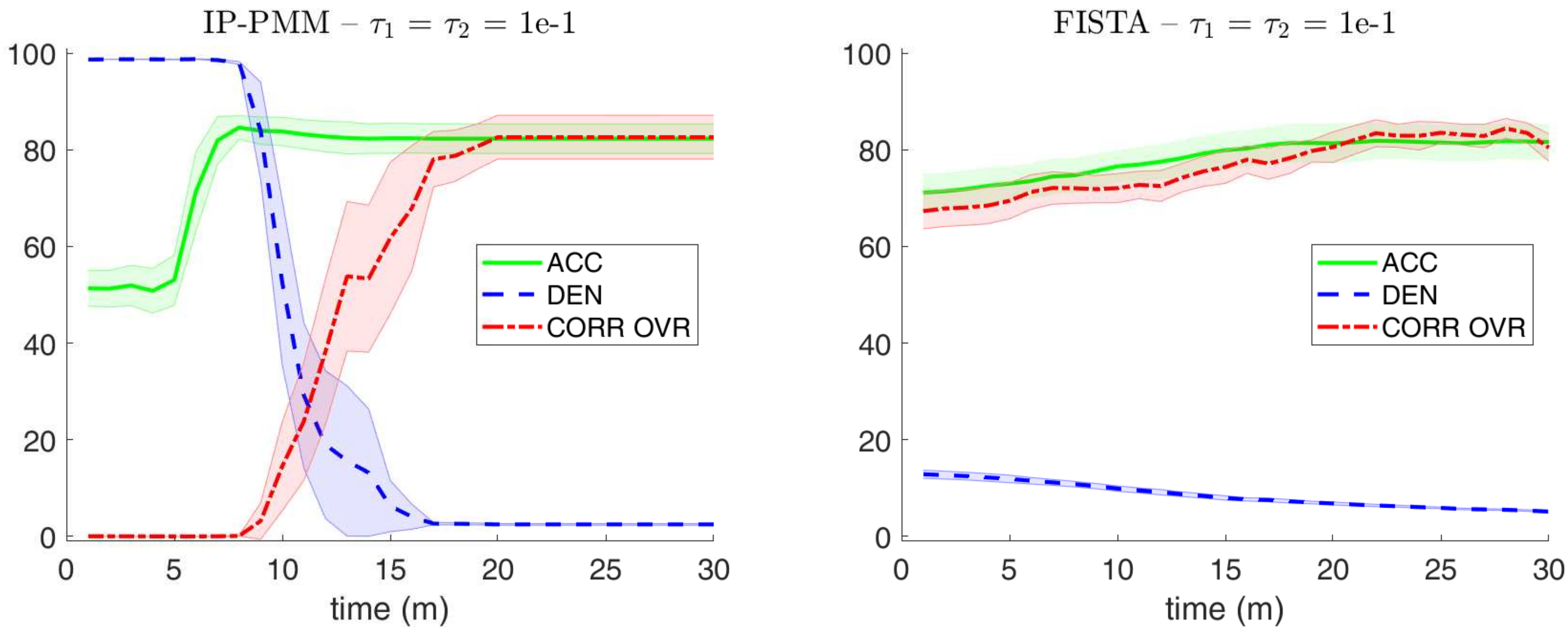
- *classification accuracy* (ACC),
- *corrected pairwise overlap* (CORR OVR); measures the "stability" of each voxel selection,
- *solution density* (DEN).

Algorithm	$\tau_1 = \tau_2$	ACC	CORR OVR	DEN
IP-PMM	10^{-2}	86.16 ± 7.11	43.47 ± 9.09	20.56 ± 6.63
	$5 \cdot 10^{-2}$	84.90 ± 4.80	62.70 ± 10.39	3.77 ± 0.84
	10^{-1}	82.29 ± 6.22	82.60 ± 9.24	2.49 ± 0.34
FISTA	10^{-2}	86.90 ± 5.01	5.43 ± 0.43	88.97 ± 0.71
	$5 \cdot 10^{-2}$	84.15 ± 5.92	65.50 ± 2.68	19.36 ± 0.86
	10^{-1}	81.62 ± 7.58	80.44 ± 5.72	5.14 ± 0.44
ADMM	10^{-2}	86.46 ± 6.91	0.03 ± 0.01	98.70 ± 0.03
	$5 \cdot 10^{-2}$	85.57 ± 5.37	0.15 ± 0.04	97.97 ± 0.05
	10^{-1}	82.07 ± 6.51	0.26 ± 0.13	97.50 ± 0.19

We want: ACC and CORR OVR *close to 100*, and *small* DEN.

Classification models for fMRI (cont'd)

Performance comparison in terms of elapsed time:



Evolution of ACC, DEN and CORR OVR with time;
IP-PMM (*left*) and FISTA (*right*).
We report average measures with 95% confidence intervals.

Optimal Transport

Significant research interest:

Gaspard Monge (1781)
Leonid Kantorovich (1942) **Nobel Prize in 1975**
Alessio Figalli (2008) **Fields Medal in 2018**

Good reading:

F. Santambrogio,
Optimal Transport for Applied Mathematicians, Birkhauser Basel, 2016.

G. Peyré and M. Cuturi,
Computational Optimal Transport: With Applications to Data Science. *Foundations and Trends in Machine Learning* 11 (2019) No 5-6, pp. 355–607.

Example 4: Discrete Optimal Transport

Kantorovich formulation of the discrete Optimal Transport problem: given a starting vector $\mathbf{a} \in \mathcal{R}^m_+$ and a final vector $\mathbf{b} \in \mathcal{R}^n_+$, such that $\sum \mathbf{a}_j = \sum \mathbf{b}_j$, find a coupling matrix $\mathcal{P}$ inside the set

$$U(\mathbf{a}, \mathbf{b}) = \left\{ \mathcal{P} \in \mathcal{R}^{m \times n}_+, \ \mathcal{P}\mathbf{e}_n = \mathbf{a}, \ \mathcal{P}^T \mathbf{e}_m = \mathbf{b} \right\}$$

that is optimal with respect to a certain cost matrix $\mathcal{C} \in \mathcal{R}^{m \times n}_+$; i.e. find the solution of the following optimization problem

$$\min_{\mathcal{P} \in U(\mathbf{a},\mathbf{b})} \sum_{i,j} \mathcal{C}_{ij} \mathcal{P}_{ij}.$$

Move the mass in the configuration $\mathbf{a}$ into the configuration $\mathbf{b}$.

Discrete Optimal Transport (cont'd)

We can rewrite the optimization problem as a standard LP:

$$\begin{aligned} \min_{\mathbf{p}\in\mathcal{R}^{mn}} \quad & \mathbf{c}^T\mathbf{p} \\ \text{s.t.} \quad & \begin{bmatrix} \mathbf{e}_n^T \otimes I_m \\ I_n \otimes \mathbf{e}_m^T \end{bmatrix} \mathbf{p} = \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix} = \mathbf{f}, \\ & \mathbf{p} \geq 0, \end{aligned}$$

where $\otimes$ denotes the Kronecker product,
$\mathbf{c} \in \mathcal{R}^{mn}$ and $\mathbf{p} \in \mathcal{R}^{mn}$ are the vectorized versions of $\mathcal{C}$ and $\mathcal{P}$, respectively, $\mathbf{c} = \mathrm{vec}(\mathcal{C})$ and $\mathbf{p} = \mathrm{vec}(\mathcal{P})$.

LP with *$m + n$ constraints* and *$m \times n$ variables*.

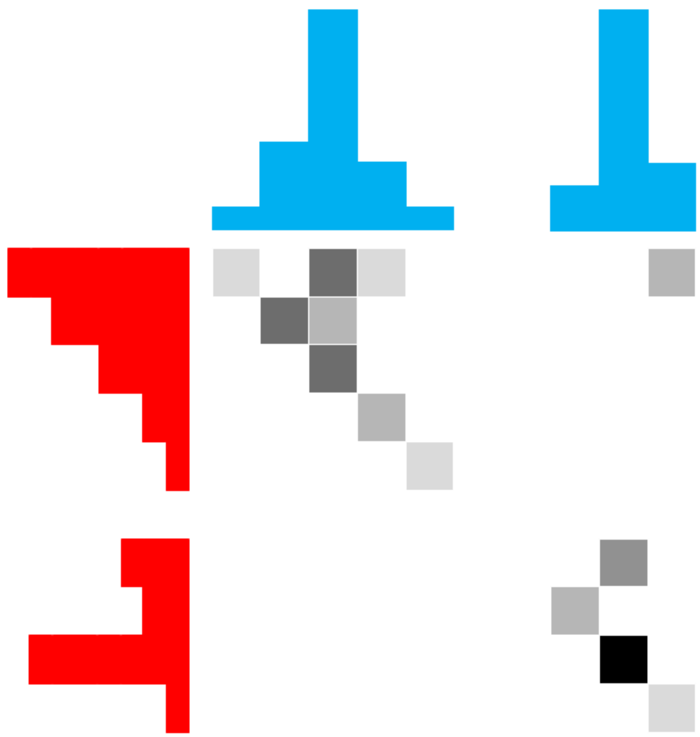
Zanetti and Gondzio,

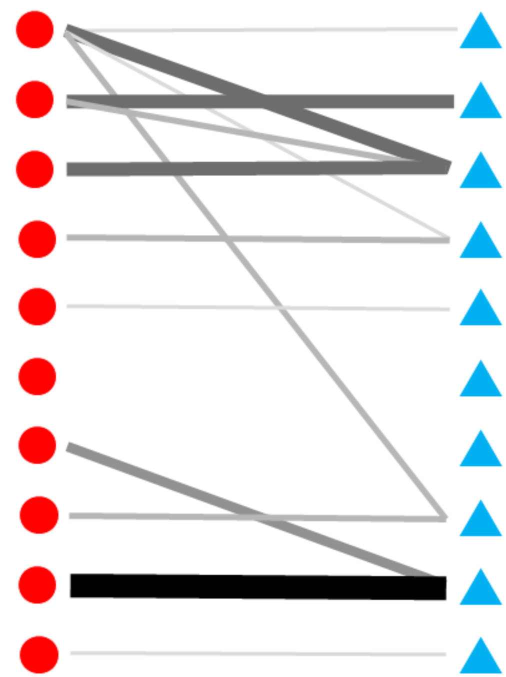
A Sparse Interior Point Method for Linear Programs arising in Discrete Optimal Transport,

(submitted 22 Jun 2022, revised 6 Dec 2022). https://arxiv.org/abs/2206.11009

Small OT Example

Move the mass in the red configuration into the blue configuration. Right figure: the corresponding bipartite graph. → Sparse solution!

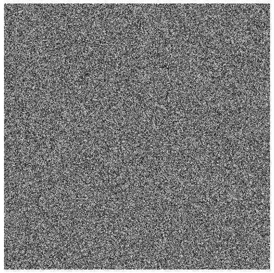


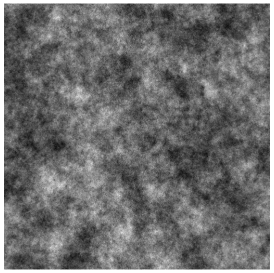


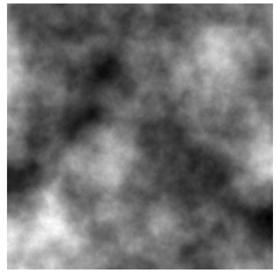
IPM Specialized for Discrete OT Problems

- Ignore "long" matrix A
  $\longrightarrow$ use column-generation-type approach
- Work with expected "sparse" solution set
  $\longrightarrow$ do not update all variables x
- Use simplex-type pricing mechanism
  $\longrightarrow$ update dual slacks only for a subset of variables x
- Simplify normal equations
  $\longrightarrow$ replace $\sum_{j=1}^{N} \theta_j A_j A_j^T$ with $\sum_{j \in \mathcal{S}} \theta_j A_j A_j^T$,
  where $\mathcal{S}$ is a likely "sparse" solution set
- Precondition Cholesky matrix of the normal equations
  $\longrightarrow$ keep it sparse at all times

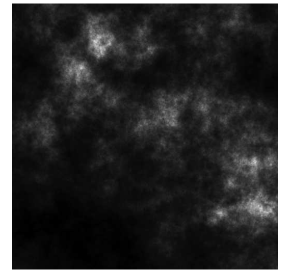
Test examples from DOTmark collection


Class 1

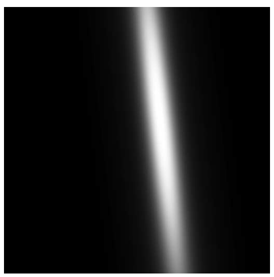

Class 2

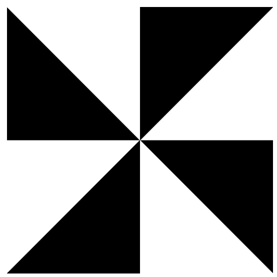

Class 3


Class 4

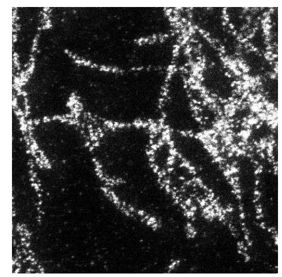

Class 5


Class 6


Class 7


Class 8


Class 9


Class 10

For the resolution r, the LP has $2r^2$ *constraints* and r^4 *variables*.
For $r = 32$: 2,048 constraints and 1 million variables;
For $r = 64$: 8,192 constraints and 16.8 million variables;
For $r = 128$: 32,768 constraints and 268.4 million variables;
For $r = 256$: 131,072 constraints and 4.295 billion variables.

Discrete Optimal Transport (cont'd)

DOTmark test collection:

Schrieber, Schuhmacher, and Gottschlich,

DOTmark - A Benchmark for Discrete Optimal Transport, *IEEE Access*, 5 (2017), pp. 271–282.

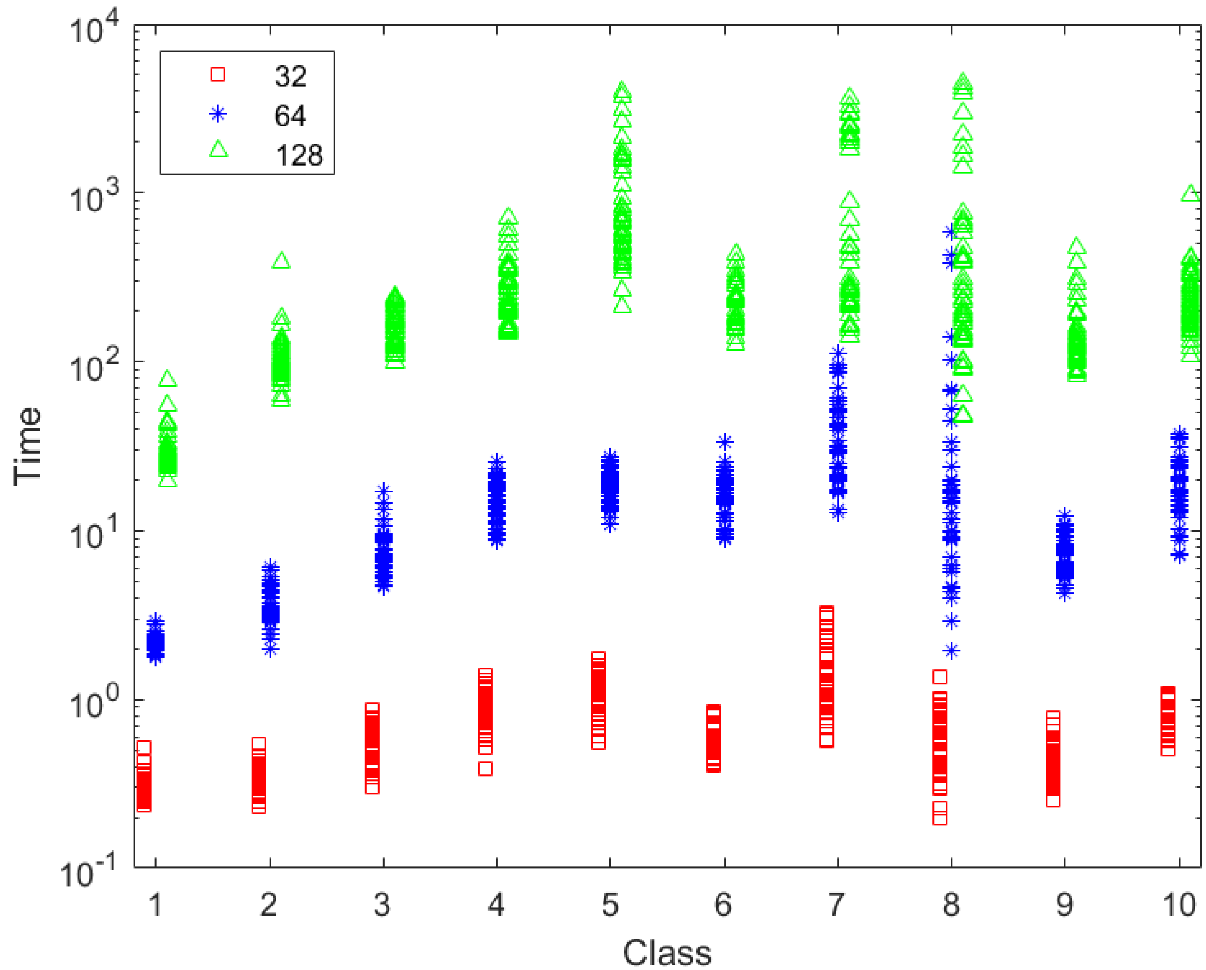
Softwares compared:

- **Cuturi**,
  Sinkhorn distances: Lightspeed computation of optimal transport,
  Proc. NIPS, (2013), pp. 2292–2300.
- **Gottschlich and Schuhmacher**,
  The Shortlist Method for Fast Computation of the Earth Mover's Distance and Finding Optimal Solutions to Transportation Problems,
  PLoS ONE, 9 (2014), p. e110214.
- **Merigot**,
  A Multiscale Approach to Optimal Transport,
  Computer Graphics Forum, 30 (2011), pp. 1583–1592.
- Network Simplex Method, IBM ILOG CPLEX.
  `https://www.ibm.com/analytics/cplex-optimizer`.
- **Kovacs**,
  Minimum-cost flow algorithms: An experimental evaluation, *OMS*, 30(1):94–127.
  `https://lemon.cs.elte.hu/trac/lemon`.

Comparison: SparseIPM vs Cplex Network

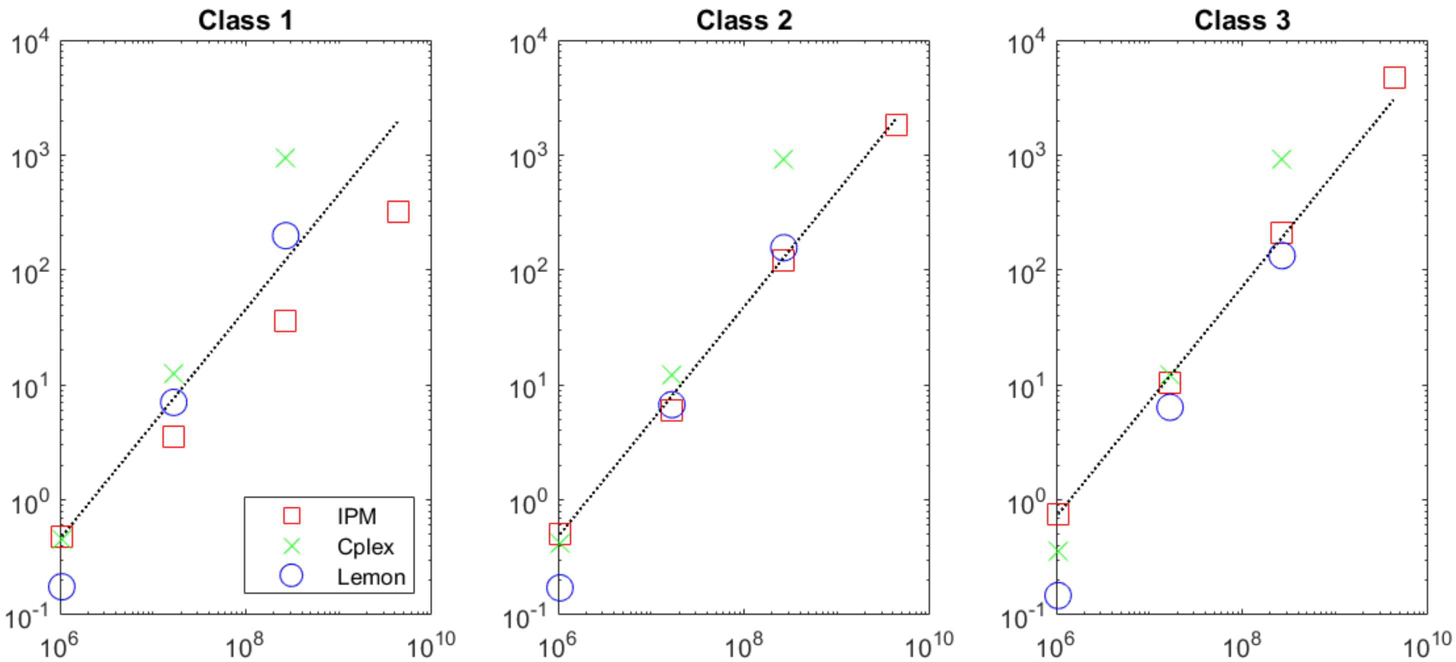
	Res = 32×32				Res = 64×64			
Class	Iter	IPM t	Cplex t	RWE	Iter	IPM t	Cplex t	RWE
1	11.4	0.35	0.62	1.2e-07	14.4	2.18	20.92	5.5e-08
2	11.7	0.39	0.60	1.4e-07	18.1	3.46	20.64	4.5e-08
3	15.9	0.59	0.61	2.4e-08	26.8	6.02	20.83	2.1e-08
4	20.3	0.85	0.57	2.0e-08	38.4	9.69	20.69	2.1e-08
5	25.6	1.16	0.61	1.4e-08	40.8	10.78	21.84	1.6e-08
6	18.8	0.72	0.64	3.3e-08	36.2	9.04	23.25	1.3e-08
7	30.8	1.47	0.57	3.8e-08	72.2	39.11	21.80	2.3e-08
8	17.4	0.65	0.58	3.8e-08	52.5	21.69	18.55	8.7e-08
9	14.9	0.52	0.60	2.8e-08	25.0	5.24	21.27	1.4e-08
10	22.4	0.92	0.62	2.0e-08	40.8	10.48	18.33	2.1e-08

CPU time of SparseIPM (1–norm, 128 pixels)



$$m = 2r^2$$
$$n = r^4$$

Comparison: Scalability of three solvers



SparseIPM for Discrete OT
Cplex (Simplex Method for Network Problems)
LEMON (Specialized Network Algorithm)

Overarching Feature of IPMs

They possess an unequalled ability to identify
the **"essential subspace"**
in which the optimal solution is hidden.

Conclusions

2nd-order methods for optimization:

- employ **inexact Newton method**
- rely on **preconditioners**
- enjoy **matrix-free** implementation

Trick:

- find the "essential subspace" and
- exploit it to simplify the linear algebra
  - works in IPMs for LP
  - works in Newton CG for ℓ_1-regularization

Simple, reliable test example for ℓ_1-regularization:
`http://www.maths.ed.ac.uk/ERGO/trillion/`