Interior solution of large-scale entropy maximization problems

ISMP 2003

Copenhagen, Denmark, August 2003

Michael Saunders and John Tomlin

Stanford University

IBM Almaden Research Center

saunders@stanford.edu, tomlin@almaden.ibm.com

Motivation

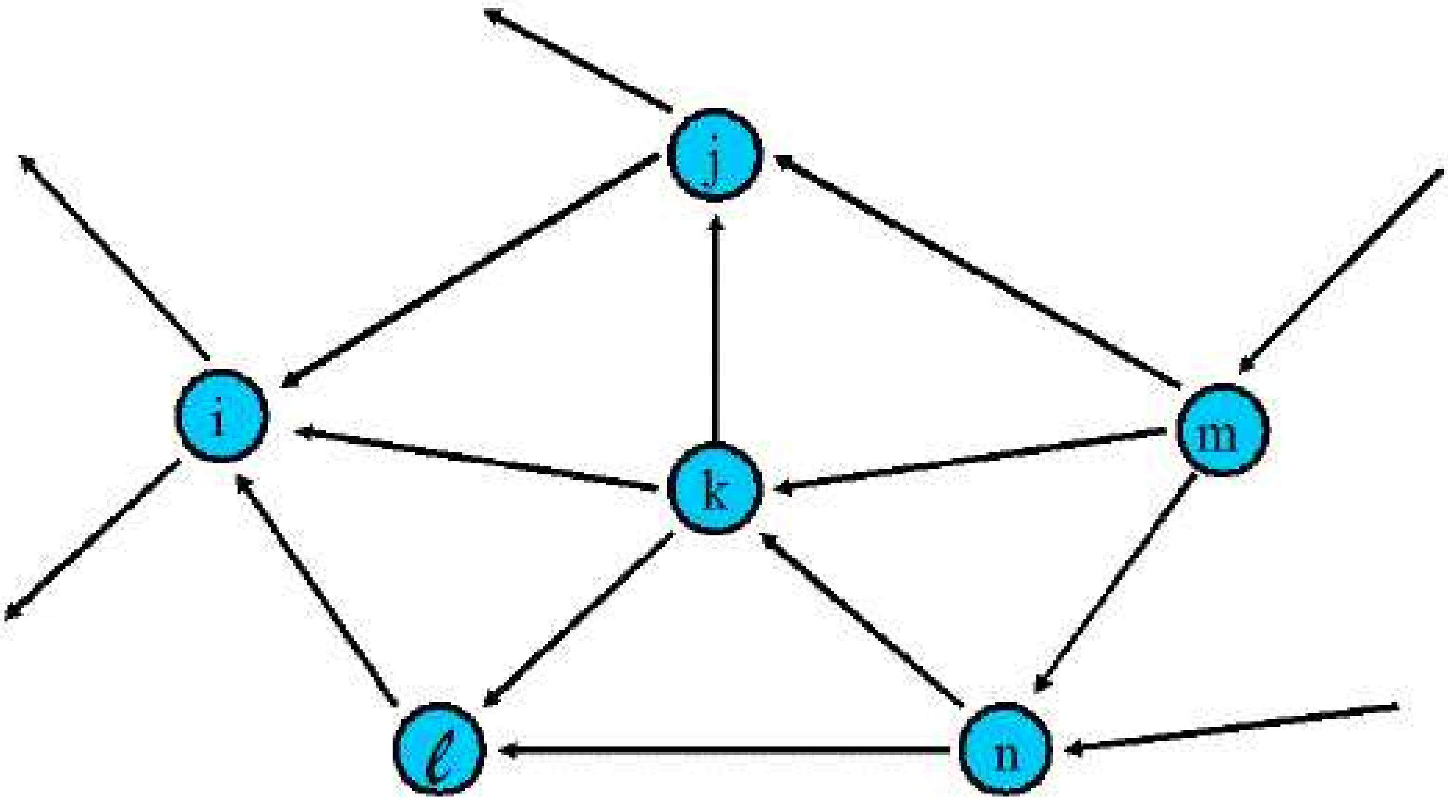
Solve large-scale maximum entropy models

$$\begin{array}{ll} \underset{x}{\text{maximize}} & S = -\sum x_j \ln x_j \\ \text{subject to} & Ax = b, \quad x > 0 \end{array}$$

Applications

- Transportation planning
- Probabilistic Query Models for transaction data
- Natural Language Processing, Knowledge Management, etc.
- Model “traffic” on the World Wide Web
  Main thrust of this talk

Network Graph



Graph model of the web:

$$G = (V,E)$$

Where V is set of vertices $(i,j,k,\ldots..)$ or nodes or pages
And E is the set of edges (i,j)

The Random Web Surfer

- Assume some notional "clock"
- At each clock tick the web surfer follows an out-link with some probability
- Model 1 Assume the probabilities are fixed
  e.g., probability of following an out-link from page i is $1/d_i$ where d_i is the out-degree of node i
- Model 2 Let probabilities be variables to be determined

1: Markov Chain Model

- Assume transition probabilities p_{ij} from page i to page j are fixed at $1/d_i$ (constant for page i)
- Let $P = (\, p_{ij} \,)$. The stationary state of the Markov Chain defined by P is then x, where

$$x = P^T x$$

  (dominant left eigenvector of the stochastic matrix P)
- The value x_i is the ideal PageRank of page i
  (as used in Google, with additional features, to assign importance to a web page)

2: Network Flow Model

An alternative (richer) class of models

Let variable y_{ij} be the number of surfers clicking on link (i, j) at each clock tick. Then

$$H_j = \sum_{i|(i,j)\in E} y_{ij}$$

is the number of hits per unit time at node j.

Conservation

$$\sum_{j|(i,j)\in E} y_{ij} - \sum_{j|(j,i)\in E} y_{ji} = 0 \quad (i = 1, \ldots, n)$$

$$Y = \sum_{i,j} y_{ij} = \sum_j H_j \quad (\text{total flow})$$

Probabilistic Network Model

- Usually we prefer to work with normalized values (probabilities) $p_{ij} = y_{ij}/Y$. Constraints become

$$\sum_{j|(i,j)\in E} p_{ij} - \sum_{j|(j,i)\in E} p_{ji} = 0 \quad (i = 1, \ldots, n)$$

$$\sum_{i,j} p_{ij} = 1$$

- The PageRank model specifies a particular solution:

$$p_{ij} = \frac{H_i}{Y d_i} \quad \forall (i,j) \in E$$

They satisfy the conservation equations, but ... only one of infinitely many solutions

Entropy Objective

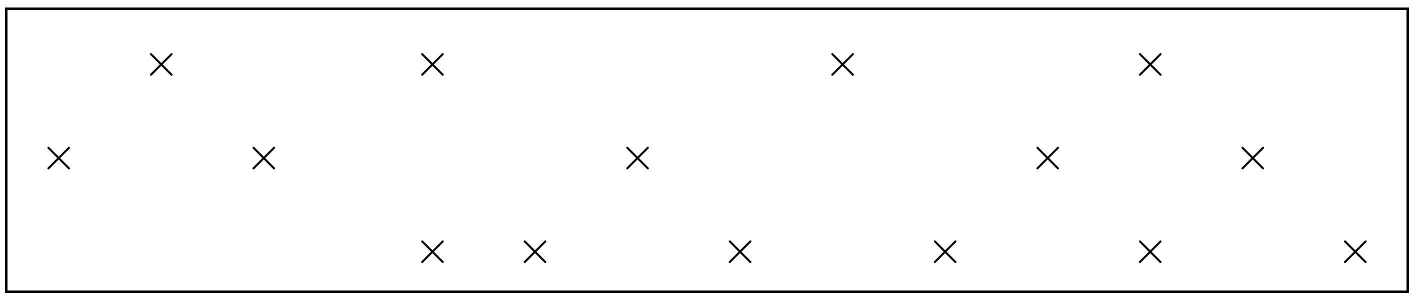
- Principle:
  In the absence of complete information about a probability distribution, choose the one that maximizes uncertainty, subject to whatever is known (Jaynes).
  This unique distribution is the (Shannon) entropy function.
- Hence we should

$$\text{maximize} \quad S = -\sum_{i,j} p_{ij} \log p_{ij}$$

subject to the constraints

The Entropy Problem

$$\begin{array}{lll} \underset{x}{\text{minimize}} & \varphi(x) = \sum x_j \ln x_j & \\ \text{subject to} & Ax = b, & x > 0. \end{array}$$

$A =$ 

$$\min \sum x_j \ln x_j \quad \text{s.t.} \quad Ax = b$$

- Optimality conditions:

$$\begin{aligned} A^T y &= g(x) \qquad\qquad \leftarrow g_j = 1 + \ln x_j \\ Ax &= b \end{aligned}$$

- Newton:

$$\begin{pmatrix} -X^{-1} & A^T \\ A & \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} g - A^T y \\ b - Ax \end{pmatrix}$$

- $X = \mathrm{diag}(x)$ (keep $x > 0$). Plausible method?
- Dual method: Erlander (1977), Eriksson (1981)

Regularized Entropy Problem

$$\begin{array}{ll} \underset{x,\,r}{\text{minimize}} & \sum x_j \ln x_j + \frac{1}{2}\|r\|^2 \\ \text{subject to} & Ax + \delta r = b, \qquad x > 0. \end{array}$$

- Ideal for MATLAB primal-dual interior solver pdco.m
- $\delta \approx 10^{-3}$ for "equalities" (optimal $r = \delta y \;\Rightarrow\; Ax + \delta^2 y = b$)
- $\delta > 0$ allows use of LSQR for Δy (CG solver, inexact Newton)

pdco.m

$$\begin{array}{ll} \underset{x,\,r}{\text{minimize}} & \varphi(x) + \tfrac{1}{2}\|\gamma x\|^2 + \tfrac{1}{2}\|r\|^2 \\ \text{subject to} & Ax + \delta r = b, \qquad \ell < x < u \end{array}$$

- MATLAB primal-dual solver for convex, separable $\varphi(x)$ like $c^T x$, $\|x\|_1$, $\sum x_j \ln x_j$
- $\left\{ \begin{array}{ll} \gamma \approx 10^{-3} & \text{Tikhonov reg'n} \\ \delta \approx 10^{-3} & \text{``equalities''} \\ \delta = 1 & \text{least squares} \end{array} \right\}$ x and y bounded, unique
- May have $\frac{1}{2}\|D_1 x\|^2$ and $Ax + D_2 r = b$ (diagonal D_1, $D_2 \succ 0$)
- Basis Pursuit signal analysis, NNLS image restoration
- New C++ implementations of pdco and LSQR

Newton vs. Primal-Dual Interior

- Newton:

$$\begin{pmatrix} -X^{-1} & A^T \\ A & \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} g - A^T y \\ b - Ax \end{pmatrix}$$

- pdco:

$$\begin{pmatrix} -\bar{H} & A^T \\ A & \delta^2 I \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} g - A^T y - \mu X^{-1} e \\ b - Ax \end{pmatrix}$$

$$\bar{H} = X^{-1} + X^{-1} Z$$

$\delta > 0$ allows LSQR to solve for Δy

Least-Squares Problem for Δy

$$\min_{\Delta y} \left\| \begin{pmatrix} DA^T \\ \delta I \end{pmatrix} \Delta y - \begin{pmatrix} Dw \\ r_1/\delta \end{pmatrix} \right\|_2$$

$$\begin{aligned} D &= (X^{-1} + X^{-1}Z)^{-1/2} \\ r_1 &= b - Ax - \delta^2 y \end{aligned}$$

- Set `atol` $= 0.001$ initially for LSQR
- Solve inexactly for Δy
- Get corresponding Δx and Δz (exactly)
- Reduce `atol` by 0.1 if necessary

pdco input parameters

```
m          =    51152    n          =   662463     nnz(A)   =  1987389
max |b |   =        1    max |x0|   =  1.5e-06     xsize    =  7.5e-05
max |y0|   =        0    max |z0|   =  1.0e-05     zsize    =  1.0e+00

x0min      =     0.01    featol     =  1.0e-05     d1max    =  0.0e+00
z0min      =    1e-05    opttol     =  1.0e-05     d2max    =  1.0e-03
mu0        =  1.0e-05    steptol    =     0.99    bigcenter=     1000

LSQR:
atol1      =  1.0e-03    atol2      =  1.0e-06
```

pdco iteration log

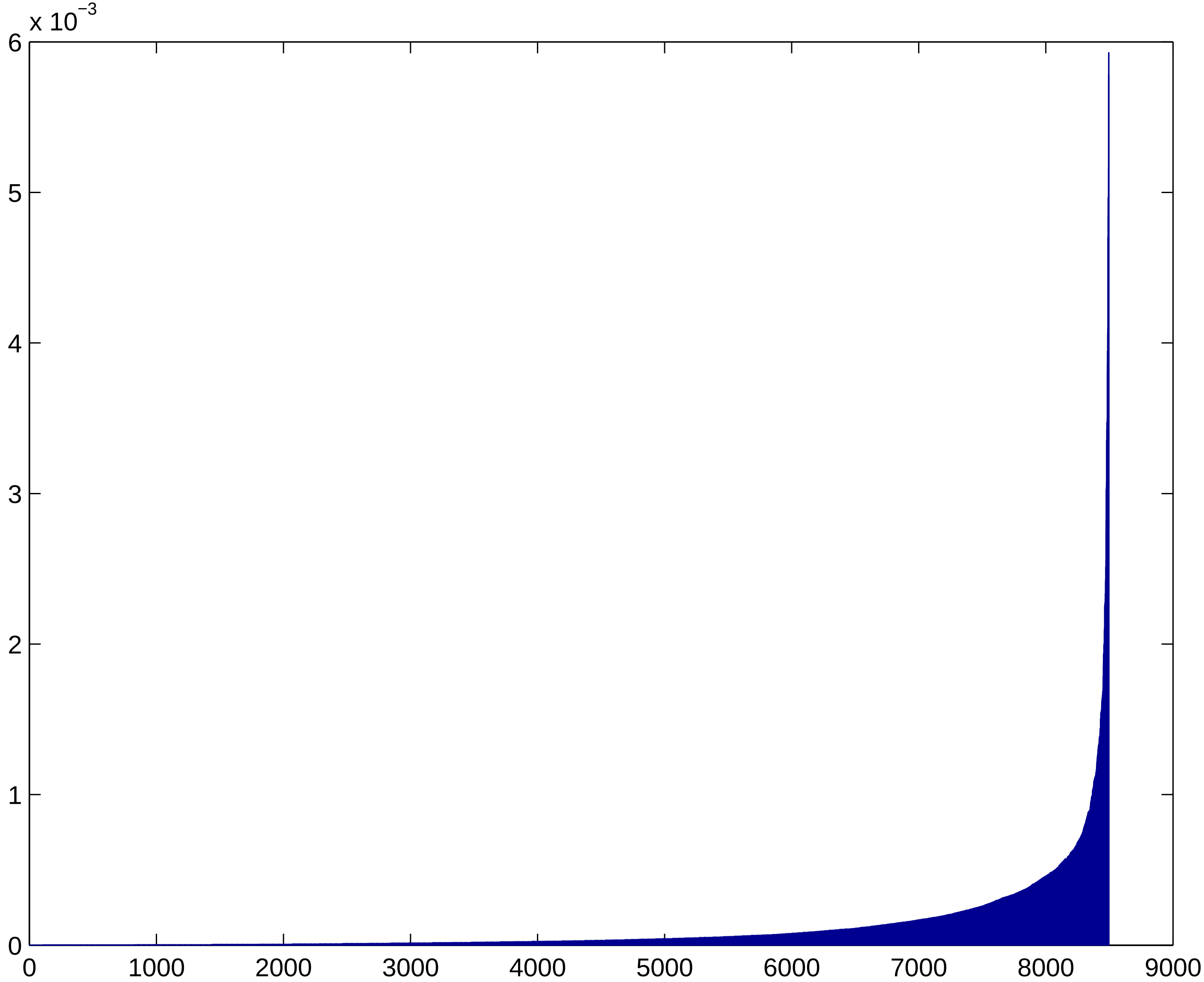
Itn	mu	step	Pinf	Dinf	Cinf	Objective	nf	center	atol	LSQR	Inexact
0			2.5	1.1	-6.7	-1.3403720e+01		1.0			
1	-5.0	0.267	2.4	1.1	-5.1	-1.3321172e+01	1	242.0	-3.0	5	0.001
2	-5.1	0.195	2.3	1.0	-5.3	-1.3220658e+01	1	36.9	-3.0	5	0.001
3	-5.2	0.431	2.1	0.9	-5.2	-1.2942743e+01	1	122.9	-3.0	5	0.001
4	-5.5	0.466	1.9	0.7	-5.3	-1.2711643e+01	1	41.8	-3.0	6	0.001
5	-5.7	0.671	1.4	0.2	-5.5	-1.2492935e+01	1	71.8	-3.0	9	0.001
6	-6.0	1.000	-0.0	-0.8	-5.8	-1.2367004e+01	1	2.7	-3.0	10	0.001
7	-6.0	1.000	-0.1	-2.3	-6.0	-1.2368200e+01	1	1.1	-3.0	9	0.002
8	-6.0	1.000	-1.1	-4.7	-6.0	-1.2367636e+01	1	1.0	-3.0	2	0.009
9	-6.0	1.000	-1.3	-5.7	-6.0	-1.2367655e+01	1	1.0	-3.0	7	0.015
10	-6.0	1.000	-2.5	-7.6	-6.0	-1.2367607e+01	1	1.0	-3.0	2	0.004
11	-6.0	1.000	-3.7	-8.6	-6.0	-1.2367609e+01	1	1.0	-3.5	8	0.004
12	-6.0	1.000	-5.9	-11.0	-6.0	-1.2367609e+01	1	1.0	-4.7	11	0.000

```
PDitns   =        12   LSQRitns =        79     time     =      134.1    (MATLAB)
                                                                  22.4    (C++)
```

Observations

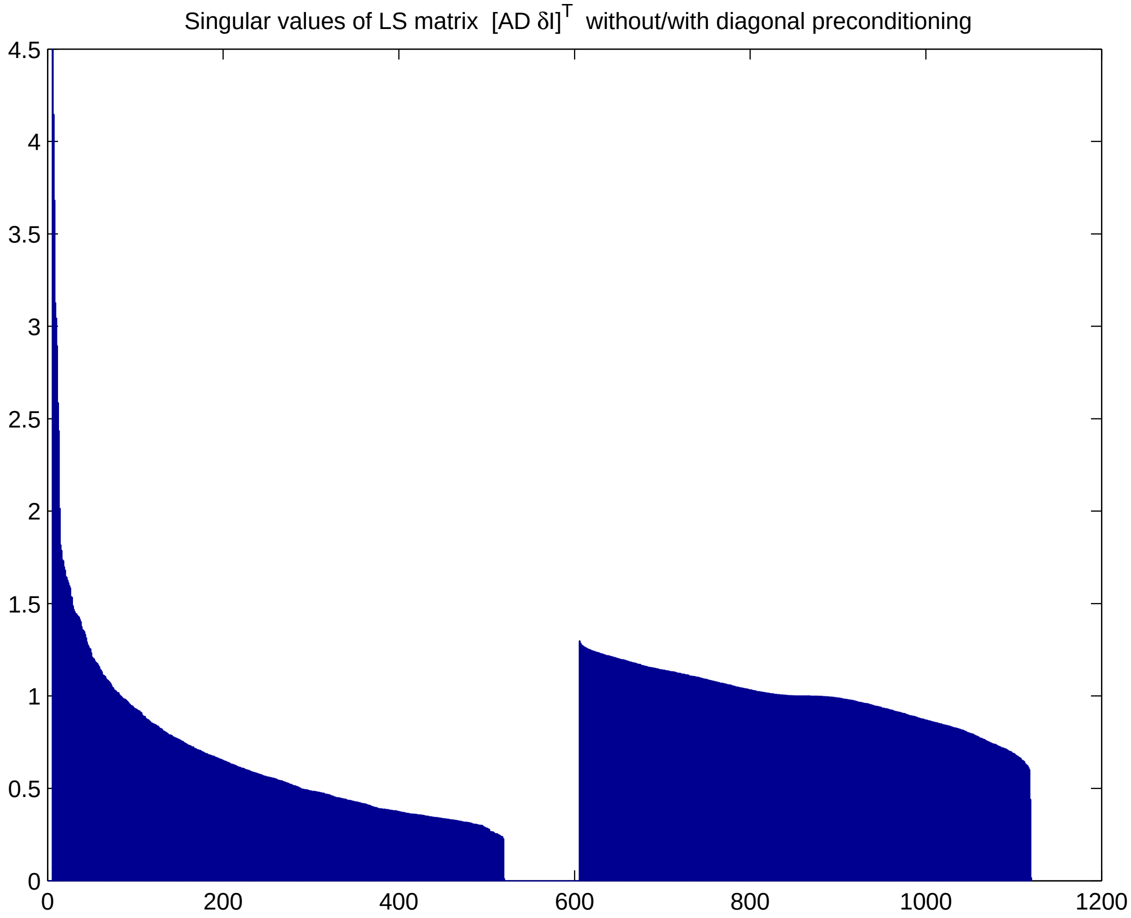
- pdco needs very few primal-dual iterations even with inexact search directions
- LSQR needs very few iterations to compute the inexact directions even near solution
- Entropy models are exceptionally friendly for interior methods

Distribution of x^*



Singular values at x^*

$$\begin{pmatrix} DA^T \\ \delta I \end{pmatrix} \qquad\qquad \begin{pmatrix} DA^T \\ \delta I \end{pmatrix} + \text{diag preconditioning}$$



Conclusions

- Network formulation more general than PageRank
- Primal-dual interior method effective for large entropy problems
- Further understanding needed
  but singular values tell a story
- Current implementations: O(1 million) variables
- For millions of nodes, need
  64-bit machines for in-core implementation
  distributed computation