Estimating matrix functionals via extrapolation

Marilena Mitrouli

Collaboration team: Dr Paraskevi Fika,
Paraskevi Roupa (PhD student)

HELLENIC REPUBLIC
National and Kapodistrian University of Athens
EST. 1837

NASCA' 18, Kalamata, Greece, 2-6 July 2018

MAIN TOPICS

- Motivation for the approximation of binary and square inverse forms $\mathbf{y^T A^{-1} x}$ and $\mathbf{y^T A^{-2} x}$.
- Extrapolation methods and estimates: Recent progress.
- Applications: Estimation of the diagonal of A^{-1}.
  The precision matrix in Statistics.
- Estimation of the GCV function.
- Numerical examples.

Motivation for estimating bilinear forms

	Classes of problems	**Application**
(I)	Statistics - Uncertainty quantification	Diagonal of the inverse
(II)	ill-posed problems (regularization)	GCV function
(III)	Real-world networks	Matrix resolvent
(IV)	Generalized matrix functions	Resolvent-based communicability.

Applications of bilinear forms

- *Determinantal Point Processes* (DPP in machine learning) for $L \in \mathbb{R}^{N \times N}$ positive definite $u^T L^{-1} u$, u from uniform sampling

  DPP judge

  Return true if $t < u^T L^{-1} u$, false otherwise

  C. Li, S. Sra, S. Jegelka, *Gaussian quadrature for matrix inverse forms with applications*, Inter. Conf. on Mach Learning, NY, 2016.

- $Tr((AA^T + \mu I)^{-1})$ in GCV function

  C. Fenu, L. Reichel, G. Rodriguez, *GCV for Tikhonov regularization via global Golub-Kahan decomposition*, Numer. Linear Algebra Appl. 23, (2016), pp. 467-484.

- Diagonal of precision matrix.

Tools of evaluation

- **explicitly**

  tools: classical matrix comput. methods $\hookrightarrow \mathbf{O(p^3)}$
- **Estimation**

  Riemann-Stieltjes integral problem approximate the integral by quadrature rules

  $\hookrightarrow \mathbf{O(jp^2)}$, j =number of iterations

  tools: $\begin{cases} \text{orthogonal polynomials} \\ \text{Lanczos procedure} \end{cases}$

- G.H. Golub, G. Meurant, *Matrices, Moments and Quadrature with Applications*, Princeton University Press, Princeton, 2010.

- Z. Bai, M. Fahey, G. Golub, *Some large scale computation problems*, J. Comput. Appl. Math., 74 (1996), pp. 71–89.

- C. Fenu, D. Martin, L. Reichel, G. Rodriguez, *Network analysis via partial spectral factorization and Gauss quadrature*, SIAM J. Sci. Comput., 35 (2013), pp. A2046-A2068.

The general problem

Let $\mathbf{A} \in \mathbb{R}^{\mathbf{p}\times\mathbf{p}}$ be a diagonalizable matrix.

We are interested in obtaining inexpensive estimations of

the bilinear inverse form (BIF) $\mathbf{y}^{\mathbf{T}}\mathbf{A}^{-\mathbf{1}}\mathbf{x}$ and
the square inverse form (SIF) $\mathbf{y}^{\mathbf{T}}\mathbf{A}^{-\mathbf{2}}\mathbf{x}$.

↪ Flexibility in the definition of the moments of A:

- **bilinear moments of $\mathbf{A}$**

  $c_r(x,y) = (x, A^r y)$ ↪ direct estimates.

- **quadratic moments of A**

  $c_r = (x, A^r x)$ ↪ polarization identity.

The mathematical framework

Mathematical tools

1. The decomposition of A

Assuming that the matrix A has the factorization

$$A = Q\Lambda Q^{-1},$$

$\mathbf{Q} = [\mathbf{q_1}, \ldots, \mathbf{q_p}]$ nonsingular, $\Lambda = diag[\lambda_1, \ldots, \lambda_p]$ and $\hat{\mathbf{q}}_i^T$, $i = 1, 2, \ldots, p$, are the rows of Q^{-1}, $\hat{q}_i q_j = \delta_{ij}$.
Then

$$\mathbf{f(A) = Qf(\Lambda)Q^{-1} = \sum_{j=1}^{p} f(\lambda_j) q_j \hat{q}_j^T}.$$

The mathematical framework

2. The moments

2.1 The quadratic moments

The **quadratic moments** of A wrt a vector x are defined by

$$\mathbf{c_r(x) = (x, A^r x) = \sum_k \lambda_k^r \alpha_k^2(x)}, \quad \mathbf{\alpha_k(x) = (x, q_k)}.$$

2.2 The bilinear moments

The **bilinear moments** of A wrt vectors x, y are defined by

$$\mathbf{c_r(x, y) = (y, A^r x) = \sum_k \lambda_k^r \alpha_k \beta_k},$$

$\alpha_k = (y, q_k),\ \beta_k = (\hat{q}_k, y)$.

Both the moments $c_i(x)$ and $c_i(x, y)$ are denoted by c_i for notational simplicity.

For $r = -1$, $c_{-1}(x, y) = BIF$ and for $r = -2$, $c_{-2}(x, y) = SIF$.

- Extrapolation of moments

By keeping only one or two terms in the summations.

Using some moments with a **non-negative integer index $\mathbf{n} \in \mathbb{N}$**, we **estimate** the moments c_{-1} and c_{-2}

by **extrapolating** the $\mathbf{c_n}$'s, at the point $\mathbf{-1}$ and $\mathbf{-2}$.

The values of λ_k and α_k are determined via interpolation conditions.

-P. Fika, M. M., Estimation of the bilinear form $y^* f(A)x$ for Hermitian matrices, Linear Algebra Appl., 502 (2016), pp. 140–158.

-C. Brezinski, Error estimates for the solution of linear systems, SIAM J. Sci. Comput., 21 (1999), pp. 764–781.

One–term estimates via extrapolation

Knowing $\mathbf{c_0, c_1, c_2}$, we look for $\mathbf{l}$ and $\mathbf{a}$ satisfying the **interpolation conditions**

$$\begin{aligned} \mathbf{c_0} &= \mathbf{a^2} \\ \mathbf{c_1} &= \mathbf{la^2} \\ \mathbf{c_2} &= \mathbf{l^2a^2} \end{aligned}$$

and, then, $\mathbf{c_{-1}}$ will be **estimated** by $\mathbf{c_{-1} \simeq l^{-1}a^2}$

Therefore, $\mathbf{c_{-1}} \simeq \mathbf{e}_\nu = \dfrac{\mathbf{c_0^2}}{\rho^\nu \mathbf{c_1}}, \quad \rho = \dfrac{\mathbf{c_0 c_2}}{\mathbf{c_1^2}} \geq \mathbf{1}, \quad \nu \in \mathbb{R}.$

- If $\rho = \mathbf{1}$ then $\mathbf{e}_\nu = \mathbf{c_{-1}}, \quad \forall \nu \in \mathbb{R}.$
- There exists an optimal value $\nu_\mathbf{o}$ such that $\mathbf{e}_{\nu_\mathbf{o}} = \mathbf{c_{-1}}.$

Specification of ν_o

- **For BIF:**

Proposition

Let $A \in \mathbb{R}^{n \times n}$ *be a symmetric matrix,* $x \in \mathbb{R}^n$, $y = Ax \in \mathbb{R}^n$ *and let* $L = \sum_{k,j=1}^{n} (x_k y_j - y_j x_k)^2 / 2$. *If for some* $\epsilon \in \mathbb{R}$, $\epsilon << 1$ $\exists$ $\delta_1(\epsilon) < \delta_2(\epsilon) : L << \delta_1(\epsilon)$ *then* $\nu_o \simeq -1$ *and otherwise, if* $L \in (\delta_1(\epsilon), \delta_2(\epsilon))$, *then* $\nu_o \simeq \dfrac{log(c_1^2/(c_0 c_2))}{log(c_1 c_3 / c_2^2)}$.

Corollary

If $\rho \simeq 1$, *then* $\nu_o \simeq -1$.

Specification of ν_o (cont'd)

Corollary

Let $A \in \mathbb{R}^{n\times n}$ be a symmetric positive definite matrix. Then, the optimal value ν_o can be bounded by

$$\frac{\log(\lambda_n c_0/c_1)}{\log(\rho)} \leq \nu_o \leq \frac{\log(\lambda_1 c_0/c_1)}{\log(\rho)}, \quad \rho \neq 1$$

$$\nu_o \leq \frac{\log((c_0^2 c_1 c_3 - c_0^2 c_2^2)/(c_1^4 + c_0^2 c_1 c_3 - 2 c_0 c_1^2 c_2))}{\log(\rho)}.$$

$\lambda_1 = \lambda_{max}, \quad \lambda_p = \lambda_{min}.$

Specification of ν_o (cont'd)

- **For SIF:**

Proposition

Let $A \in \mathbb{R}^{n\times n}$ be a symmetric matrix, $x \in \mathbb{R}^n$, $y = Ax \in \mathbb{R}^n$ and $L = \sum_{1\le i<j\le n} (x_i y_j - x_j y_i)^2$. Let ϵ be a real number such that $\epsilon \ll 1$. If $L \in [0,\epsilon)$, then the optimal value for the one-term family parameter for the SIF can be approximated by $\nu_o \simeq -3/2$.

Corollary

If $\rho \simeq 1$, then $\nu_o \simeq -3/2$.

Two–term estimates via extrapolation

By keeping **two terms** $\mathbf{c_{-1} \simeq l_1^{-1} a_1^2 + l_2^{-1} a_2^2}$.

Interpolation conditions: $\mathbf{c_n = l_1^n a_1^2 + l_2^n a_2^2, \ n \in \mathbb{Z}}$

c_n's satisfy **the difference equation**

$\mathbf{c_{n+1} - s c_n + q c_{n-1} = 0, \quad s = l_1 + l_2, \ q = l_1 l_2}$,

Solving the system of $\mathbf{c_{n+1} - s c_n + q c_{n-1} = 0}$ and

$\mathbf{c_{n+2+k} - s c_{n+1+k} + q c_{n+k} = 0, \ k, n \in \mathbb{Z}}$, we get

$$\mathbf{s = \frac{c_{n-1} c_{n+2+k} - c_{n+1} c_{n+k}}{c_{n-1} c_{n+1+k} - c_n c_{n+k}}, \quad q = \frac{c_n c_{n+2+k} - c_{n+1} c_{n+1+k}}{c_{n-1} c_{n+1+k} - c_n c_{n+k}}},$$

$$\mathbf{l_{1,2} = (r \pm \sqrt{s^2 - 4q})/2, \quad a_1^2 = \frac{c_0 l_2 - c_1}{l_2 - l_1}, \quad a_2^2 = \frac{c_1 - c_0 l_1}{l_2 - l_1}}.$$

Then, $\mathbf{c_{-1} \simeq \hat{e}_{n,k} = (c_0 s - c_1)/q}$

Three–term estimates via extrapolation

By keeping **three terms** $\mathbf{c_{-1}(x) \simeq l_1^{-1}a_1^2 + l_2^{-1}a_2^2 + l_3^{-1}a_3^2.}$

Interpolation conditions: $\mathbf{c_n(x) = l_1^n a_1^2 + l_2^n a_2^2 + l_3^n a_3^2,\ n \in \mathbb{Z}}$

$c_n(x)$'s satisfy **the difference equation**

$$\mathbf{c_{n+2} - sc_{n+1} + tc_n - qc_{n-1} = 0,\ n \in \mathbb{Z},}$$

where $s = l_1 + l_2 + l_3,\ t = l_1 l_2 + l_1 l_3 + l_2 l_3$ and $q = l_1 l_2 l_3$.

Solving the system of

$$\mathbf{c_{n+2} - sc_{n+1} + tc_n - qc_{n-1} = 0,}$$
$$\mathbf{c_{n+k+3} - sc_{n+k+2} + tc_{n+k+1} - qc_{n+k} = 0,}$$
$$\mathbf{c_{n+\ell+4} - sc_{n+\ell+3} + tc_{n+\ell+2} - qc_{n+\ell+1} = 0, n, k, \ell \in \mathbb{Z},}$$

we can obtain formulae for the quantities s, t and q which are dependent on the moments c_n, for different integers n.

Three-term estimates via extrapolation (Cont'd)

Therefore, we can obtain a family of **three-term estimates** $\tilde{\mathbf{e}}_{\mathbf{n},\mathbf{k},\ell}$ for the BIF which is given by

$$\mathbf{c_{-1}(x) \simeq \tilde{e}_{n,k,\ell}(x) = l_1^{-1} a_1^2 + l_2^{-1} a_2^2 + l_3^{-1} a_3^2},$$

where

$$a_1^2 = \frac{c_2 - l_2 c_1 - l_3 c_1 + l_2 l_3 c_0}{(l_1 - l_2)(l_1 - l_3)},$$

$$a_2^2 = -\frac{c_2 - l_1 c_1 - l_3 c_1 + l_1 l_3 c_0}{(l_1 - l_2)(l_2 - l_3)},$$

$$a_3^2 = \frac{c_2 - l_1 c_1 - l_2 c_1 + l_1 l_2 c_0}{(l_1 - l_3)(l_2 - l_3)},$$

Three-term estimates via extrapolation (Cont'd)

$$l_{1,2} = \alpha \pm \beta i,$$

$$\alpha = \frac{s}{3} - \frac{A_p}{3 \cdot 2^{4/3}} + \frac{3p - s^2}{3 \cdot 2^{2/3} A_p}, \quad \beta = 2^{-2/3} 3^{-1/2} \frac{3p - s^2 + 2^{-2/3} A_p^2}{A_p},$$

$$l_3 = B_p + \frac{s}{3} + \frac{s^2 - 3p}{9B_p},$$

$$A_p = \left(27q + 3\sqrt{3}\sqrt{4t^3 - t^2 s^2 - 18tqs + 27q^2 + 4qs^3} - 9ts + 2s^3\right)^{1/3}$$

$$B_p = \left(\frac{q}{2} - \frac{ts}{6} + \sqrt{\left(\frac{s^3}{27} - \frac{ts}{6} + \frac{q}{2}\right)^2 + \left(\frac{t}{3} - \frac{s^2}{9}\right)^3} + \frac{s^3}{27}\right)^{1/3}.$$

We note that $\tilde{e}_{n,k,\ell}(x) = \sum_{i=1}^{3} l_i^{-1} a_i^2 \in \mathbb{R}$.

Complexity

Complexity

The estimates require only **few matrix–vector products** and **some inner products**.

- $\mathbf{e}_{\nu}$ needs only $\mathbf{O(p^2)}$ flops **(1 mvp)**,
- $\mathbf{e}_{1,0}$ requires $\mathbf{O(2p^2)}$ flops.

Extrapolation	
$\mathbf{e}_{\nu}$	$\mathbf{e}_{n,k}$
$\mathcal{O}(\mathbf{p^2})$	$\mathcal{O}(\lceil \frac{\mathbf{n+k+2}}{\mathbf{2}} \rceil \mathbf{p^2})$

- The BIF requires the double complexity.

Complexity of the three-term estimates

Family of three-term estimates $\tilde{e}_{n,k,\ell}$.
Set $\mu = \max\{n+k+3, n+\ell+4\}$.

dense	banded (bandwidth= s)
$\mathcal{O}\left(\mu p^2\right)$	$\mathcal{O}\left(\mu s p\right)$

Table: Computational complexity of three-term estimates for BIF.

Good choice for family parameters: **integers** of **small** values resulting to an **inexpensive computation**, usually from zero to 2 or 3.

By taking the mean value of these approximations we can have a better estimation of the desired quantity.

Applications of BIF

Applications

- Estimating the elements of A^{-1} via the one-term estimates.
- **Statistics**

  (estimation of the precision matrix)
- BIF via two-term and three-term estimates.

Estimation of the elements of $\mathbf{A}^{-1}$

Let $\mathbf{A} = [\mathbf{a_1}, \ldots, \mathbf{a_n}] \in \mathbb{R}^{\mathbf{n}\times\mathbf{n}}$.

$$(\mathbf{A}^{-1})_{\mathbf{ii}} \simeq \mathbf{d_i} = \frac{\mathbf{1}}{\rho^{\nu}\ \mathbf{a_{ii}}},$$

where $\mathbf{a_{ii}}$ is the *i*th diagonal entry of the matrix $\mathbf{A}$,

$\rho = \mathbf{z_i}/\mathbf{a_{ii}^2}, \quad \mathbf{z_i} = \mathbf{a_i^T a_i}, \quad \mathbf{i} = \mathbf{1}, \mathbf{2}, \ldots, \mathbf{n}, \quad \nu \in \mathbb{R}.$

Diagonal Estimation

$$\mathbf{diag(A^{-1}) \simeq 1./(R_a.\hat{}\nu. * d_a),}$$

where $\mathbf{R_a = S_a./(d_a.\hat{}2)}$, $\mathbf{d_a = diag(A)}$ and $\mathbf{S_a = [z_1, \ldots, z_n]}$.

$.\hat{}$ is the element-wise power operation

$./$ is the element-wise division

$.*$ is the element-wise multiplication.

Complexity

- the estimation of $(A^{-1})_{ii}$ requires 1 inner product
- the estimation of the whole diagonal requires n inner products

Estimation of the precision matrix in Statistics

Let $\tilde{\mathbf{X}} = [\mathbf{x_1}\ \mathbf{x_2} \dots \mathbf{x_p}]$ be the $\mathbf{m} \times \mathbf{p}$ **data matrix**.

The column x_j of $\tilde{\mathbf{X}}$ gives the j–th variable's scores for the m items.

Centered matrix $\mathbf{X}$: $\mathbf{X_{ij}} = \tilde{\mathbf{X}}_{\mathbf{ij}} - \sum_{\mathbf{k=1}}^{\mathbf{m}} \tilde{\mathbf{X}}_{\mathbf{kj}}/\mathbf{m}$.

$\mathbf{A} = \mathbf{X^T X/m}$ is the sample covariance matrix

(reveals correlations between variables)

$\mathbf{A^{-1}}$ is the precision matrix

(encodes conditional correlations between pairs of variables given the remaining variables)

Computing the matrix $\mathbf{A^{-1}}$: high computational complexity

$\hookrightarrow$ **estimates**

Numerical examples

Example 1: The precision matrix in uncertainty quantification: $\mathbf{A} = [\mathbf{a_{ij}}]$, $\mathbf{a_{ii}} = \mathbf{1} + \mathbf{i}$, $\mathbf{a_{ij}} = \frac{\mathbf{1}}{|\mathbf{i}-\mathbf{j}|}$ for $\mathbf{i} \neq \mathbf{j}$.

	Rel. error	Exec. time
1000	3.3003e-4	3.64e-4 sec
5000	6.6098e-5	1.39e-2 sec
10000	3.3055e-5	5.60e-2 sec
50000	6.6118e-6	1.49e0 sec
100000	3.3060e-6	9.64e0 sec

Mean relative error for the diagonal entries = $(\sum_{\mathbf{i}=1}^{\mathbf{p}} |(\mathbf{A}^{-1})_{\mathbf{ii}} - \mathbf{d_i}|/|(\mathbf{A}^{-1})_{\mathbf{ii}}|)/\mathbf{p}$.

Numerical examples

In uncertainty quantification:

$\mathbf{A} = [\mathbf{a_{ij}}]$, $\mathbf{a_{ii}} = \mathbf{1} + \mathbf{i}^{\mathbf{1/2}}$, $\mathbf{a_{ij}} = \frac{\mathbf{1}}{|\mathbf{i-j}|^{\mathbf{2}}}$ for $\mathbf{i} \neq \mathbf{j}$.

	Rel. error	Exec. time
4000	1.4211e-4	1.10e-2 sec

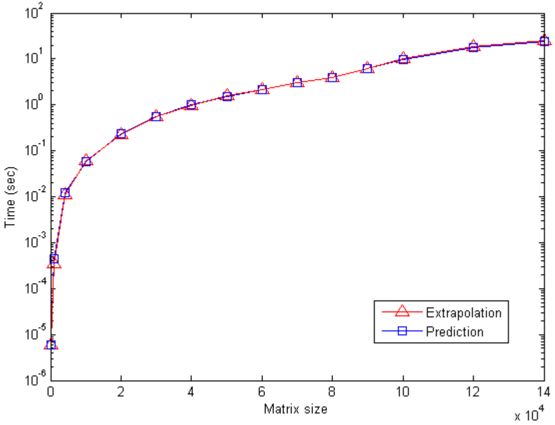
– In V. Kalantzis, C. Bekas, A. Curioni, E. Gallopoulos, Accelerating data uncertainty quantification by solving linear systems with multiple right-hand sides, Numer. Algor., 62 (2013), 637–653 the mean relative error has order $\mathcal{O}(10^{-4})$.

– For $p = 29000$, more that 100 sec are required. The extrapolation method requires almost 0.9 sec.

Numerical examples

In uncertainty quantification:

$\mathbf{A} = [\mathbf{a_{ij}}]$, $\mathbf{a_{ii}} = \mathbf{1} + \mathbf{i}^{\mathbf{1/2}}$, $\mathbf{a_{ij}} = \frac{\mathbf{1}}{|\mathbf{i}-\mathbf{j}|^{\mathbf{2}}}$ for $\mathbf{i} \neq \mathbf{j}$.



Numerical examples

In uncertainty quantification:

$\mathbf{A} = [\mathbf{a_{ij}}]$, $\mathbf{a_{ii}} = \mathbf{1} + \mathbf{i}$, $\mathbf{a_{ij}} = \frac{\mathbf{1}}{|\mathbf{i}-\mathbf{j}|}$ for $\mathbf{i} \neq \mathbf{j}$.

	$\mathbf{e_0}$ = Gauss $\mathbf{k = 1}$	$\mathbf{e_{\tilde{\nu}_o}}$	Gauss $\mathbf{k = 5}$	Gauss $\mathbf{k = 15}$	Gauss $\mathbf{k = 20}$
Rel. error	1.747e-1	5.985e-3	6.107e-2	1.224e-2	5.070e-3
Estimates	5.000e-1	6.022e-1	5.688e-1	5.984e-1	6.028e-1

Table: Relative errors and estimates for $\mathbf{A^{-1}_{1,1}} = \mathbf{6.0584e-1}$, for *Covariance* matrix A of dimension $\mathbf{p} = \mathbf{1000}$.

Numerical examples

Example 2: Estimating the GCV function

The **GCV function** $V(\mu)$ is given by the formula

$$V(\mu) = \frac{\| Ax_\mu - b \|^2}{(Tr(I_p - A_\mu))^2},$$

where

- $A_\mu = A(A^T A + \mu I_m)^{-1} A^T$ is the influence matrix
- μ is the **regularization parameter**.

By setting $B = AA^T + \mu I_p$, the GCV function is expressed as

$$V(\mu) = \frac{b^T B^{-2} b}{(Tr(B^{-1}))^2}$$

L. Reichel, G. Rodriguez, S. Seatzu, Error estimates for large-scale ill-posed problems, Numer. Algorithms 51(3), 341-361 (2009).

The **minimization of the GCV function** over μ leads to an **approximation** of the regularization parameter μ.

Numerical examples

We consider the **Heat test problem** from P.C. Hansen's software.

- size of the matrix A 100×100
- noise level $\sigma = 10^{-4}$
- x the true solution
- x_μ the regularized solution.

Method	μ	$\| x - x_\mu \|$
one-term	1.0389e-7	4.8332e-2
two-term	2.3073e-8	6.1547e-2
gcv	2.2734e-4	5.9325e-1

- P. C. Hansen, Regularization Tools Version 4.0 for MATLAB 7.3, Numerical Algorithms 46, 189-194 (2007).

Numerical examples

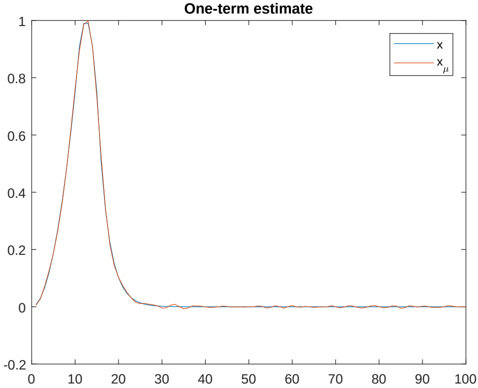


Figure: Solution of the Heat test problem by using the one-term estimates.

Numerical examples

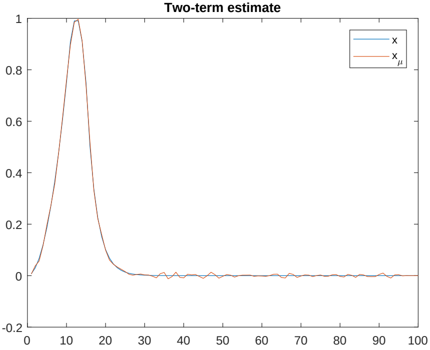


Figure: Solution of the Heat test problem by using the two-term estimates.

- M. M., P. Roupa, Estimates for the generalized cross-validation function via an extrapolation and statistical approach, Calcolo, (2018) 55: 24. https://doi.org/10.1007/s10092-018-0266-3.

Numerical examples

Example 3: Real-world networks

We consider the **real-world networks**

email ($p = 1133$), autobahn ($p = 1168$), internet ($p = 22963$).

- Specification of the most important node (n^*).
- Matrix resolvent $(I_p - aA)^{-1}$ for $a = 0.85/\lambda_{max}$.

Network	node n^*	time Extrapolation	time Single Gauss	time Block Gauss
email	105	2.3757e-4 sec	5.4249e-2 sec	5.7525e-2 sec
autobahn	693	2.5414e-4 sec	4.3187e-2 sec	5.3417e-2 sec
internet	4	3.6079e-3 sec	1.6920e1 sec	1.5087e1 sec

Table: The execution time for the determination of the most important node in real-world networks.

Numerical examples

Network	$\alpha = 0.9/\sigma_1$	$\alpha = 0.85/\sigma_1,$	$\alpha = 0.5/\sigma_1,$
Harvard500	1.8612e-3	6.6772e-4	2.7202e-6
Air500	9.9439e-4	3.2096e-4	6.2341e-7
Airlines	1.6613e-3	5.8440e-4	1.8022e-6
Roget	1.3686e-3	5.3213e-4	1.5874e-6
Celegans	2.6503e-3	9.6012e-4	2.7275e-6
GD06Java	5.1817e-4	1.8024e-4	4.7705e-7
Twitter	7.0058e-5	2.2371e-5	4.8174e-8

Table: Mean Relative errors estimating the entire diagonal of the matrix $(I - a^2 A^* A)^{-1}$ for directed networks.

Numerical examples

Network	n	edges	mean ρ	Rel. error	Time (sec)
Email	1133	10902	1.0162	6.9794e-3	5.0620e-4
Autobahn	1168	2486	1.1051	2.6378e-2	5.5800e-4
Power	4941	13188	1.0344	5.4984e-3	1.1617e-2
Internet	22963	96872	1.0006	2.0038e-4	3.6538e-1
Facebook	63731	1545686	1.0011	1.9522e-4	2.1193e0

Table: Estimating the whole diagonal of the matrix resolvent.

Numerical examples

Network	n	edges	mean ρ	Time (sec)
wing	62032	243088	1.1777	2.3192e0
fe rotor	99617	1324862	1.0346	6.9895e0
usroads	129164	330870	1.1210	1.0681e1
fe ocean	143437	819186	1.1160	1.6375e1
caidaRouterLevel	192244	1218132	1.0110	4.1321e1
coAuthorsCiteseer	227320	1628268	1.0007	6.8387e1

Table: Estimating the whole diagonal of the matrix resolvent.

Conclusions

The *Extrapolation* method

- provides simple formulae of low complexity.
- it can be implemented in vectorized forms for the approximation of the whole diagonal.
- attains a satisfactory execution time and a fair accuracy.
- it is strongly recommended in applications where a high accuracy is not required, such as in statistics and in networks.
- it is implemented in Matlab and in Fortran, on the high-performance computing system ARIS*.

*each node consists of Intel Xeon E5-2680v2 processors & 512 Gb RAM

Conclusions

Further work

- Thorough comparison of these estimates with other methods.
- Estimation of vector estimates and further application to real world problems.
- In Statistics, derivation of GCV estimates through extrapolation and their application to high dimensional statistical modelling.
- The implementation of all the families of estimates in a parallel computing environment.

Further References

- F. Arrigo, M. Benzi, C. Fenu, Computation of generalized matrix functions, SIAM Journal on Matrix Analysis and Applications, 37 (2016), pp. 836–860.
- C. Bekas, E. Kokiopoulou, and Y. Saad, An estimator for the diagonal of a matrix, Applied Numerical Mathematics, 57(11-12) (2007), pp. 1214–1229.
- M. Benzi, P. Boito, Quadrature rule-based bounds for functions of adjacency matrices, Linear Algebra and its Applications, 433 (2010), pp. 637–652.
- C. Brezinski, Error estimates for the solution of linear systems, SIAM J. Sci. Comput., 21 (1999), pp. 764–781.
- J. Fan, Y. Liao, H. Liu, An overview on the estimation of large covariance and precision matrices, Econometrics Journal, 19 (2016), pp. C1–C32.

Further References

- C. Fenu, D. Martin, L. Reichel, G. Rodriguez, Network analysis via Partial Spectral Factorization and Gauss Quadrature, SIAM J. Sci. Comput., 35(4) (2013), pp. 2046–2068.
- P. Fika, M. Mitrouli, P. Roupa, Estimating the diagonal of matrix functions, Math. Meth. Appl. Sci., 41 (2018), pp. 1083–1088.
- M. Mitrouli, P. Roupa, Vector estimates for $f(A)b$ via extrapolation, Electron. Trans. Numer. Anal., 47, (2017), pp. 179–196.
- J. Tang, Y. Saad, A probing method for computing the diagonal of a matrix inverse, Numer. Linear Algebra Appl., 19 (2012), pp. 485–501.