Trilogy on Computing Maximal Eigenpair

Mu-Fa Chen

(Beijing Normal University)

School of Math, Sichuan Univ

2017 年 7 月 6 日 [redacted]

0 Introduction

Theorem (O. Perron, 1907; G.Frobenius 1912)

For $A \geqslant 0$, irreducible, "Trace $(A) > 0$", $\exists!$ maximal eigenvalue $\rho(A) > 0$ with unique left-eigenvector $u > 0$ and right-eigenvector $g > 0$, up to constant.

$$\boxed{uA = \lambda u, \quad Ag = \lambda g, \quad \lambda = \rho(A)}$$

Example (L.K. Hua, 1984)

$$A = \frac{1}{100}\begin{pmatrix} 25 & 14 \\ 40 & 12 \end{pmatrix},$$

$$\rho(A) = \left(37 + \sqrt{2409}\,\right)/200.$$

$$u = \left(5\left(13+\sqrt{2409}\,\right)/7,\ 20\right),$$

$$\boxed{\approx 44.34397483}$$

$$g = \left(\left(13+\sqrt{2409}\,\right)/4,\ 20\right)^*$$

Let $A \geqslant 0$ be irreducible and invertible.

Input-output method:

$$x_n = x_0 A^{-n}, \qquad n \geqslant 1.$$

$x_n = \left(x_n^{(0)}, \cdots, x_n^{(d)}\right)$: products in nth year

Theorem (Hua's Fundamental Theorem, 1984)

- The optimal choice is $x_0 = u$, it has the fastest grow: $x_n = x_0 \rho(A)^{-n}$.
- Unless $A^{-1} \geqslant 0$, if $x_0 \neq u$, then collapse: $\exists n, j$ such that $x_n^{(j)} \leqslant 0$.

Importance of $u \approx (44.34397483, 20)$

Table Input and collapse time

x_0 [decimals]	Collapse time n
$(44, 20)$	3
$(44.344, 20)$	8
$(44.34397483, 20)$	13

Importance of $(\rho(A), u)$:
Need high precision, large system

Remarks

1) Need study right-eigenvector only:

$$uA = \lambda u \iff A^* u^* = \lambda u^*,$$

2) Diagonals are free:

$$(A+mI)g = \lambda g \iff Ag = (\lambda - m)g$$

The eigenvector is kept, eigenvalues are shifted

I Tridiagonal case. Example $Q=$

$$\begin{pmatrix} -1 & 1^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1^2 & -5 & 2^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2^2 & -13 & 3^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3^2 & -25 & 4^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4^2 & -41 & 5^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5^2 & -61 & 6^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 6^2 & -85 & 7^2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 7^2 & -113 \end{pmatrix}$$

$\rho(Q) \approx -0.525268$, $\boxed{\rho(\text{Infinite } Q) = -1/4}$

Power iteration/method, 1929

Given $v_0 \in \mathbb{R}^{N+1}$, $v_0 \not\perp g$ with $\|v_0\| = 1$, define

$$v_k = \frac{Av_{k-1}}{\|Av_{k-1}\|}, \quad z_k = \|Av_k\|, \quad k \geqslant 1,$$

Then $v_k \to g$ and $z_k \to \rho(A)$ as $k \to \infty$.

$$v_k = \frac{A^k v_0}{\|A^k v_0\|} \qquad \text{“Power”}$$

Power iteration/method, 1929

$$g \approx (55.878, 26.5271, 15.7059, 9.97983, 6.43129, 4.0251, 2.2954, 1)^*$$

$$\tilde{v}_0 = (1,\ 0.587624,\ 0.426178,\ 0.329975,\ 0.260701,\ 0.204394,\ 0.153593,\ 0.101142)^*,$$

$$v_0 = \frac{\tilde{v}_0}{\|\tilde{v}_0\|}, \qquad \|v\| = \sum_k |v_k|,\ \ell^1\text{-norm}$$

Power iteration/method, 1929

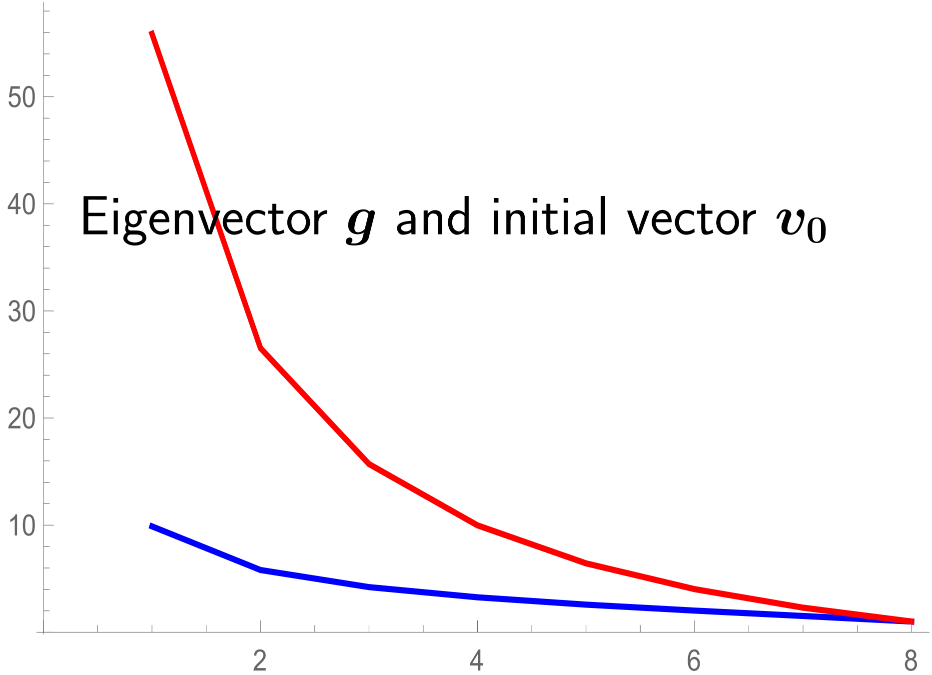
$$g \approx (55.878, 26.5271, 15.7059, 9.97983, 6.43129, 4.0251, 2.2954, 1)^*$$

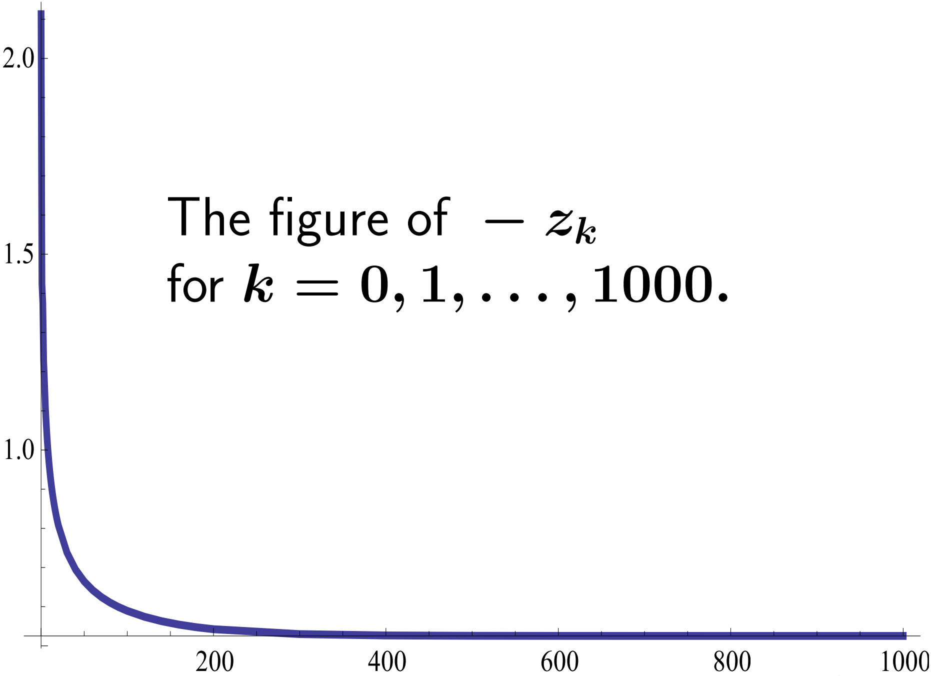
$$\tilde{v}_0 = (1,\ 0.587624,\ 0.426178,\ 0.329975,\ 0.260701,\ 0.204394,\ 0.153593,\ 0.101142)^*,$$

$$v_0 = \frac{\tilde{v}_0}{\|\tilde{v}_0\|}, \quad \|v\| = \sum_k |v_k|,\ \ell^1\text{-norm}$$

180 times, 10^3 iterations, 64 pages

Eigenvector g and initial vector v_0





The figure of $-z_k$ for $k = 0, 1, \ldots, 1000.$

Convergence speed of Power Md

(λ_j, g_j). Write $v_0 = \sum_{j=0}^N c_j g_j$, $c_0 \neq 0$

$$A^k v_0 = \sum_{j=0}^N c_j \lambda_j^k g_j$$

$$= c_0 \lambda_0^k \left[g_0 + \sum_{j=1}^N \frac{c_j}{c_0} \left(\frac{\lambda_j}{\lambda_0} \right)^k g_j \right]$$

$$\frac{A^k v_0}{\|A^k v_0\|} = \frac{c_0}{|c_0|} g_0 + O\left(\left| \frac{\lambda_1}{\lambda_0} \right|^k \right)$$

where $|\lambda_1| := \max\{|\lambda_j| : j > 0\}$.

(Shift) Inverse iteration, 1944

For given v_0, at the kth step, define

$$v_k = \frac{A^{-k} v_0}{\|A^{-k} v_0\|}, \qquad k \geqslant 1$$

i.e., the input-output method in economy

(Shift) Inverse iteration, 1944

For given v_0, at the kth step, define

$$v_k = \frac{A^{-k} v_0}{\|A^{-k} v_0\|}, \qquad k \geqslant 1$$

i.e., the input-output method in economy

$\Longleftrightarrow$ Double integral approx (C. 2001)

Shift: $$v_k = \frac{(zI - A)^{-k} v_0}{\|(zI - A)^{-k} v_0\|}, \qquad k \geqslant 1$$

Rayleigh quotient iteration, 1944

Choose $(z_0, v_0) \approx (\rho(A), g)$ with $v_0^* v_0 = 1$, where $v^* =$ transpose of v. L^2-norm. Particular, $z_0 = v_0^* A v_0$ for given v_0. At the kth step ($k \geqslant 1$), def

$$v_k = \frac{(z_{k-1} I - A)^{-1} v_{k-1}}{\|(z_{k-1} I - A)^{-1} v_{k-1}\|},$$

$$z_k = v_k^* A v_k \left[= \frac{v_k^* A v_k}{v_k^* v_k} \longleftarrow \mathrm{RQ} \right]$$

Rayleigh quotient iteration, 1944

where $I =$ identity. Then

$$v_k \to g, \quad z_k \to \rho(A) \qquad \text{as } k \to \infty$$

provided (z_0, v_0) is closed enough to $(\rho(A), g)$.

What can we expect for RQI?

100 iterations?

What can we expect for RQI?

100 iterations?

Example

The same matrix Q and $\tilde{v}_0$, need 2 steps only:

$$z_1 \approx -0.528215, \quad z_2 \approx -0.525268.$$

$\tilde{v}_0$ efficient!

“Too Good” is dangerous. Pitfall

$\lambda_j := \lambda_j(-Q)$. $\lambda_0 = -\rho(Q) > 0$.

Example

Let Q be the same as above. Choose

$$v_0 = (1, 1, 1, 1, 1, 1, 1, 1)^*/\sqrt{8}.$$

Then $\lambda_2 \approx 5.91867$

$(z_1, z_2, z_3, z_4) \approx (4.78557, 5.67061, 5.91766, 5.91867)$

Google's PageRank 1998

Langville, A.N., Meyer, C. D. (2006).

Google's PageRank and Beyond:

The Science of Search Engine Rankings.

Princeton University Press.

Power Iteration, included.

Inverse Iteration & RQI, not touched!

Need large system, fast algorithm

- Google's search–PageRank
- Input–output method in economy.
  Eigenvector
- Stability speed in stochastic systems.
  Stationary distribution: eigenvector
  Stability rate: eigenvalue
- Principal component analysis
  –BigData, five-diagonal

- Image recognition, Poisson, Toeplitz, Block-tridiagonal matrices. PDE
- Random algorithm, MCMC
- Phase transitions. Next eigenpair

Estimation of $(g, \rho(A))$, central (hard) problem in many branches of math!

- Image recognition, Poisson, Toeplitz, Block-tridiagonal matrices. PDE
- Random algorithm, MCMC
- Phase transitions. Next eigenpair

Estimation of $(g, \rho(A))$, central (hard) problem in many branches of math!

For large N, guess: # of iterations $\sim N^\alpha$.

$N = 10^4$? Subvert/overturn

Large N. $\lambda_0 = 1/4$ if $N = \infty$. $\leqslant 30$ Sec

Use $\tilde{v}_0$ and δ_1. Let $z_0 = 7/(8\delta_1) + v_0^*(-Q)v_0/8$.

$N+1$	z_0	z_1	$z_2 = \lambda_0$
10^4	0.31437	0.302586	0.302561

Examples for $N > 8$ computed by Yue-Shuang Li

Large N. $\lambda_0 = 1/4$ if $N = \infty$. $\leqslant 30$ Sec

Use $\tilde{v}_0$ and δ_1. Let $z_0 = 7/(8\delta_1) + v_0^*(-Q)v_0/8$.

$N+1$	z_0	z_1	$z_2 = \lambda_0$	upper/lower
8	0.523309	0.525268	**0.525268**	$1+10^{-11}$
100	0.387333	0.376393	**0.376383**	$1+10^{-8}$
500	0.349147	0.338342	**0.338329**	$1+10^{-7}$
1000	0.338027	0.327254	**0.32724**	$1+10^{-7}$
5000	0.319895	0.30855	**0.308529**	$1+10^{-7}$
7500	0.316529	0.304942	**0.304918**	$1+10^{-7}$
10^4	0.31437	0.302586	**0.302561**	$1+10^{-7}$

Examples for $N > 8$ computed by Yue-Shuang Li

I Efficient initials. Tridiagonal case

$E = \{0, 1, \ldots, N\}$.

$Q^c = A - mI$, $m := \max_{i \in E} \sum_{j \in E} a_{ij}$.

$Q^c = (q_{ij}) =$

$$\begin{pmatrix} -b_0 - c_0 & b_0 & 0 & 0 & \cdots \\ a_1 & -a_1 - b_1 - c_1 & b_1 & 0 & \cdots \\ 0 & a_2 & -a_2 - b_2 - c_2 & b_2 & \cdots \\ \vdots & \vdots & \ddots & \ddots & \ddots \\ 0 & 0 & 0 & \cdots\ a_N & -a_N - c_N \end{pmatrix}$$

$a_i > 0, \quad b_i > 0, \quad c_i \geqslant 0, \ \neq 0.$

$q_{k,k+1} = b_k, \ q_{k,k-1} = a_k, \ q_{kk} = -a_k - b_k - c_k$

Define $\{h_n\}$. Explicit

$$h_0 = 1, \quad h_n = h_{n-1} r_{n-1}, \quad 1 \leqslant n \leqslant N,$$

where $r_0 = 1 + c_0/b_0$,

$$r_n = 1 + \frac{a_n + c_n}{b_n} - \frac{a_n}{b_n r_{n-1}}, \quad 1 \leqslant n < N,$$

$$c_i \equiv 0 (i < N) \Longrightarrow h_i \equiv 1.$$

$Q^c \backslash$the last row $h = 0$.

$$\widetilde{Q} = \text{Diag}(h_i)^{-1} Q^c \, \text{Diag}(h_i).$$

Define $\{\mu_n\}$ and $\{\varphi_n\}$. Explicit

$$Q = \widetilde{Q}$$

$$\begin{pmatrix} -b_0 & b_0 & 0 & 0 & \cdots \\ a_1 & -a_1 - b_1 & b_1 & 0 & \cdots \\ 0 & a_2 & -a_2 - b_2 & b_2 & \cdots \\ \vdots & \vdots & \ddots & \ddots & \ddots \\ 0 & 0 & 0 \; \cdots & a_N & -a_N - b_N \end{pmatrix},$$

where $a_i, \ b_i > 0$. (c_i) erased!

$$\mu_0 = 1, \ \mu_n = \mu_{n-1} b_{n-1} / a_n, \ 1 \leqslant n \leqslant N.$$

$$\varphi_n = \sum_{k=n}^{N} (\mu_k b_k)^{-1}, \quad 0 \leqslant n \leqslant N.$$

RQI: tridiagonal case

$$\tilde{v}_0(i) = \sqrt{\varphi_i}, \ i \leqslant N; \ v_0 = \tilde{v}_0 / \|\tilde{v}_0\|_{\mu,2};$$

$$\delta_1 = \max_{0 \leqslant n \leqslant N} \left[\sqrt{\varphi_n} \sum_{k=0}^{n} \mu_k \sqrt{\varphi_k} + \frac{1}{\sqrt{\varphi_n}} \sum_{n+1 \leqslant j \leqslant N} \mu_j \varphi_j^{3/2} \right] =: z_0^{-1}.$$

Solve w_k: $(-Q - z_{k-1} I) w_k = v_{k-1}$,

$$v_k = \frac{w_k}{\|w_k\|_\mu} \to g, \ z_k = (v_k, -Q v_k)_\mu \to \lambda_0$$

Ⅱ Efficient initials. General case

Let $A = (a_{ij} : i, j \in E)$ be irreducible, $a_{ij} \geqslant 0$, $\forall i \neq j$. Define $A_i = \sum_{j \in E} a_{ij}$ and

$$Q^c = A - \Big(\max_{i \in E} A_i\Big) I.$$

Let $h = (h_0, h_1, \ldots, h_N)^*$ (with $h_0 = 1$) solve $Q^c \setminus \text{the last row}\, h = 0$ and define $\widetilde{Q} = \text{Diag}(h_i)^{-1} Q^c\, \text{Diag}(h_i)$.

$$c_0 = \ldots = c_{N-1} = 0,\ c_N =: q_{N,N+1} > 0$$

$Q := \widetilde{Q}$. Let $\varphi = (\varphi_0, \varphi_1, \ldots, \varphi_N)^*$ (with $\varphi_0 = 1$) solve

$$\varphi^{\setminus\text{the first row}} = P^{\setminus\text{the first row}}\,\varphi,$$ where

$$P = \text{Diag}\big((-q_{ii})^{-1}\big)Q + I.$$

$\mu := (\mu_0, \mu_1, \ldots, \mu_N)$ with $\mu_0 = 1$ solves $Q^{*\setminus\text{the last row}}\mu^* = 0$.

Algorithm. Def $\tilde{v}_0(i) = \sqrt{\varphi_i},\ i \leqslant N$;

$$v_0 = \tilde{v}_0 / \|\tilde{v}_0\|_\mu; \quad z_0 = (v_0, -Qv_0)_\mu.$$

Alternative z_0

For $k \geqslant 1$, let w_k solve

$(-Q - z_{k-1} I) w_k = v_{k-1}$ and set

$v_k = w_k / \|w_k\|_\mu$ $\boxed{\to g}$,

$z_k = (v_k, -Qv_k)_\mu$ $\boxed{\to \lambda_0 \text{ as } k \to \infty.}$

Alter choice of $z_0 = (v_0, -Qv_0)_\mu$:

$$z_0^{-1} = (1 - \varphi_1)^{-1} \max_{0 \leqslant n \leqslant N} \left[\sqrt{\varphi_n} \times \sum_{k=0}^{n} \mu_k \sqrt{\varphi_k} + \frac{1}{\sqrt{\varphi_n}} \sum_{n+1 \leqslant j \leqslant N} \mu_j \varphi_j^{3/2} \right]$$

Example. Block-tridiagonal matrix

$$Q = \begin{pmatrix} A_0 & B_0 & 0 & 0 & \cdots \\ C_1 & A_1 & B_1 & 0 & \cdots \\ 0 & C_2 & A_2 & B_2 & \cdots \\ \vdots & \vdots & \ddots & \ddots & \ddots \\ 0 & 0 & 0 & C_N & A_N \end{pmatrix}$$

$A_k, B_k, C_k:$ 40×40

$B, C's$: Identity

$A's$: tridiagonal

$N+1$	z_0	z_1	$z_2 = \lambda_0$
1600	7.985026	7.988219	7.988263
3600	7.993232	7.994676	7.994696
6400	7.996161	7.988256	7.987972

Example. Toeplitz Matrix [Xu Zhu]

$$A = \begin{pmatrix} 1 & 2 & 3 & \cdots & n-1 & n \\ 2 & 1 & 2 & \cdots & n-2 & n-1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ n-1 & n-2 & n-3 & \cdots & 1 & 2 \\ n & n-1 & n-2 & \cdots & 2 & 1 \end{pmatrix}$$

$N+1$	$z_0 \times 10^6$	$z_1 \times 10^6$	$z_2 \times 10^6$	$z_3 = \lambda_0$
1600	0.156992	0.451326	0.390252	0.389890
3600	0.157398	2.30731	1.97816	1.97591
6400	0.157450	7.32791	6.25506	6.24718

Ⅲ Global algorithms

Tridiagonally dominant matrices.

More negative? complex?

Analytic: Block-tri, lower triangular $+\mathbf{1}$,

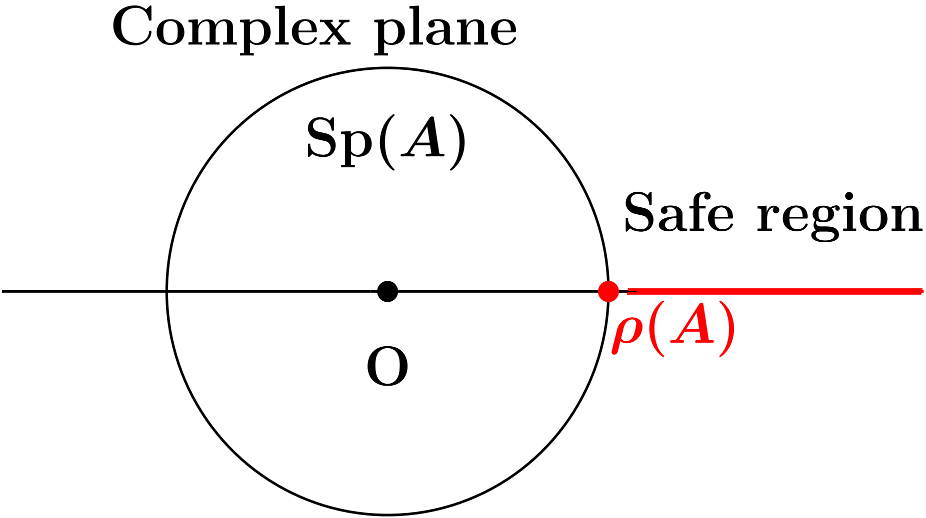
Upper triangular $+$ one line, ... AlphaGo

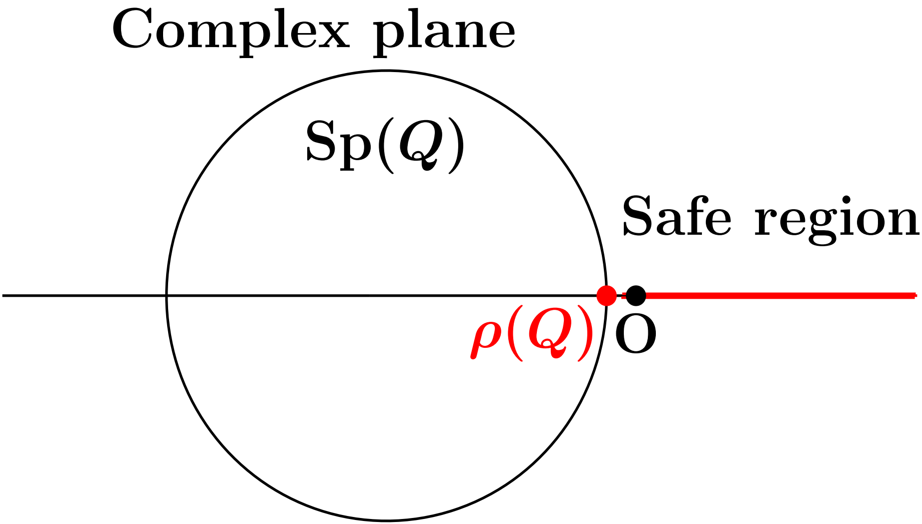
Ask computer to find efficient initials?

v_0=uniform distr. $[(1, \cdots, 1)^*/\sqrt{N+1}\,]$

How to choose z_0 (z_n) in unified way?

$$z_0 \neq \frac{v_0^* A v_0}{v_0^* v_0} \text{ [RQ!]}, \qquad z_n = ?$$


Complex plane
Sp(A)
Safe region
$\rho(A)$
O


Complex plane
Sp(Q)
Safe region
ρ(Q)
O

Ⅲ Global algorithms: A1 & A2

$v_0 =$ unif. distri., $\quad z_0 = \max_{0 \leqslant i \leqslant N} \frac{A v_0}{v_0}(i).$

A1. Specific Rayleigh quotient iteration

(C. 2016: §4.1 with Choice I)

For $k \geqslant 1$, $\quad v := v_{k-1}$, $\quad z := z_{k-1}$,

let w solve equation $(zI - A)w = v$.

Set $v_k = w/\|w\|$, $\quad z_k = v_k^* A v_k$.

A2. Shifted inverse iteration

$$z_k = \max_{0 \leqslant i \leqslant N} \frac{A v_k}{v_k}(i).$$

Lower trianglar + upper-diagonal

$$Q=$$

$$\begin{pmatrix} -1 & 1 & 0 & 0 & \cdots\cdots & 0 & 0 \\ a_1 & -a_1-2 & 2 & 0 & \cdots\cdots & 0 & 0 \\ a_2 & 0 & -a_2-3 & 3 & \cdots\cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \cdots\cdots & N-1 & 0 \\ a_{N-1} & 0 & 0 & 0 & \cdots & -a_{N-1}-N & N \\ a_N & 0 & 0 & 0 & \cdots\cdots & 0 & -a_N-N-1 \end{pmatrix}$$

$$q_{k,k+1}=k+1,\ q_{k0}=1/(k+1).$$

We have computed 4 cases:

$$a_k=1/(k+1),\quad \equiv 1,\quad =k,\quad =k^2.$$

The outputs for different N by Algorithm 2

$N+1$	z_1	z_2	z_3
8	0.276727	0.427307	0.451902
16	0.222132	0.367827	0.399959
32	0.187826	0.329646	0.370364
50	0.171657	0.311197	0.357814
100	0.152106	0.287996	0.343847
500	0.121403	0.247450	0.321751
1000	0.111879	0.233257	0.313274
5000	0.0947429	0.205212	0.293025
10^4	0.0888963	0.194859	0.284064

The outputs for different N by Algorithm 2

$N+1$	z_4	z_5	z_6
8	0.452339		
16	0.400910		
32	0.372308	0.372311	
50	0.360776	0.360784	
100	0.349166	0.349197	
500	0.336811	0.337186	
1000	0.334155	0.335009	0.335010
5000	0.328961	0.332609	0.332635
10^4	0.326285	0.332113	0.332188

Upper trianglar + lower-diagonal

$$Q=$$

$$\begin{pmatrix} -1 & p_2 & p_3 & p_4 & \cdots\cdots\cdots & p_{N-1} & \sum_{k\geqslant N} p_k \\ 2p_0 & -2 & 2p_2 & 2p_3 & \cdots\cdots\cdots & 2p_{N-2} & 2\sum_{k\geqslant N-1} p_k \\ 0 & 3p_0 & -3 & 3p_2 & \cdots & 3p_{N-3} & 3\sum_{k\geqslant N-2} p_k \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & -N+1 & (N-1)\sum_{k\geqslant 2} p_k \\ 0 & 0 & 0 & 0 & \cdots\cdots\cdots & Np_0 & -Np_0 \end{pmatrix}$$

$$p_0=\alpha/2,\ p_1=0,\ p_n=(2-\alpha)/2^n,$$

$\alpha \in (4/3, 2)$. $\alpha = 7/4$.

The outputs in the subcritical case

N	z_1	z_2	z_3	z_4
8	0.637800	0.638153		
16	0.621430	0.625490	0.625539	
50	0.609976	0.624052	0.624997	0.625000
100	0.606948	0.623377	0.624991	0.625000
500	0.604409	0.622116	0.624962	0.625000
10^3	0.604082	0.621688	0.624944	0.625000
$\frac{10^4}{2}$	0.603817	0.620838	0.62489	0.625000
10^4	0.603784	0.620511	0.624861	0.625000

Matrix with more "−" elements

$$A = \begin{pmatrix} -1 & 8 & -1 \\ 8 & 8 & 8 \\ -1 & 8 & 8 \end{pmatrix}.$$

Eigenvalues of A: **17.5124**, $-7.4675, 4.95513$.

n	z_n: Algorithm 1	z_n: Algorithm 2
1	17.3772	18.5316
2	**17.5124**	17.5416
3		17.5124

Complex matrix $A =$

$$\begin{pmatrix} 0.75-1.125\,i & 0.5882-0.1471\,i & 1.0735+1.4191\,i \\ -0.5-i & 2.1765+0.7059\,i & 2.1471-0.4118\,i \\ 2.75-0.125\,i & 0.5882-0.1471\,i & -0.9265+0.4191\,i \end{pmatrix}$$

Eigenvalues of A: 3, $-2-i$, $1+i$.

Outputs of "Algorithm 2"

y_1	$3.03949-0.0451599\,i$
y_2	$3.00471-0.0015769\,i$
y_3	3 Fast cubic

$\mathrm{Re}(A^n) > 0 \ \ \forall n \gg 1$ up to mI.

For Further Reading I

C. (2016): Efficient initials for computing the maximal eigenpair, Front. Math. China 11(6): 1379–1418.

A package based on the paper is available on CRAN now [by X.J. Mao]: https://cran.r-project.org/web/packages/EfficientMaxEigenpair/index.html

For Further Reading II

- C. (2017a). Global algorithms for maximal eigenpair. Preprint
- C. (2017b). Efficient algorithm for principal eigenpair of discrete p-Laplacian. Preprint

http://math0.bnu.edu.cn/~chenmf

4 volumes, 19 popularizing papers, 8 videos

The end!

Thank you, everybody!

谢谢大家！

49