On finding multiplicities of characteristic polynomial factors of black-box matrices

J-G. DUMAS, C. PERNET and B. D. SAUNDERS

Grenoble Univ. France
University of Delaware, USA.

ISSAC 2009, Seoul,
July 31

Graph isomorphism

Problem

$\texttt{Graph-isomorphism} \in P$?

Graph isomorphism

Problem

$\mathtt{Graph\text{-}isomorphism} \in P$?

[Audenaert,& al. 2007] : the spectrum of a symmetric power of the graph determines its isomorphism class ???

Graph isomorphism

Problem

$\mathtt{Graph\text{-}isomorphism} \in P$?

[Audenaert,& al. 2007] : the spectrum of a symmetric power of the graph determines its isomorphism class ???

Experiments: symmetric powers of families of strongly regular graphs

Graph isomorphism

Problem

$\texttt{Graph-isomorphism} \in P$?

[Audenaert,& al. 2007] : the spectrum of a symmetric power of the graph determines its isomorphism class ???

Experiments: symmetric powers of families of strongly regular graphs

- $k = 2$: wrong ([Godsil, Royle & al. 2006])
- $k = 3$: true up to 29 edges (70 cases, $n = 3654$)
- $k = 3$: true up to 36 edges (36 510 cases, $n = 7140$)

Graph isomorphism

Problem

$\mathtt{Graph\text{-}isomorphism} \in P$?

[Audenaert,& al. 2007] : the spectrum of a symmetric power of the graph determines its isomorphism class ???

Experiments: symmetric powers of families of strongly regular graphs

- $k = 2$: wrong ([Godsil, Royle & al. 2006])
- $k = 3$: true up to 29 edges (70 cases, $n = 3654$)
- $k = 3$: true up to 36 edges (36 510 cases, $n = 7140$)

Compute characteristic polynomials over Z

Computing the characteristic polynomial

Overview on the main approaches:

Traces of powers: Leverrier 1881, Faddeev 59, ... $\mathcal{O}\left(n^4\right)$

$\Rightarrow$ $\boxed{\text{Dense, over a ring, best in parallel}}$

Determinant expansion: Samuelson 42, Berkowitz 84 $\mathcal{O}\left(n^4\right)$

$\Rightarrow$ $\boxed{\text{Dense, over a ring}}$

Computing the characteristic polynomial

Overview on the main approaches:

Traces of powers: Leverrier 1881, Faddeev 59, ... $\mathcal{O}\left(n^4\right)$

$\Rightarrow$ Dense, over a ring, best in parallel

Determinant expansion: Samuelson 42, Berkowitz 84 $\mathcal{O}\left(n^4\right)$

$\Rightarrow$ Dense, over a ring

Elimination based: Danilevskii 37, Hessenberg 41, ... $\mathcal{O}\left(n^3\right)$

$\Rightarrow$ Dense, over a field

Computing the characteristic polynomial

Overview on the main approaches:

Traces of powers: Leverrier 1881, Faddeev 59, ... $\mathcal{O}\left(n^4\right)$

$\Rightarrow$ $\boxed{\text{Dense, over a ring, best in parallel}}$

Determinant expansion: Samuelson 42, Berkowitz 84 $\mathcal{O}\left(n^4\right)$

$\Rightarrow$ $\boxed{\text{Dense, over a ring}}$

Elimination based: Danilevskii 37, Hessenberg 41, ... $\mathcal{O}\left(n^3\right)$

$\Rightarrow$ $\boxed{\text{Dense, over a field}}$

Explicit Krylov: Keller-Gehrig 85, Giesbrecht 93, ... $\mathcal{O}\left(n^\omega \log n\right)$

$\Rightarrow$ $\boxed{\text{Dense Black-box, over a field}}$

Implicit Krylov: P. & Storjohann 07, ... $\mathcal{O}\left(n^\omega\right)$

$\Rightarrow$ $\boxed{\text{Dense over a field}}$

Computing the characteristic polynomial

Overview on the main approaches:

Traces of powers: Leverrier 1881, Faddeev 59, ... $\mathcal{O}\left(n^4\right)$

$\Rightarrow$ Dense, over a ring, best in parallel

Determinant expansion: Samuelson 42, Berkowitz 84 $\mathcal{O}\left(n^4\right)$

$\Rightarrow$ Dense, over a ring

Elimination based: Danilevskii 37, Hessenberg 41, ... $\mathcal{O}\left(n^3\right)$

$\Rightarrow$ Dense, over a field

Explicit Krylov: Keller-Gehrig 85, Giesbrecht 93, ... $\mathcal{O}\left(n^\omega \log n\right)$

$\Rightarrow$ Dense Black-box, over a field

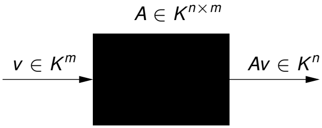
Implicit Krylov: P. & Storjohann 07, ... $\mathcal{O}\left(n^\omega\right)$

$\Rightarrow$ Dense over a field

Black Box linear algebra

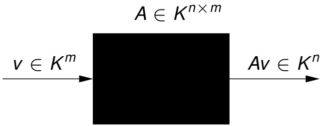
Black Box linear algebra

- Matrices viewed as linear operators
- algorithms based on matrix vector apply only $\Rightarrow$cost $E(n)$



Black Box linear algebra

- Matrices viewed as linear operators
- algorithms based on matrix vector apply only $\Rightarrow$cost $E(n)$

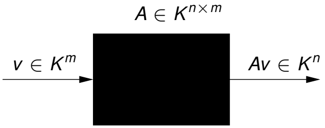


Structured matrices: Fast apply (e.g. $E(n) = \mathcal{O}(n \log n)$)

Sparse matrices: Fast apply and no fill-in

Black Box linear algebra

- Matrices viewed as linear operators
- algorithms based on matrix vector apply only $\Rightarrow$cost $E(n)$



Structured matrices: Fast apply (e.g. $E(n) = \mathcal{O}(n \log n)$)

Sparse matrices: Fast apply and no fill-in

$\Rightarrow$

- Iterative methods
- No access to coefficients, trace, no elimination
- Matrix **multiplication** $\Rightarrow$ Black-box **composition**

Black box linear algebra

Minimal polynomial: [Wiedemann 86]

$\Rightarrow$adapts numerical iterative Krylov/Lanczos methods

$\Rightarrow\mathcal{O}\left(dE(n) + n^2\right)$ operations

Rank, Det, Solve: [Kaltofen & Saunders 90, Chen& Al. 02]

$\Rightarrow$reduced to minimal polynomial and preconditioners

Black box linear algebra

Minimal polynomial: [Wiedemann 86]

$\Rightarrow$adapts numerical iterative Krylov/Lanczos methods

$\Rightarrow \mathcal{O}\left(dE(n) + n^2\right)$ operations

Rank, Det, Solve: [Kaltofen & Saunders 90, Chen& Al. 02]

$\Rightarrow$reduced to minimal polynomial and preconditioners

$\Rightarrow \mathcal{O}\tilde{}\left(nE(n)\right)$ operations

Black-box characteristic polynomial

Open Problem [Kaltofen 98 Pb. 3]

`CharPoly` in $\mathcal{O}(nE(n)) + \mathcal{O}\tilde{}(n^2)$ operations and $\mathcal{O}(n)$ memory

Black-box characteristic polynomial

Open Problem [Kaltofen 98 Pb. 3]

`CharPoly` in $\mathcal{O}(nE(n)) + \mathcal{O}^{\sim}(n^2)$ operations and $\mathcal{O}(n)$ memory

State of the art:

Eberly 2000 : adaptive in the number Φ of invariant factors $\mathcal{O}^{\sim}(nE(n) + \Phi n^2)$

Villard 2000 : adaptive in the number Ψ of distinct invariant factors $\mathcal{O}^{\sim}(n^{1.5}E(n))$

Black-box characteristic polynomial

Open Problem [Kaltofen 98 Pb. 3]

`CharPoly` in $\mathcal{O}(nE(n)) + \mathcal{O}\tilde{}(n^2)$ operations and $\mathcal{O}(n)$ memory

State of the art:

Eberly 2000 : adaptive in the number Φ of invariant factors $\mathcal{O}\tilde{}(nE(n) + \Phi n^2)$

Villard 2000 : adaptive in the number Ψ of distinct invariant factors $\mathcal{O}\tilde{}(n^{1.5}E(n))$

Present Contribution:

- algorithms and heuristics efficient in practice
- improving the best complexity by a log n factor, under a conjectured property

Outline

Outline

1. Computing multiplicities
2. Hybrid algorithms
3. Experiments

Towards a fast heuristic:

$$\text{MinPoly} \mid \text{Charpoly}$$

⇒only differ in the multiplicities of irreducible factors.

$$\begin{aligned} MinPoly &= \prod_i P_i^{e_i} \\ CharPoly &= \prod_i P_i^{m_i} \text{ with } m_i \geq e_i, d_i = \deg P_i \end{aligned}$$

Towards a fast heuristic:

MinPoly|Charpoly

$\Rightarrow$only differ in the multiplicities of irreducible factors.

$$\begin{aligned} MinPoly &= \prod_i P_i^{e_i} \\ CharPoly &= \prod_i P_i^{m_i} \text{ with } m_i \geq e_i, d_i = \deg P_i \end{aligned}$$

- compute MinPoly
- Factor it
- Determine the multiplicities m_i

The method of the nullities

Definition

Companion matrix of $P = X^n - a_{n-1}X^{n-1} - \cdots - a_0$*:*

$$C_P = \begin{bmatrix} 0 & & & a_0 \\ 1 & & & a_1 \\ & \ddots & & \vdots \\ & & 1 & a_{n-1} \end{bmatrix}$$

The method of the nullities

Definition

Companion matrix of
$P = X^n - a_{n-1}X^{n-1} - \cdots - a_0$:

$$C_P = \begin{bmatrix} 0 & & & a_0 \\ 1 & & & a_1 \\ & \ddots & & \vdots \\ & & 1 & a_{n-1} \end{bmatrix}$$

Property

$$P(C_P) = 0$$

The method of the nullities

Definition

Companion matrix of $P = X^n - a_{n-1}X^{n-1} - \cdots - a_0$:

$$C_P = \begin{bmatrix} 0 & & & a_0 \\ 1 & & & a_1 \\ & \ddots & & \vdots \\ & & 1 & a_{n-1} \end{bmatrix}$$

Property

$$P(C_P) = 0$$

$$P_1\left(\begin{bmatrix} C_{P_0} & \\ & C_{P_1} \end{bmatrix}\right) = \left(\begin{bmatrix} * & \\ & 0 \end{bmatrix}\right)$$

$\Rightarrow nullity = deg(P_1)$

The method of the nullities

Definition

Companion matrix of $P = X^n - a_{n-1}X^{n-1} - \cdots - a_0$:

$$C_P = \begin{bmatrix} 0 & & & a_0 \\ 1 & & & a_1 \\ & \ddots & & \vdots \\ & & 1 & a_{n-1} \end{bmatrix}$$

Property

$$P(C_P) = 0$$

$$P_1\left(\begin{bmatrix} C_{P_0} & \\ & C_{P_1} \end{bmatrix}\right) = \left(\begin{bmatrix} * & \\ & 0 \end{bmatrix}\right)$$

$\Rightarrow$ *nullity* $= deg(P_1)$

Property

$$\begin{aligned} P(C_{P^k}) &\equiv P\left(\begin{bmatrix} C_P & 1 & & \\ & & \ddots & \\ & & & 1 \\ & & & C_P \end{bmatrix}\right) \\ &= \begin{bmatrix} 0 & 1 & & \\ & & \ddots & 1 \\ & & & 0 \end{bmatrix} \end{aligned}$$

$\Rightarrow$

nullity $(P(C_{P^k})) = n-(k-1)deg(P)$

nullity $(P^k(C_{P^k})) = n$

The method of the nullities

Theorem

$$nullity\,(P_i^{e_i}(A)) = m_i d_i$$

$$\Rightarrow m_i = \left(\frac{n - rank(P_i^{e_i}(A))}{d_i}\right)$$

The method of the nullities

Theorem

$$nullity\,(P_i^{e_i}(A)) = m_i d_i$$

$$\Rightarrow m_i = \left(\frac{n - rank(P_i^{e_i}(A))}{d_i}\right)$$

$$\left[\begin{array}{cc|cc|c} C_{(X+1)^3} & & & & \\ & C_{(X+2)^5} & & & \\ \hline & & C_{(X+1)^2} & & \\ & & & C_{X+2} & \\ \hline & & & & C_{X+1} \end{array}\right] \xrightarrow{(X+1)^3} \left[\begin{array}{cc|cc|c} 0 & & & & \\ & C_{(X+2)^5} & & & \\ \hline & & 0 & & \\ & & & C_{X+2} & \\ \hline & & & & 0 \end{array}\right]$$

The method of the nullities

Theorem

$$\textit{nullity}\left(P_i^{e_i}(A)\right) = m_i d_i$$

$$\Rightarrow m_i = \left(\frac{n - \textit{rank}(P_i^{e_i}(A))}{d_i}\right)$$

$$\left[\begin{array}{cc|cc|c} C_{(X+1)^3} & & & & \\ & C_{(X+2)^5} & & & \\ \hline & & C_{(X+1)^2} & & \\ & & & C_{X+2} & \\ \hline & & & & C_{X+1} \end{array}\right] \xrightarrow{(X+1)^3} \left[\begin{array}{cc|cc|c} 0 & & & & \\ & C_{(X+2)^5} & & & \\ \hline & & 0 & & \\ & & & C_{X+2} & \\ \hline & & & & 0 \end{array}\right]$$

Characteristics of the method

Cost : $\mathcal{O}\left(e_i d_i E(n)\right)$

- only for small factors, with small multiplicity e_i
- still possible to get partial information applying powers $k < e_i$ of P_i to A.

The combinatorial search method

Total degree equation:

$$\sum_i d_i m_i = n$$

- Integer programming problem
- *Branch & Bound* strategy
  - incrementally increase the multiplicity of each factor
  - list every admissible candidate
- Several candidates are possible $\Rightarrow$discriminate them one evaluation at a random value: CharPoly$\lambda_0 = \det(\lambda_0 I - A)$

The combinatorial search method

Total degree equation:

$$\sum_i d_i m_i = n$$

- Integer programming problem
- *Branch & Bound* strategy
  - incrementally increase the multiplicity of each factor
  - list every admissible candidate
- Several candidates are possible $\Rightarrow$discriminate them one evaluation at a random value: $\text{CharPoly}\lambda_0 = \det(\lambda_0 I - A)$

Characteristics of the method

- Mostly efficient with factors of large degree d_i
- exponential complexity
  $\Rightarrow$Experimentally: limited to the 5 largest factors

The index calculus method

$$\prod_{j=1}^{k} P_j(\lambda)^{m_j} = \det(\lambda I - A) \mod p$$

The index calculus method

$$\prod_{j=1}^{k} P_j(\lambda)^{m_j} = \det(\lambda I - A) \mod p$$

$$\sum_{j=1}^{k} \log_g(P_j(\lambda)) m_j = \log_g(\det(\lambda I - A)) \mod (p-1)$$

Taking t λ_i's at random: $\Rightarrow t \times k$ linear system in the m_j

$$\begin{bmatrix} \log_g P_1(\lambda_1) & \dots & \log_g P_k(\lambda_1) \\ \vdots & & \vdots \\ \log_g P_1(\lambda_t) & \dots & \log_g P_k(\lambda_t) \end{bmatrix} \begin{bmatrix} m_1 \\ \vdots \\ m_k \end{bmatrix} = \begin{bmatrix} \log_g \det(\lambda_1 I - A)) \\ \vdots \\ \log_g \det(\lambda_t I - A)) \end{bmatrix}$$

The index calculus method

$$\prod_{j=1}^{k} P_j(\lambda)^{m_j} = \det(\lambda I - A) \mod p$$

$$\sum_{j=1}^{k} \log_g(P_j(\lambda)) m_j = \log_g(\det(\lambda I - A)) \mod (p-1)$$

Taking t λ_i's at random: $\Rightarrow t \times k$ linear system in the m_j

$$\begin{bmatrix} \log_g P_1(\lambda_1) & \dots & \log_g P_k(\lambda_1) \\ \vdots & & \vdots \\ \log_g P_1(\lambda_t) & \dots & \log_g P_k(\lambda_t) \end{bmatrix} \begin{bmatrix} m_1 \\ \vdots \\ m_k \end{bmatrix} = \begin{bmatrix} \log_g \det(\lambda_1 I - A)) \\ \vdots \\ \log_g \det(\lambda_t I - A)) \end{bmatrix}$$

- If non singular: the unique solution gives the m_i.
- Conjecture: the system is *likely* to be be non singular

Outline

Hybrid algorithms

Combination according to the predilection domain:

$d_i e_i$ small : nullities

d_i large : combinatorial search

remaining cases : index calculus

Hybrid algorithms

Combination according to the predilection domain:

$d_i e_i$ small : nullities

d_i large : combinatorial search

remaining cases : index calculus

Improvement

Multiple candidates treated as multiple RHS for the index calculus system

Further improvements

Villard 2000: $MinPoly(A + UV)$, s.t. $\text{rank}(UV) = k$
$\Rightarrow$ k-th invariant factor
$\Rightarrow$ partial information on the multiplicities

Further improvements

Villard 2000: $MinPoly(A + UV)$, s.t. $\text{rank}(UV) = k$
$\Rightarrow$ k-th invariant factor
$\Rightarrow$ partial information on the multiplicities

Combined with index calculus:

- compute the largest invariant factors by decreasing order
- until only $\sqrt{n}$ unknown multiplicities remain
- solve a $\sqrt{n} \times \sqrt{n}$ index calculus system

Further improvements

Villard 2000: $MinPoly(A + UV)$, s.t. $\text{rank}(UV) = k$
$\Rightarrow$ k-th invariant factor
$\Rightarrow$ partial information on the multiplicities

Combined with index calculus:

- compute the largest invariant factors by decreasing order
- until only $\sqrt{n}$ unknown multiplicities remain
- solve a $\sqrt{n} \times \sqrt{n}$ index calculus system

$\Rightarrow$ at most $\sqrt{n}$ invariant factors computed
$\Rightarrow$ $\mathcal{O}\left(n^{1.5}E(n)\right)$, (saving a log factor)

But non singularity is only conjectured

Computing Charpoly over $\mathbb{Z}$

- compute P_M = MinPoly over $\mathbb{Z}$
- decompose it into irreducible factors P_i
- pick a prime p at random
- compute P_C = CharPoly mod p
- compute the multiplicities of the $P_i \mod p$ in P_C.

Outline

Experiments

Matrix	EX1	EX2	EX3	EX4	EX5
n: dimension	560	560	2600	2600	6545
d: deg ($P_{\min}$)	54	103	1036	1552	2874
ω: sparsity	15.6	15.6	27.6	27.6	45.2
$\mathbb{Z}$-`Minpoly`	0.11s	0.26s	117s	260s	5002s
$\mathbb{Z}[X]$ factorize	0.02s	0.07s	9.4	18.15	74.09s
Nullity/comb.	3.37s	5.33s	33.2s	30.15s	289s
Total	3.51s	5.66s	159.4s	308.1s	5366s
Index calc.	3.46s	4.31s	64.0s	57.0s	647s
Total	3.59s	4.64s	190.4s	336.4s	5641s

Pentium4 (x86 3.2 GHz; 1 Gb)

Experiments

Matrix	n	ω	dense	null-comb	index
A	300	1.9	0.32s	0.08s	0.07s
AA^T	300	2.95	0.81s	0.12s	0.12s
B	600	4	4.4s	1.52s	1.97s
BB^T	600	13	2.15s	3.96	7.48s
TF12	552	7.6	6.8s	5.53s	5.75s
mk9b3	1260	3	31.25s	10.51s	177s
Tref500	500	16.9	65.14s	25.14s	25.17s

Athlon (1.8 GHz; 2 Gb)

`dense:` BB $\mathbb{Z}$-Minpoly + 1 dense charpoly mod p
`null-comb:` BB $\mathbb{Z}$-Minpoly + nullities & Comb. search
`index:` BB $\mathbb{Z}$-Minpoly + index calculus

Perspectives

- Conjectured behaviour of the index calculus
- Can these computations provide a certificate for the MinPoly (Wiedemann algorithm is only Monte-Carlo) ?
- Block-Wiedemann techniques for computing the k-th invariant factor

Perspectives

- Conjectured behaviour of the index calculus
- Can these computations provide a certificate for the MinPoly (Wiedemann algorithm is only Monte-Carlo) ?
- Block-Wiedemann techniques for computing the k-th invariant factor

Thank you