Second Order Methods for L1-Regularization

Kimon Fountoulakis and Jacek Gondzio

with extra thanks to

Kristian Woodsend and Pavel Zlobich

Outline

- Motivation: Why not *2nd-order* methods?
- Interior Point Methods and Continuation
- Inexact Newton directions
  - Krylov subspace methods
  - Preconditioner is a must
- Computational results
  - Compressed Sensing
  - Google Problem
  - Machine Learning Problems
- Linear Algebra viewpoint on ℓ_1-regularization
- Conclusions

ℓ_1 regularization

Convex optimization problem:

$$\min_x \quad \tau\|x\|_1 + \phi(x),$$

where $\|.\|_1$ is the ℓ_1 norm, and
$\phi : \mathcal{R}^n \mapsto \mathcal{R}$ is a convex function (often strongly convex).

Usual example:

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

where $A \in \mathcal{R}^{m \times n}$ (often $m \geq n$ or $m \gg n$).

Two features:

Difficulty:
non-differentiability of $\|x\|_1$

Triviality:
unconstrained optimization

It is fashionable to use the 1st-order methods to solve these problems. Marketed as *Haute Couture.*

Prête-à-porter. What about the 2nd-order methods???

Observation

- First-order methods
  - complexity $\mathcal{O}(1/\varepsilon)$ or $\mathcal{O}(1/\varepsilon^2)$
  - produce a rough approx. of solution quickly
  - but ... struggle to converge to high accuracy
- IPMs are second-order methods
  (they apply Newton method to barrier subprobs)
  - complexity $\mathcal{O}(\log(1/\varepsilon))$
  - produce accurate solution in a few iterations
  - but ... one iteration may be expensive

Just think

For example, $\varepsilon = 10^{-3}$ gives
$1/\varepsilon = 10^3$ and $1/\varepsilon^2 = 10^6$, but $\log(1/\varepsilon) \approx 7$.

For example, $\varepsilon = 10^{-6}$ gives
$1/\varepsilon = 10^6$ and $1/\varepsilon^2 = 10^{12}$, but $\log(1/\varepsilon) \approx 14$.

But **ML Community** loves the 1st-order methods.

Stirring up a hornets nest:

Give 2nd-order/IPMs a serious consideration!

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

Two possible tricks:

- Splitting $x = u - v$ with $u, v \geq 0$
- Huber or pseudo-Huber regression

Splitting: $\boldsymbol{x} = \boldsymbol{u} - \boldsymbol{v},\ \boldsymbol{u} \geq \mathbf{0},\ \boldsymbol{v} \geq \mathbf{0}$

Replace $x_i = u_i - v_i$,
where $u_i = \max\{x_i, 0\}$ and $v_i = \max\{-x_i, 0\}$.

Then $x_i = u_i - v_i$ and $|x_i| = u_i + v_i$.

Hence $\|x\|_1 = \sum_{i=1}^{n}(u_i + v_i)$.

Removes nondifferentiability, but:

- doubles the dimension,
- introduces inequality constraints (fine for IPMs).

Huber: **Replace $\|\mathbf{x}\|_1$ with $\boldsymbol{\psi_\mu(x)}$**

Huber approximation: replaces $\|x\|_1$ with $\sum_{i=1}^n \Big[\phi_\mu(x)\Big]_i$

$$\Big[\phi_\mu(x)\Big]_i = \begin{cases} \frac{1}{2}\mu^{-1}x_i^2, & \text{if } |x_i| \le \mu \\ |x_i| - \frac{1}{2}\mu, & \text{if } |x_i| \ge \mu \end{cases} \qquad i = 1, 2, \ldots, n$$

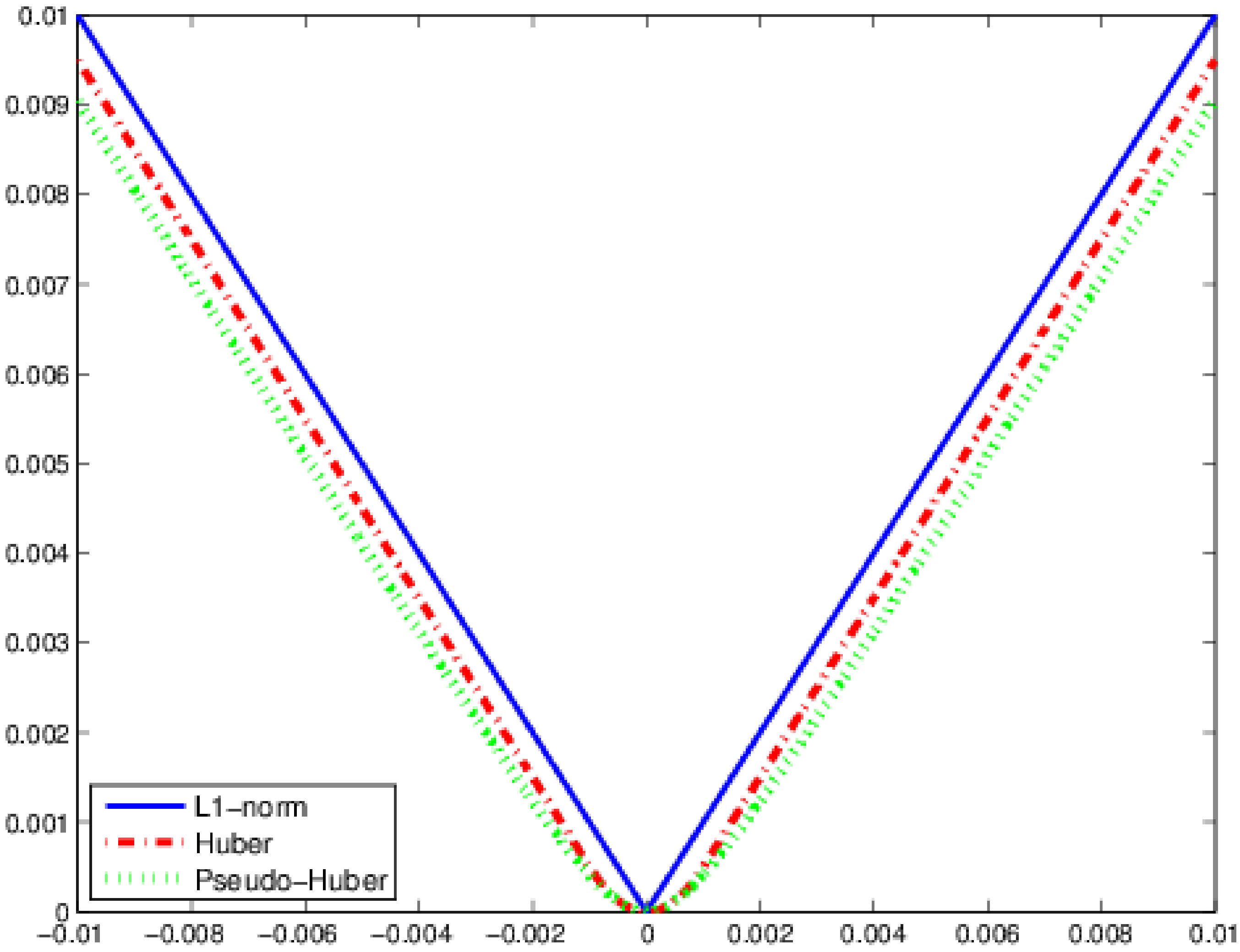
where $\mu > 0$. Only first-order differentiable.

Pseudo Huber approximation: replaces $\|x\|_1$ with

$$\psi_\mu(x) = \mu \sum_{i=1}^n \Big(\sqrt{1 + \frac{x_i^2}{\mu^2}} - 1\Big)$$

Smooth function, has derivatives of any degree.

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

2nd-order method

Use 2nd-order information (Newton direction).

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

Use Inexact Newton Method

Dembo, Eisenstat & Steihaug,
SIAM J. on Num Analysis 19 (1982) 400–408.

Continuation

Embed inexact Newton Meth 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.

Theory ???

Theory for IPM:

G., Matrix-Free Interior Point Method, *Computational Optimization and Applications*, vol. 51 (2012) 457–480.

G., Convergence Analysis of an Inexact Feasible IPM for Convex QP, *Tech Rep ERGO-2012-008*, July 2012.

Theory for Continuation:

Fountoulakis and G.
Second-order Methods for Strongly Convex ℓ_1 Regularization, *Tech Rep ERGO-2013-* (in preparation) 2013.

Three examples of simple ℓ_1 regularization

- Compressed Sensing
  with **K. Fountoulakis and P. Zhlobich**
- Google Problem
  with **K. Woodsend**
- Machine Learning Problems
  with **K. Fountoulakis**

Example One

- **Compressed Sensing**

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

Compressed Sensing

> *Relatively small number of random projections of a sparse signal can contain most of its salient information.*

If a signal is sparse (or approximately sparse) in some orthonormal basis, then an accurate reconstruction can be obtained from random projections of the original signal. A has the form $A = RW$, where

- R is a low-rank randomised sensing matrix
- W is a basis over which the signal has a sparse representation

Candès, Romberg & Tao,
Comm on Pure and Appl Maths 59 (2005) 1207-1233.

Compressed Sensing joint work with

Kimon Fountoulakis and **Pavel 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, G., Zhlobich
Matrix-free IPM for Compressed Sensing Problems,
ERGO Technical Report, 2012.

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

Two-way Orthogonality of A

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

$$AA^T = I_m.$$

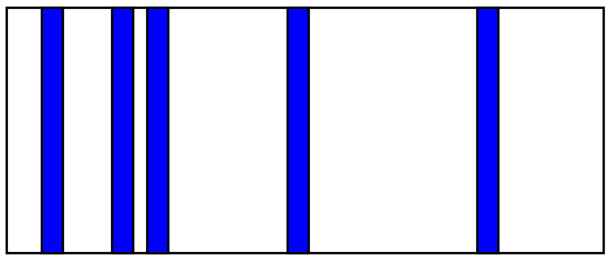
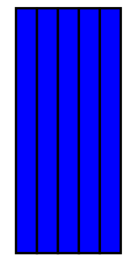
- small subsets of *columns* of A are nearly-orthogonal to each other: *Restricted Isometry Property (RIP)*

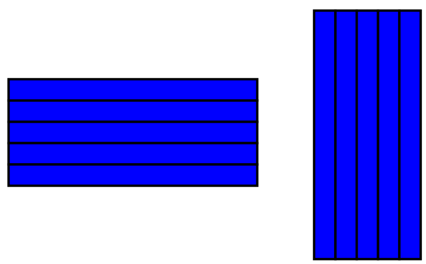
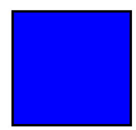
$$\|\bar{A}^T \bar{A} - \frac{m}{n} I_k\| \leq \delta_k \in (0, 1).$$

Candès, Romberg & Tao,
Comm on Pure and Appl Maths 59 (2005) 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 $A \in \mathcal{R}^{m \times n}$.

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

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

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

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

Preconditioner

Approximate

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

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

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

Theorem

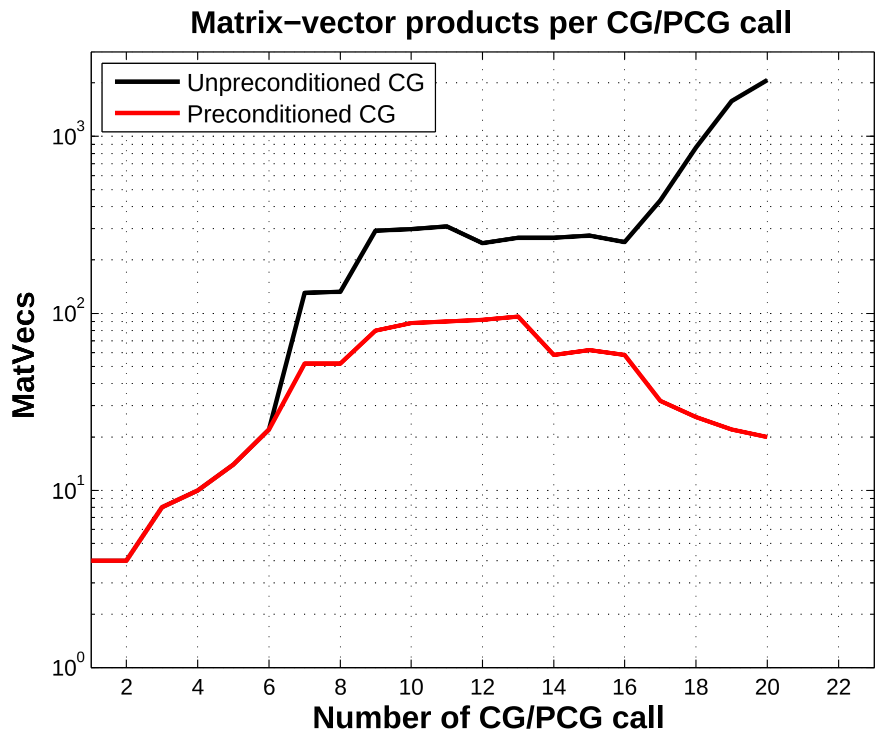
- Exactly n eigenvalues of $\mathcal{P}^{-1}\mathcal{M}$ are 1.
- The remaining n eigenvalues satisfy

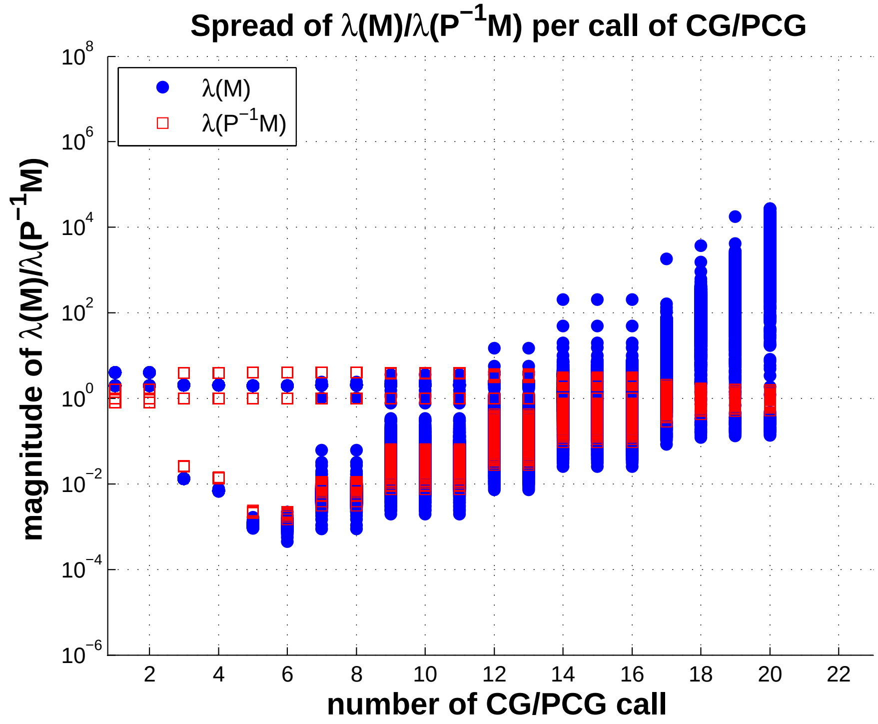
$$|\lambda(\mathcal{P}^{-1}\mathcal{M}) - 1| \leq \delta_k + \frac{n}{m\delta_k L},$$

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

Fountoulakis, G., Zhlobich
Matrix-free IPM for Compressed Sensing Problems,
ERGO Technical Report, 2012.

Preconditioning





$\longrightarrow$ good clustering of eigenvalues

Computational Results: Comparing MatVecs

Prob size	k	**NestA**	**mf-IPM**
4k	51	424	301
16k	204	461	307
64k	816	453	407
256k	3264	589	537
1M	13056	576	613

NestA, Nesterov's smoothing gradient
Becker, Bobin and Candés,
`http://www-stat.stanford.edu/~candes/nesta/`

mf-IPM, Matrix-free IPM
Fountoulakis, G. and Zhlobich,
`http://www.maths.ed.ac.uk/ERGO/`

SPARCO problems

Comparison on 18 out of 26 classes of problems (all but 6 complex and 2 installation-dependent ones).

Solvers compared:

PDCO, *Saunders and Kim*, Stanford,
$\boldsymbol{\ell_1 - \ell_s}$, *Kim, Koh, Lustig, Boyd, Gorinevsky*, Stanford,
FPC-AS-CG, *Wen, Yin, Goldfarb, Zhang*, Rice,
SPGL1, *Van Den Berg, Friedlander*, Vancouver, and
mf-IPM, *Fountoulakis, G., Zhlobich*, Edinburgh.

On 36 runs (noisy and noiseless problems), **mf-IPM**:

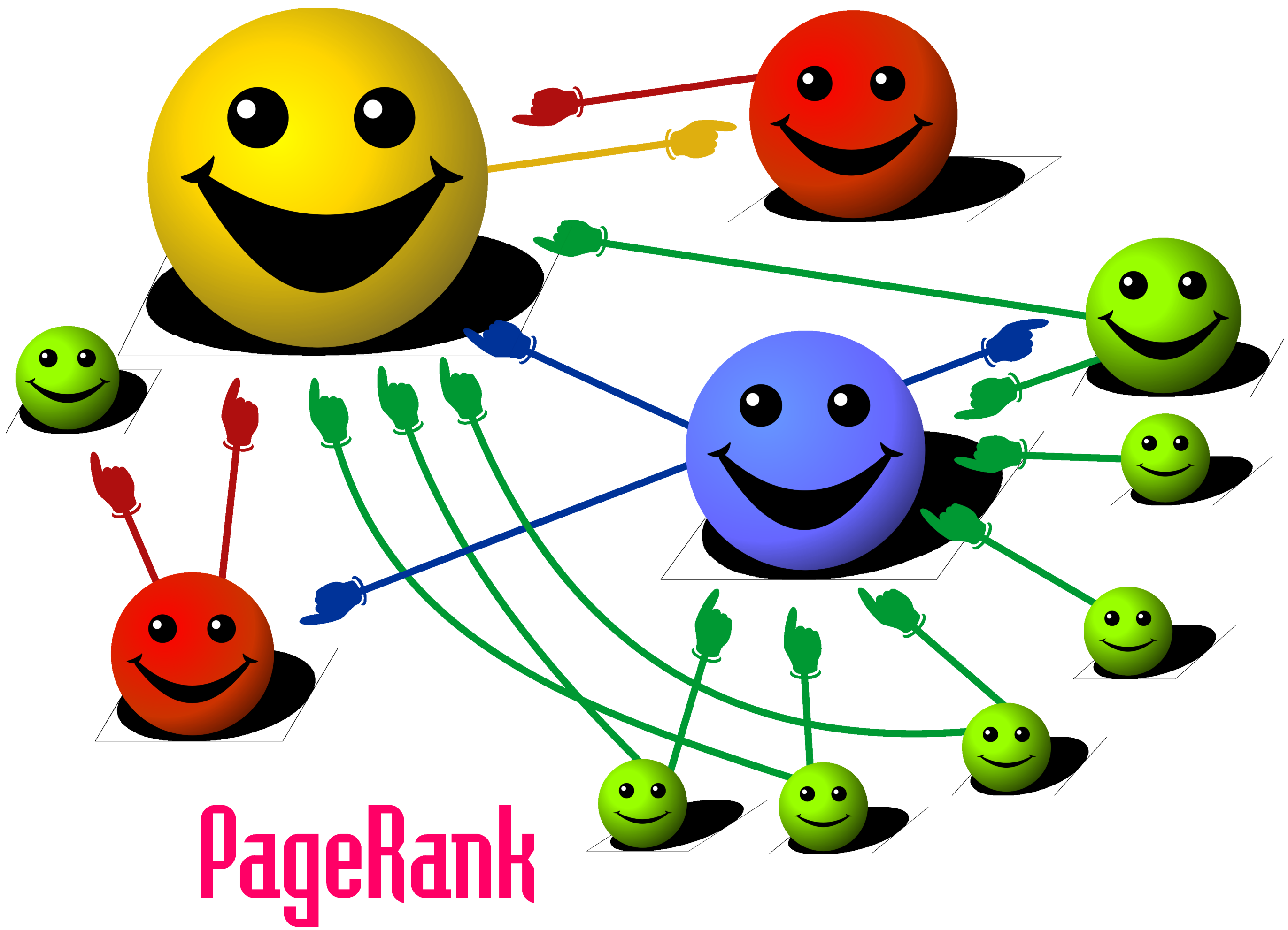
- is the fastest on 11,
- is the second best on 14, and
- overall is very robust.

Example Two

- **Google Problem**

  with **K. Woodsend**

Ranking of nodes in networks



Google Problem joint work with
Kristian Woodsend

An adjacency matrix $G \in \mathcal{R}^{n \times n}$ of web-page links is given (web-pages are the nodes). G is *column-stochastic*.

Teleportation:

$$M = \lambda G + (1-\lambda)\frac{1}{n}ee^T,$$

with $\lambda \in (0,1)$, usually $\lambda = 0.85$.

Find the *dominant right eigenvector* x of M with eigenvalue equal to 1

$$Mx = x, \quad \text{such that} \quad e^T x = 1, \; x \geq 0.$$

and use x as a **ranking vector**.

Google Problem

$$\begin{array}{ll} \min & \frac{1}{2}\,\|Mx - x\|_2^2 \\ \text{s.t.} & e^T x = 1,\ x \geq 0 \end{array}$$

Rearrange:

$$\|Mx - x\|_2^2 = x^T(M - I)^T(M - I)x$$

to produce a standard QP formulation with

$$Q = (M - I)^T(M - I).$$

A very easy QP problem!

Preconditioner for Google Problem

Approximate

$$\mathcal{M} = \begin{bmatrix} Q + \Theta^{-1} & e \\ e^T & 0 \end{bmatrix}$$

with

$$\mathcal{P} = \begin{bmatrix} D_Q & e \\ e^T & 0 \end{bmatrix},$$

where $D_Q = diag\{Q + \Theta^{-1}\}$.

G., Woodsend
Matrix-free IPM for Google Problems,
ERGO Technical Report (in preparation) 2013.

Computational Results: mf-IPM

	Size	degree	IPM-iters	**MatVecs**
$\lambda = 0.85$	16k	20	5	8
	64k	20	4	5
	256k	20	3	4
	1M	20	3	11
$\lambda = 1.0$	16k	20	5	8
	64k	20	4	5
	256k	20	3	6
	1M	20	3	14

mf-IPM faster than Nesterov's coordinate descent.
Nesterov (SIOPT 2012) solves them in 45-70 **MatVecs**.

Real-life Networks

Stanford Large Network Dataset Collection

`http://snap.stanford.edu/data/`

Data set	Nodes	Edges	Conjugate Grad Iters IPM	Conjugate Grad Iters CG	Conjugate Grad time t(s)	Precond. CG Iters IPM	Precond. CG Iters PCG	Precond. CG time t(s)
p2p-Gnutella04	10879	50873	13	233	2.7	13	151	2.3
p2p-Gnutella24	26518	91887	14	214	7.3	15	161	6.3
p2p-Gnutella25	22687	77392	13	216	5.4	13	143	4.6
p2p-Gnutella30	36682	125010	13	196	8.4	13	123	7.2
p2p-Gnutella31	62586	210478	14	205	15.6	14	140	13.6
soc-Epinions1	75888	584725	28	588	31.9	35	459	48.6
amazon0601	403394	3790782	16	191	76.1	18	72	49.3
web-Google	916428	6021467	15	193	197.1	13	47	85.6
wiki-Talk	2394385	7415795	15	110	256.8	15	55	198.9
web-BerkStan	685231	8285826	12	204	106.7	12	52	52.0

The number of CG (PCG) iterations is equal to the number of **matrix-vector** products.

Example Three

- **Machine Learning Problems**

  with **K. Fountoulakis**

Machine Learning Problems joint work with **Kimon Fountoulakis**.

Nesterov, *Math Prog*, 103 (2005) 127-152.

Nesterov, Gradient methods for minimizing composite objective function. *CORE Discussion Papers 2007076*, September 2007.

Richtárik and Takáč, Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Math Prog*, 2012.

Richtárik and Takáč, Parallel coordinate descent methods for big data optimization. *Tech Rep ERGO-2012-013*, November 2012.

Huge-Scale LASSO problem

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

where $A \in \mathcal{R}^{m \times n}$ ($m = 2n$: overdetermined system).
Dimensions: $\boldsymbol{m = 4 \times 10^9,\; n = 2 \times 10^9}$.
Very sparse: 20 nonzero entries per column.

- **Parallel CD (Richtárik and Takáč)**
  solves it doing 34-37 scans through the matrix
  35 iterations, CPU time: 10779s;
- **Truncated Newton (Fountoulakis and G.)**
  solves it using 12-13 matrix-vector multiplications
  13 iterations, CPU time: 5079s.

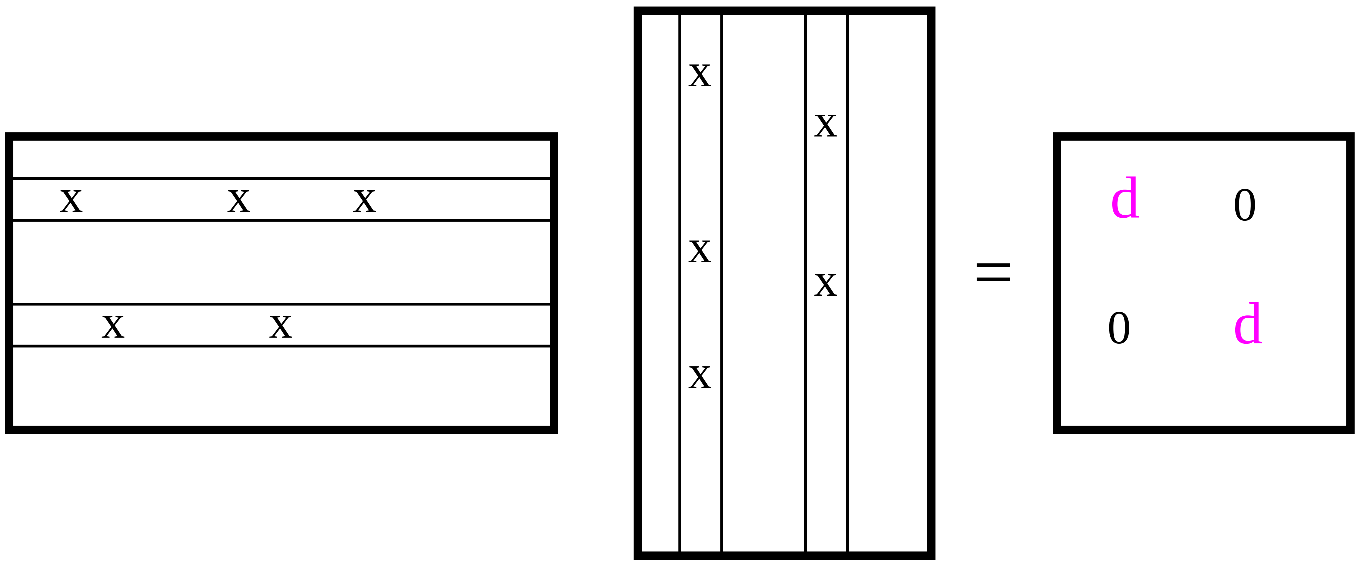
Trivial problem

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

where $A \in \mathcal{R}^{m\times n}$. Highy overdetermined system: $m = 2n$.

Strongly diagonally dominant matrix A^TA.

$A^TA =$



More Machine Learning Problems

			CPU time	
Problem	Features	Training size	**CD**	**TN**
gisette_scale	5,000	6,000	11.65	8.63
real_sim	20,958	72,309	1.85	0.78
epsilon	2,000	400,000	1,658	314
rcv1_train	47,236	20,242	0.77	0.57
news20_binary	1,355,191	19,996	3.30	9.57

CD Coordinate Descent, **Chih-Jen Lin**, `Liblinear`:
`http://www.csie.ntu.edu.tw/~cjlin/liblinear/`

TN Truncated Newton Meth, **Fountoulakis and G.**

What is going on? Linear Algebra Viewpoint

$$\min_x \ \tau\|x\|_1 + \underbrace{\frac{1}{2}\|Ax-b\|_2^2}_{\phi(x)},$$

Quadratic Opt. with $Q = A^T A$. For overdetermined systems ($m > n$), Q is likely to be very well conditioned.

Small exercise:
Ignore ℓ_1 term and compute:
$\nabla\phi(x) = A^T(Ax-b)$ and $\nabla^2\phi(x) = A^T A$
$d_{SD} = -\nabla\phi(x)$ and $d_N = -(\nabla^2\phi(x))^{-1}\nabla\phi(x)$

If $\boldsymbol{\nabla^2\phi(x) \approx I}$ then $\boldsymbol{d_{SD} \approx d_N}$.

Conclusions

The **2nd-order information** can (sometimes should) be used also in trivial optimization.

Achievable by using **inexact Newton directions** in:

- IPMs
- continuation approach

Final Comments

- **large/huge** does not always mean **difficult**
- Many **Big Data** problems are trivial!