Matrix Input and Toeplitz Determinant

Undergraduate Research Projects with LinBox

Daniel S. Roche

University of Waterloo

December 8, 2006

Outline

1. Overview of LinBox
2. Matrix Input
   - The Problem
   - The Solution: `MatrixStream`
3. Toeplitz Determinant
   - Motivation
   - Methods
   - Implementation
   - Results
4. Conclusion

General aspects of LinBox

LinBox is a high-performance C++ library for exact computational linear algebra over the integers and finite fields.

(www.linalg.org)

- Large project involving multiple universitites in the U.S., Canada, and France
- Originally based around black box algorithms for sparce matrix computation
- Now includes many algorithms for more general matrix problems.

Typical problems to be solved with LinBox

- Solve $Ax = B$
- Rank
- Determinant
- Minpoly
- Smith form

Solutions are usually found over $\mathbb{Z}$, $\mathbb{Q}$, GF(q), or GF(q^e).

So many formats, so little time

There are a plethora of different formats for representing matrix data.

- Standardized (e.g. MatrixMarket, XML)
- Specialized (e.g. Sparse-Row, SMS)
- Computer algebra software formats (e.g. Maple, Magma, Mathematica, Matlab, etc.)

Example 1: MatrixMarket

$$A = \begin{bmatrix} 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 4 \end{bmatrix}$$

Example (MatrixMarket Sparse Format)

```
%%MatrixMarket matrix coordinate integer general
% Comments
% ...
2 4 3
1  2  2
2  3  3
2  4  4
```

Example 2: Sparse-Row

$$A = \begin{bmatrix} 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 4 \end{bmatrix}$$

Example (Sparse-Row Input Format)

```
2 4
1   1 2
2   2 3   3 4
```

Example 3: Standard Maple

$$A = \begin{bmatrix} 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 4 \end{bmatrix}$$

Example (Maple Dense Matrix Declaration Format)

```
A := Matrix(2, 4, [[0,2,0,0],[0,0,3,4]],
datatype = anything, storage = rectangular,
order = Fortran_order, shape = []);
```

Internal Representations

- LinBox has many different classes for holding matrix data in memory
- Choice of internal representation should be based on matrix *content* and the choice of the algorithm, not on *format*
- Prior to this work, each internal representation class had its own set of readers and supported external formats.

Requirements

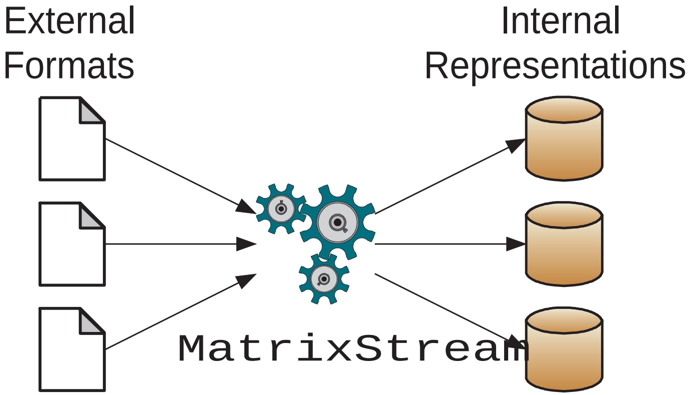
Automatic Recognition It would be too cumbersome to define a way for the user to specify the format of an input matrix, and filename extensions are unreliable.

Robustness This one tool must be able to handle all different types of input without crashing.

Extensibility It is conceivable and in fact highly likely that more formats will be added in the future, and this should be as simple as possible.

Stream Input Matrix input could come from the web or some other arbitrary source and not a file. Also, one file or input source could contain multiple matrices.

General Idea



Implementation: MatrixStreamReaders

Each format known to the MatrixStream has a corresponding implementation of the MatrixStreamReader abstract class.

- Readers are given small chunks of the data to process.
- Can be viewed as a competition where each Reader "drops out" when it is unable to read the given file.
- Each Reader should quickly recognize whether the given file is in its supported format.

Implementation: Matrix data interface

Simple, generic interface to the internal matrix representations:

`nextTriple` Gives the row, column, and value of
the next entry in the matrix
(order is unspecified)

`dimensions` Gives the dimensions of the matrix (rows, columns)

`isSparse`, `isDense`, etc. Gives some indication of the structure or sparsity of the input matrix, if these indications are given in the matrix file format.

Toeplitz matrices

Definition

A Toepliz matrix is of the following form:

$$T_n = \begin{bmatrix} t_{n-1} & t_n & \cdots & t_{2n-3} & t_{2n-2} \\ t_{n-2} & \ddots & \ddots & & t_{2n-3} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ t_1 & & \ddots & \ddots & t_n \\ t_0 & t_1 & \cdots & t_{n-2} & t_{n-1} \end{bmatrix}$$

So a Toeplitz matrix is sparse in information
but dense in structure.

The need for fast Toeplitz determinant

Robert Chapman (University of Exeter) - 2004

Conjecture

The determinants over $\mathbb{Z}$ of a certain infinite class of Toeplitz matrices formed from Legendre symbols $\left(\frac{a}{p}\right)$ are all 1.

Goal: Find a counterexample to this conjecture.

Toeplitz determinant

Problem statement

Given: Toeplitz matrix T_n over $\mathbb{Z}$
Find: $\det(T_n)$

- Dense methods take $O(n^2)$ space and $O(n^3)$ time.
- We know that T_n can be stored in $O(n)$ space
- Can the running time be improved?

Subresultants

- Let D be a unique factorization domain (for our purposes, either $\mathbb{Z}$ or some field)
- Let $f_1, f_2 \in D[x]$

Definition

The j^{th} *subresultant* of f_1 and f_2 is denoted $S_j(f_1, f_2)$. If we write

$$\begin{aligned} f_1 &= a_0 + a_1 x + a_2 x^2 + \cdots + a_m x^k \\ f_2 &= b_0 + b_1 x + b_2 x^2 + \cdots + b_n x^l \end{aligned}$$

Then $S_j(f_1, f_2)$ is the determinant of the $(k + l - 2j) \times (k + l - 2j)$ matrix given on the following page.

Subresultant matrix

$$S_j(f_1, f_2) = \begin{vmatrix}
a_k & \cdots & & a_0 & & & x^{l-j-1} f_1 \\
 & & & & \ddots & & \\
 & \ddots & & & & a_0 & \vdots \\
 & & & & & \vdots & \\
 & a_k & & \cdots & & a_{j+1} & f_1 \\
b_l & \cdots & b_0 & & & & x^{k-j-1} f_2 \\
 & & & & \ddots & & \\
 & \ddots & & & & b_0 & \vdots \\
 & & & & & \vdots & \\
 & & b_l & \cdots & & b_{j+1} & f_2
\end{vmatrix}$$

Subresultant of Toeplitz polynomial

Let T_n be a Toeplitz matrix with entries t_0 to t_{2n-2}.
So t_0 is at the bottom-right corner,
t_{n-1} is on the main diagonal,
and t_{2n-2} is at the top-right corner.
Define:

$$\begin{aligned} f_1 &= x^{2n-1} \\ f_2 &= t_0 + t_1 x + \cdots + t_{2n-2} x^{2n-2} \end{aligned}$$

Then the $(n-1)^{\mathrm{th}}$ subresultant of f_1 and f_2 is ...

Subresultant of Toeplitz polynomial

$$S_{n-1}(f_1, f_2) = \begin{vmatrix} 1 & & & & & & x^{3n-3} \\ & \ddots & & & & & \vdots \\ & & 1 & & & & x^{2n-1} \\ t_{2n-2} & \cdots & t_n & t_{n-1} & \cdots & t_1 & x^{n-1} f_2 \\ & \ddots & & & & \vdots & \vdots \\ & & & t_{2n-2} & \cdots & t_n & f_2 \end{vmatrix}$$

Examining the coefficients of x_i in the subresultant,
we see that the the coefficients are all 0 for $i > n-1$,
and the coefficient of x^{n-1} is given by ...

Subresultant of Toeplitz polynomial

$$
\begin{aligned}
\mathrm{lcoeff}(S_{n-1}(f_1, f_2)) &= \begin{vmatrix}
1 & & & & & & 0 \\
 & \ddots & & & & & \vdots \\
 & & 1 & & & & 0 \\
t_{2n-2} & \cdots & t_n & t_{n-1} & \cdots & t_1 & t_0 \\
 & \ddots & & & & \vdots & \vdots \\
 & & & t_{2n-2} & \cdots & t_n & t_{n-1}
\end{vmatrix} \\
&= \det(T_n^T) \\
&= \det(T_n)
\end{aligned}
$$

Polynomial Remainder Sequences

So $\mathrm{lcoeff}(S_{n-1}(f_1, f_2)) = \det(T_n)$. [Kaltofen and Lobo 1996]
Now for each $i \geq 3$, define $f_i, q_i \in D[x]$ and $\alpha_i \in D$ such that:

$$f_i = \alpha_i f_{i-2} - q_i f_{i-1},$$

where $\deg(f_i) < \deg(f_{i-1})$.
Then the sequence $(f_1, f_2, f_3, \ldots)$ is called a *polynomial remainder sequence* or PRS.
Let n_i and c_i be the degree and leading coefficient (respectively) of f_i.

Correlation between subresultants and PRS

Let m be the least integer such that n_m (the degree of f_m) is less than n. Then the following is a simple consequence from the "Fundamental Theorem" in [Brown and Traub 1971]

Theorem

$$S_{n-1}(f_1, f_2) = \begin{cases} f_m c_m^{n_{m-1}-n_m-1} \prod_{j=3}^{m} \dfrac{c_{j-1}^{n_{j-2}-n_j}(-1)^{(n_{j-2}-n_m)(n_{j-1}-n_m)}}{\alpha_j^{n_{j-1}-n_m}}, & n_m = n-1 \\ 0, & n_m < n-1 \end{cases}$$

Computing $\det(T_n)$

Then, since we know $S_{n-1}(f_1, f_2) = \det(T_n)$, this gives a way to compute the determinant of a Toeplitz matrix from half of the PRS of f_1 and f_2.

- Only need the leading coefficient of $S_{n-1}(f_1, f_2)$
- Can be found by computing the PRS $(f_1, f_2, \ldots, f_m)$
- No need to store the whole PRS in memory at the same time
- If D is a field, then we have exact division, so each $\alpha_i = 1$, and we can ignore them.

The algorithm — Part 1

INPUT: Toeplitz matrix T_n with entries
$t_0, t_1, \ldots, t_{2n-2}$ in the field F

OUTPUT: Determinant of T_n over F

f_{i-2} <- x^{2n-1}
f_{i-1} <- $t_0 + t_1 x + \cdots + t_{2n-2} x^{2n-2}$
det <- 1
sign <- 1

The algorithm — Part 2

```
while deg(f_{i-2}) ≥ n do
   f_i <- f_{i-2} mod f_{i-1}
   det <- det * lcoeff(f_{i-1})^(deg(f_{i-2})-deg(f_i))
   if (deg(f_{i-2}) - n) and (deg(f_{i-1}) - n) both even
      sign <- sign * -1
   t_{i-2} <- t_{i-1}
   t_{i-1} <- t_i
end while
if deg(f_{i-2}) < n-1 then return 0
else return det
```

Implementation details

- Implemented in LinBox
- NTL library used for polynomial operations
- Finite field and integer versions

Asymptotic analysis

Time complexity:

- At most n iterations through the loop
- Each iteration involves polynomial division taking $O(n)$ time
- Total running time is $O(n^2)$

Space complexity:

- Just store 3 polynomials and 2 scalars
- Total space is $O(n)$

Conjecture testing

- Tested a larger class of matrices than previously possible
- Testing performed over finite field to speed computation
- No counterexamples found
- Running time improvement only seen for large matrices

References

Richard P. Brent, Fred G. Gustavson, and David Y. Y. Yun.
Fast solution of Toeplitz systems of equations and computation of Padé approximants.
J. Algorithms, 1(3):259–295, 1980.

W. S. Brown and J. F. Traub.
On Euclid's algorithm and the theory of subresultants.
J. Assoc. Comput. Mach., 18:505–514, 1971.

Robin Chapman.
Determinants of Legendre symbol matrices.
Acta Arith., 115(3):231–244, 2004.

Erich Kaltofen and Austin Lobo.
On rank properties of toeplitz matrices over finite fields.
In *ISSAC 1996*, pages 241–249. ACM, New York, 1996.

Other projects with LinBox

- Benchmarking for finite field representations
- LinBox web computation server
- Block methods for rank, determinant, etc. (total failure)

Current interests

- Symbolic-numeric matrix computations
  (e.g. using numeric techniques or subroutines)
- Structured matrix computations
  (including matrices symbolic in size?)