Solution analysis and continuation algorithms for multilinear pagerank

Federico Poloni
Joint work with B. Meini

University of Pisa

GAMM 2018, Munich

Pagerank extensions

Pagerank [Page '98]

Input: transition probabilities $P_{ij} = P[i \to j]$, 'personalization vector' v.
Output: 'importance' score x_i of each node

$$x = \alpha P x + (1-\alpha) v, \qquad \mathbf{1}^\top x = 1.$$

Multilinear Pagerank [Gleich-Lim-Yu '15]

Input: two-step transition probabilities $P_{ijk} = P[i \to j \to k]$, 'personalization vector' v.
Output: 'importance' score x_i of each node

$$x = \alpha \sum_{i,j=1}^{n} P_{ij:} x_i x_j + (1-\alpha) v, \qquad \mathbf{1}^\top x = 1.$$

Not just a second-order Markov chain — that would rank edges, X_{ij}.

Ways to look at it

Viewpoint 1: tensor eigenvalues.

$$x = \alpha \sum_{i,j=1}^{n} P_{ij:} x_i x_j + (1-\alpha) v x_i x_j, \qquad \mathbf{1}^\top x = 1.$$

Since $\mathbf{1}^\top x = 1$, we can homogenize. Z-eigenvector problem.
[e.g., Qi, Chen, Chen book '18]

Viewpoint 2: branching processes and quadratic vector equations.

$$x = \sum_{i,j=1}^{n} \widehat{P}_{ij:} x_i x_j + \widehat{v}, \qquad x \geq 0.$$

[Kolmogorov 1940s, Etessami-Yannakakis '05, Bean-Kontoleon-Taylor '08, P. '13]

We focus on Viewpoint 2 here.

Quadratic vector equations: the theory

$\widehat{P} := \alpha P$, $\widehat{v} := (1-\alpha)v$.

$$x = \sum_{i,j=1}^{n} \widehat{P}_{ij:} x_i x_j + \widehat{v}, \qquad x \geq 0.$$

- Among all nonnegative solutions, there is a minimal one x_* (i.e., $0 \leq x_* \leq x$ for all other solutions x).
- If we start from $x^{(0)} = 0$, most iterations (fixed-point, Newton, ...) converge monotonically to x_*.
- x_* is typically easy to compute, but algorithms are slower when there are other non-minimal solutions very close to it. Double=trouble.

[Hautphenne-Latouche-Remiche '11, Etessami-Stewart-Yannakakis '12, P '13]

Adapting the theory

In MLPR, $\mathbf{1}^\top P = \mathbf{1}^\top$, $\mathbf{1}^\top v = 1$. The iteration

$$x^{(k+1)} = \sum_{i,j=1}^{n} \widehat{P}_{ij:} x_i^{(k)} x_j^{(k)} + \widehat{v}, \quad x^{(0)} = \mathbf{0}$$

has predictable behaviour on the 'total mass': if $\mathbf{1}^\top x^{(k)} = w$, then $\mathbf{1}^\top x^{(k+1)} = \alpha w^2 + (1-\alpha)$.

Theorem

- If $\alpha \leq \frac{1}{2}$, $x^{(k)} \to x_*$, the unique minimal solution with $\mathbf{1}^\top x_* = 1$.
- If $\alpha > \frac{1}{2}$, $x^{(k)} \to x_*$, the unique minimal solution with $\mathbf{1}^\top x_* = \frac{1-\alpha}{\alpha}$. There may be several solutions $x \geq x_*$ with $\mathbf{1}^\top x = 1$.

Uniqueness (or not) of stochastic solutions also in [Gleich-Lim-Yu '15].

The Perron vector trick

Idea

Is it possible to deflate the minimal solution x_*?

Yes! (adapting an idea from [Meini-P '11] for a case with known $x_* = \mathbf{1}$).
Let $x = x_* + y$. Then the equation becomes

$$y = H_y y, \qquad (H_y) = \sum_j \widehat{P}_{:j:} + \widehat{P}_{j::} + \widehat{P}_{j::} y_j$$

$y =$ Perron eigenvector of the nonnegative matrix H_y.

- $y^{(k)} =$ Perron (maximal) eigenvector of $H_{y^{(k-1)}}$.
- Normalize $y^{(k)}$ (s.t. $x + y^{(k)}$ is stochastic).
- Iterate.

Large and small α

- The good When $\alpha \leq \frac{1}{2}$, there is a unique stochastic solution, and many fixed point iterations (e.g. Newton's method, Gauss-Seidel-like variants. . .) converge to it monotonically.
- The bad When $\alpha > \frac{1}{2}$, there may be multiple stochastic solutions, and convergence may be problematic — even if we enforce $\mathbf{1}^\top x^{(k)} = 1$.

Large and small α

- The good When $\alpha \leq \frac{1}{2}$, there is a unique stochastic solution, and many fixed point iterations (e.g. Newton's method, Gauss-Seidel-like variants...) converge to it monotonically.
- The bad When $\alpha > \frac{1}{2}$, there may be multiple stochastic solutions, and convergence may be problematic — even if we enforce $\mathbf{1}^\top x^{(k)} = 1$.
- The ugly Perron-based iterations can be used for the bad case $\alpha > \frac{1}{2}$:
  - Compute minimal (sub-stochastic) solution x^*;
  - Change variable $y = x - x^*$;
  - Interpret resulting equation as $y = H_y y$;
  - Fixed-point iteration: $y^{(k)} =$ Perron eigenvector of $H_{y^{(k-1)}}$ (or variants)

(Another algorithm involving Perron vectors is in [Benson-Gleich-Lim '17].)

Improvements

Improvement 1: use Newton's method on $y - \text{perron}(H_y) = 0$, where $\text{perron}(\cdot)$ is the map that computes the Perron vector of a matrix.

An expression for the Jacobian can be found using eigenvector derivatives.

Theorem [Meini-P '17]

The Jacobian of the map $w = \text{perron}(H_y)$ is

$$J = \alpha(w\mathbf{1}^\top - (I - H_y + w\mathbf{1}^\top)^{-1} \sum_j P_{:j:} w_j).$$

(and [Bini-Meini-P '11] for the branching process problem.)

Improvements

Improvement 2 Use perturbation / continuation techniques: first solve the problem for an 'easy' α, then increase its value.

Theorem [Meini-P '17]

Let x_α be the solution vector for a certain value of $\alpha \in (0,1)$. Then,

$$x_{\alpha+h} = x_\alpha + \left(I - \alpha(\sum_j P_{:j:} + P_{j::}) \right)^{-1} \left(\sum_{i,j} P_{ij:}(x_\alpha)_i (x_\alpha)_j - v \right) h + O(h^2).$$

Step-size heuristic: estimate the neglected second-order term $\frac{\mathrm{d}x_\alpha}{\mathrm{d}\alpha^2}$, and use it to choose the next step size.

Variants

- Second-order Taylor
- Linear extrapolation from previous two values
- "semi-implicit method": replace α in the formula with $\alpha + h$.

Experiments

For now, just small scale examples (but with an eye to scalability).

[Gleich-Lim-Yu '15] contains 29 small-size benchmark problems ($n \in \{3, 4, 6\}$), some of them with difficult convergence.

For $\alpha = 0.99$, the best existing methods (`newton` and `innout`) can solve 23 and 26 of them, respectively.

Goal: a numerical method that scores 29/29 (in a reasonable number of iterations).

Example (`R6_3`, [Gleich-Lim-Yu '15])

$$\frac{1}{4}\left[\begin{array}{cccccc|cccccc|cccccc|cccccc|cccccc|cccccc}
0&0&0&0&0&0&0&0&0&0&0&2&1&0&0&0&0&0&0&0&4&0&4&0&0&2&4&0&0&0&0&0&0&0&0&4\\
0&0&0&0&0&0&0&0&4&0&0&0&0&4&0&0&0&0&0&0&0&0&0&0&0&2&0&0&2&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&4&0&0&0&2&0&4&2&0&0&2&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&4&0&4&0&0&0&0&0&0&0&2&0&4&0&0\\
0&0&0&0&0&0&0&4&0&0&0&2&0&0&0&0&0&0&0&4&0&0&0&0&0&0&0&4&0&0&0&0&4&0&2&0\\
4&4&4&4&4&4&4&0&0&4&4&0&1&0&4&4&4&4&4&0&0&0&0&0&0&0&0&0&0&4&0&0&0&0&0&0
\end{array}\right]$$

Numerical results

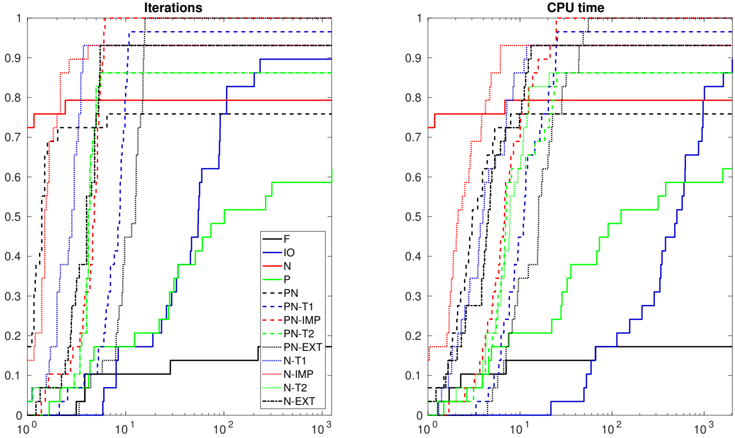


Figure: Performance profiles for the 29 benchmark tensors and $\alpha = 0.99$

Left = faster. Top = more reliable.

Numerical results

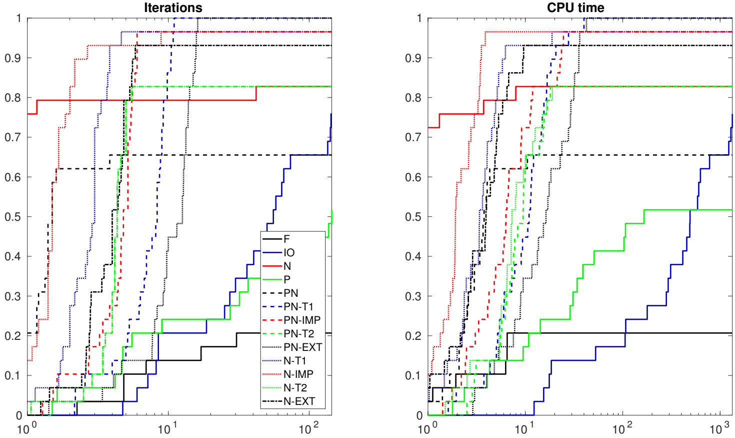


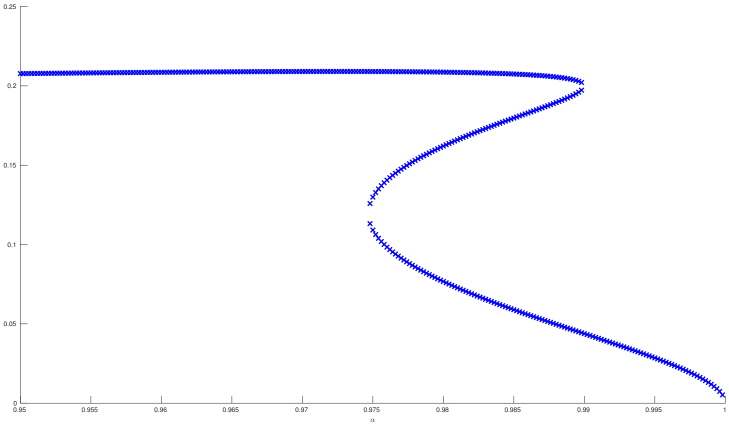
Figure: Performance profiles for the 29 benchmark tensors and $\alpha = 0.999$

Left = faster. Top = more reliable.

A difficult experiment

F	IO	N	O	ON	ON-T1	ON-IMP	ON-T2	ON-EXT	N-T1	N-IMP	N-T2	N-EX
NaN	42	5	21	6	35	22	22	46	10	6	22	13
23	48	6	26	7	50	36	24	80	17	13	24	29
23	48	6	10	7	50	36	24	80	17	13	24	29
114	24	4	19	6	28	22	22	39	8	6	22	11
22	41	17	15	7	74	26	24	110	23	29	24	37
NaN	639	7	NaN	NaN	73	39	25	109	26	14	25	38
NaN	234	5	369	8	44	24	24	63	13	7	24	20
NaN	1,221	6	1,845	9	46	26	26	69	16	9	26	24
NaN	532	5	NaN	7	44	24	25	63	15	7	25	20
NaN	339	6	NaN	7	59	30	25	85	20	9	25	29
NaN	228	5	NaN	7	45	23	25	64	15	7	25	20
NaN	357	6	362	NaN	62	31	26	90	21	9	26	30
NaN	156	6	7,509	6	52	31	25	77	18	10	25	27
NaN	329	6	NaN	NaN	65	32	25	92	22	11	25	33
NaN	550	6	NaN	8	38	23	25	56	13	7	25	19
NaN	NaN	5	NaN	7	31	23	23	47	10	7	23	14
NaN	1,173	NaN	NaN	71	99	32	NaN	87	NaN	11	NaN	NaN
NaN	541	6	1,595	NaN	60	31	25	89	21	10	25	32
NaN	273	NaN	248	8	53	37	NaN	78	17	12	NaN	27
NaN	230	7	196	7	37	24	24	56	12	7	24	17
NaN	533	6	NaN	9	59	31	26	90	19	10	26	29
NaN	NaN	7	205	6	31	32	24	50	10	16	24	17
NaN	435	NaN	504	NaN	59	35	NaN	89	23	NaN	NaN	31
NaN	147	5	137	7	46	29	25	64	15	8	25	20
NaN	835	NaN	NaN	44	76	59	41	121	40	36	41	44
NaN	654	NaN	NaN	NaN	72	33	NaN	103	24	12	NaN	37
NaN	NaN	NaN	8,025	1,326	NaN	903	652	2,878	NaN	NaN	652	NaN
NaN	334	6	307	NaN	57	37	24	91	21	13	24	33
2,005	2,100	9	909	10	34	33	27	52	14	14	27	20

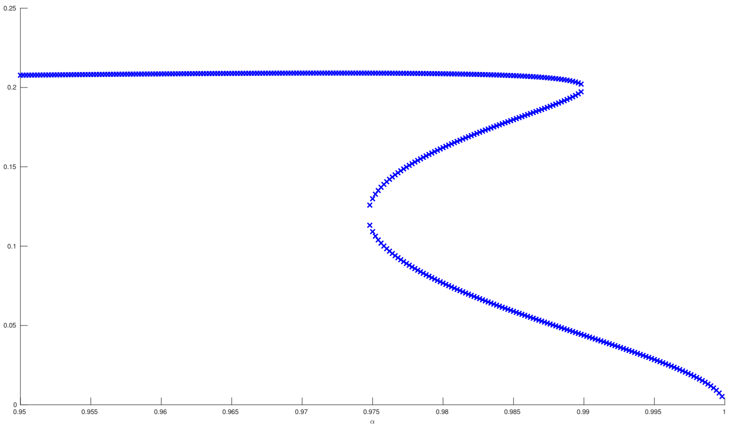
R6_3: solution vs. α



First entry x_1 of the solution(s) vs. value of parameter α.

Solutions for smaller values of α are a poor initial point for $\alpha = 0.99$.

R6_3: a possible fix



Fix: compute solution for $\alpha = 0.9999$ first, then use it as starting point to 'go back' to $\alpha = 0.99$.

17+8 iterations vs. 1326 iterations.

Conclusions

- New numerical strategies: Perron-based methods, continuation.
- Can handle all the benchmark problems successfully.
- Insight: consider all solutions (even if we're looking for only one).
- Coming next: tests at larger scale.

Conclusions

- New numerical strategies: Perron-based methods, continuation.
- Can handle all the benchmark problems successfully.
- Insight: consider all solutions (even if we're looking for only one).
- Coming next: tests at larger scale.

Thanks for your attention!