MPRI – Cours 2.12.2

F. Morain

Linear algebra for IF and DLP

2019/10/15

The slides are available on http://www.lix.polytechnique.fr/Labo/Francois.Morain/MPRI/2019

I. Introduction

M is $n \times n$ and contains relations, generally a row is a relation:

- IF: solve $X \cdot M = 0$ over $\mathbb{F}_2$;
- DLP: (Ordinary) right kernel: $M \cdot X = 0$ over $\mathbb{F}_\ell$.

Traditional elimination:

- Gauß: if enough memory available, $O(n^3)$.
- Structured Gaussian Elimination (SGE) in order to reduce the size of the sparse input matrix.

Final system:

- (Gauß: if enough memory available.)
- Algorithms for sparse matrices (plain or block versions): Krylov-based methods as
  - Wiedemann: $O(n^{2+\varepsilon})$; easier to analyze.
  - Lanczos: $O(n^{2+\varepsilon})$.

Examples of IF matrices

Fundamental property: combination matrices are sparse, since $\Omega(N) \leq \log_2 N$.

N	size	#coeffs $\neq 0$ per relation
RSA-100	$50,000 \times 50,000$	
RSA-110	$80,000 \times 80,000$	
RSA-120	$252,222 \times 245,810$ $(89,304 \times 89,088)$	
RSA-129	$569,466 \times 524,338$ $(188,614 \times 188,160)$	47
RSA-130	$3,504,823 \times 3,516,502$	39
RSA-140	$4,671,181 \times 4,704,451$	32
RSA-155	$6,699,191 \times 6,711,336$	62
$6^{353}-1$	$19,591,108 \times 19,590,832$	229
RSA-768	$192,796,550 \times 192,795,550$	144

Examples of DL matrices

For $\mathbb{F}_p$ and $\ell = (p-1)/2$ prime; annoucer on NMBRTHRY only.

size p	who / when	orig $\rightarrow$ SGE (row density)
90dd	1998(Lercier)	976062×674564 $\rightarrow 61136 \times 61036\,(93)$
110dd	2001(Lercier)	1437324×767381 $\rightarrow 160329 \times 160224\,(85)$
120dd	2001(Lercier)	2685597×1242551 $\rightarrow 271654 \times 271552\,(84)$
130dd	2005(Joux)	1946916×1466090 $\rightarrow 433181 \times 432172\,(82)$
135dd	2006(Matyukhin)	2541033×2739345 $\rightarrow 564000 \times 565000\,(573)$
160dd	2007(Kleinjung)	423671492 rels $\rightarrow 2177226 \times 2177026\,(133)$
180dd (596b)	2014(Jeljeli)	$175M$ rels $\rightarrow 7.28M(150)$
768b	2016(Kleinjung)	$\rightarrow 24M(135)$

II. Linear algebra for IF

Solving $X \cdot M = 0$

A) Gauß

$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & & & \\ 0 & 1 & & \\ 0 & 0 & 1 & \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & & & \\ 0 & 1 & & \\ 0 & 0 & 1 & \\ 0 & 1 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ x & x & x & x \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & & & \\ 0 & 1 & & \\ 1 & 0 & 1 & \\ 1 & 1 & 0 & 1 \end{pmatrix} \begin{matrix} \\ \\ L_1 + L_3 = 0 \\ L_1 + L_2 + L_4 = 0 \end{matrix}$$

Implementation remarks

- The companion matrix can be merged into M.
- additings rows use XOR's on `unsigned long` (in C). Still in $O(k^3)$ but with a very small constant.
- **Problem:** Gauß cannot be distributed easily.

B) Structured Gaussian Elimination (SGE)

(Pomerance/Smith, Odlyzko/LaMacchia, etc.)

Principle: perform the ordinary gaussian process keeping the matrix as sparse as possible as long as possible!
Works for any finite field, $\mathbb{F}_2$ somewhat easier.

Iterate:

- remove columns of weight 0;
- remove columns of weight 1;
- remove columns of weight $w \leq w_{\max}$;
- remove some heavy rows (keeping $n' \geq m'$).

Rem. Many variants possible; usually $w_{\max}$ is small.

Fine points:

- data structures: wr, wc;
- heap for a selection rule (Markowicz count, etc.); usually $|M_{i,j}| = 1$ or $|M_{i,j}| \leq c$.

C) Wiedemann's algorithm in a nutshell

M is $n \times n$

f^M: minimal polynomial of M, and/or that of $\{M^i\}_{i=0}^{\infty}$.

$f^{M,b}$: minimal polynomial of $\{M^i b\}_{i=0}^{\infty}$; of course $f^{M,b} \mid f^M$.

If u is any row vector, the minimal polynomial of $\{uM^i b\}_{i=0}^{\infty}$ is $f_u^{M,b}$ and $f_u^{M,b} \mid f^{M,b}$.

Rem. for random b, $f^{M,b} = f^M$ with high probability.

Idea: compute $f_u^{M,b}$ using $\{uM^i b\}_{i=0}^{2n-1}$ and use Berlekamp-Massey in time $O(n^2)$ (or faster $O(\mathsf{M}(n) \log n)$).

Cost: $2n$ applications Mb (black-box operation). If M has $n^{1+\varepsilon}$ non-zero coeffs, then this is $O(n^{2+\varepsilon})$.

Proofs: Kaltofen, etc.

Application to IF

Trick: find minimal polynomial of z for random z. Then (probably)

$$P_M(X) = X + p_2X^2 + \cdots + p_rX^r$$

and $P_M(M)(z) = 0 = M(z + p_2Mz + \cdots + p_rM^{r-1}z)$, so that we probably have an element of the kernel.

For our example: $P_M(X) = X^2 + X^3$; put $N = M + M^2$: $v = (1111)^t$, $w = N \cdot v = (1101)^t$, $M \cdot w = 0$.

Distributed versions:

- cut M into slices, evaluate Mz this way.
- block version (Coppersmith): operate on block X of 64 vectors. Generalize Berlekamp-Massey to this case (see E. Thomé articles).

Application to DLP

Goal: find (all?) solutions of $Mx = b$ where M is a sparse $n \times m$ matrix, b is $m \times 1$, $n \gg m$, $\mathrm{rank}(M) = r$ is unknown (close to m?).

If M is non-singular

$$f^{M,b}(X) = c_0 + c_1X + \cdots c_{m-1}X^{m-1} + X^m$$

with $c_0 \in K^*$, from which

$$x = -\frac{1}{c_0}\left(M^{m-1}b + c_{m-1}M^{m-2}b + \cdots + c_1b\right).$$

If M is singular, use randomization.

Randomization to save the world

Thm. Let M be of rank r and u_i, w_i selected randomly and uniformly from S in

$$\tilde{M} = UML, \quad U = \begin{pmatrix} 1 & u_2 & u_3 & \cdots & u_n \\ & 1 & u_2 & \cdots & u_{n-1} \\ & & 1 & \ddots & \vdots \\ & & & \ddots & u_2 \\ & & & & 1 \end{pmatrix}, \quad L = \begin{pmatrix} 1 & & & & \\ w_2 & 1 & & & \\ w_3 & w_2 & 1 & & \\ \vdots & & & \ddots & \\ w_n & w_{n-1} & \cdots & w_2 & 1 \end{pmatrix}$$

Then

$$\mathrm{Prob}(\mathrm{Det}(\tilde{M}_i) \neq 0, 1 \leq i \leq r\} \geq 1 - \frac{r(r+1)}{\#S}.$$

(generic rank profile)

Solve $\tilde{M}\tilde{x} = Ub$, from which $x = L\tilde{x}$ solves $Mx = b$.

Product Toeplitz times vector

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} w_1 & & & & \\ w_2 & w_1 & & & \\ w_3 & w_2 & w_1 & & \\ \vdots & & & \ddots & \\ w_n & w_{n-1} & \cdots & w_2 & w_1 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \\ v_n \end{pmatrix}$$

using

$$(w_1 + \cdots + w_nz^{n-1})(v_1 + \cdots + v_nz^{n-1}) \equiv y_1 + y_2z + \cdots + y_nz^{n-1} \bmod z^n.$$

The story for RSA-768

(from article in CRYPTO 2010)

- 64 334 489 730 relations, 150 bytes each compressed to 5 terabytes of disk space;
- 27.4% duplicates; add 57 223 462 free relations;
- $\rightarrow$ 47 762 243 404 relations involving 35 288 334 017 prime ideals;
- iteratively remove singletons: 24 615 168 385 relations involving at most 9 976 671 468 prime ideals;
- clique removal: 2 458 248 361 relations with 697 618 199 prime ideals;
- SGE: 192 796 550 $\times$ 192 795 550 of total weight 27 797 115 920 (on average 144 non-zeros per row) $\rightarrow$ 105 GB;
- Block Wiedemann: 119 days calendar time; 3 spots (EPFL, LORIA, NTT).