Determinants

Lecture 11

Computing Determinants comes up frequently

Numerous calculations can be phrased as matrix determinant calculations

Eigenvalues

Resultants

Solution of linear systems

This doesn't mean you SHOULD compute these by determinants necessarily...

Sparse Matrices

A numerical matrix of size n X m is sparse if it has k << mn non-zero terms.

A symbolic matrix could either be numerically sparse or could have many terms where each term is a sparse polynomial.

The most extreme sparse matrix in this second sense would be one in which each entry is a distinct term that combines with nothing else. e.g.

Sparse Matrices

A numerical matrix of size n X m is sparse if it has k << mn non-zero terms.

A symbolic matrix could either be numerically sparse or could have many terms where each term is a sparse polynomial.

The most extreme sparse matrix in this second sense would be one in which each entry is a distinct term that combines with nothing else. e.g. 3X4

$$\begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} \end{pmatrix}$$

How large is a sparse NxN determinant?

N! terms. Each term is size N.

For example, the generic 3x3 matrix has determinant $a_{1,1}a_{2,2}a_{3,3}-a_{1,2}a_{2,1}a_{3,3}-a_{1,1}a_{2,3}a_{3,2}+a_{1,3}a_{2,1}a_{3,2}+a_{1,2}a_{2,3}a_{3,1}-a_{1,3}a_{2,2}a_{3,1}$.

with, as expected, six terms

How large is a sparse NxN determinant?

and since we are using a computer to generate this, we can show you the 24 terms of the 4x4 determinant:

$a_{1,1}a_{2,2}a_{3,3}a_{4,4}-a_{1,2}a_{2,1}a_{3,3}a_{4,4}-a_{1,1}a_{2,3}a_{3,2}a_{4,4}+a_{1,3}a_{2,1}a_{3,2}a_{4,4}+a_{1,2}a_{2,3}a_{3,1}a_{4,4}-a_{1,3}a_{2,2}a_{3,1}a_{4,4}-a_{1,1}a_{2,2}a_{3,4}a_{4,3}+a_{1,2}a_{2,1}a_{3,4}a_{4,3}+a_{1,1}a_{2,4}a_{3,2}a_{4,3}-a_{1,4}a_{2,1}a_{3,2}a_{4,3}-a_{1,2}a_{2,4}a_{3,1}a_{4,3}+a_{1,4}a_{2,2}a_{3,1}a_{4,3}+a_{1,1}a_{2,3}a_{3,4}a_{4,2}-a_{1,3}a_{2,1}a_{3,4}a_{4,2}-a_{1,1}a_{2,4}a_{3,3}a_{4,2}+a_{1,4}a_{2,1}a_{3,3}a_{4,2}+a_{1,3}a_{2,4}a_{3,1}a_{4,2}-a_{1,4}a_{2,3}a_{3,1}a_{4,2}-a_{1,2}a_{2,3}a_{3,4}a_{4,1}+a_{1,3}a_{2,2}a_{3,4}a_{4,1}+a_{1,2}a_{2,4}a_{3,3}a_{4,1}-a_{1,4}a_{2,2}a_{3,3}a_{4,1}-a_{1,3}a_{2,4}a_{3,2}a_{4,1}+a_{1,4}a_{2,3}a_{3,2}a_{4,1}$

typeset kind of peculiarly, but not too bad.

This is actually a sum over all permutations of the indices X a coefficient which is the "sign" of the permutation

As happens with many facts about determinants, you should be able to prove that this determinant has N! terms a few different ways. Here's another...

How large is a sparse NxN determinant?

Here are the 24 terms of the 4x4 determinant expanded in minors around the first row:

$a_{1,1}(a_{2,2}(a_{3,3}a_{4,4}-a_{3,4}a_{4,3})-a_{2,3}(a_{3,2}a_{4,4}-a_{3,4}a_{4,2})+a_{2,4}(a_{3,2}a_{4,3}-a_{3,3}a_{4,2}))-a_{1,2}$
$(a_{2,1}(a_{3,3}a_{4,4}-a_{3,4}a_{4,3})-a_{2,3}(a_{3,1}a_{4,4}-a_{3,4}a_{4,1})+a_{2,4}(a_{3,1}a_{4,3}-a_{3,3}a_{4,1}))+a_{1,3}(a_{2,1}$
$(a_{3,2}a_{4,4}-a_{3,4}a_{4,2})-a_{2,2}(a_{3,1}a_{4,4}-a_{3,4}a_{4,1})+a_{2,4}(a_{3,1}a_{4,2}-a_{3,2}a_{4,1}))-a_{1,4}(a_{2,1}(a_{3,2}a_{4,3}-a_{3,3}a_{4,2})-$
$a_{2,2}(a_{3,1}a_{4,3}-a_{3,3}a_{4,1})+a_{2,3}(a_{3,1}a_{4,2}-a_{3,2}a_{4,1}))$

Which also gives us another proof there are N! terms.

Looking around for on-line summaries of the many useful and elementary properties of the determinant, I found this:

http://mathworld.wolfram.com/Determinant.html

N*N! vs O(1).

The size of any numerical (floating point) determinant of any size is a constant (one fp number).

Are there intermediate positions and complexities? How best to compute symbolic determinants? Note that SOME symbolic determinants will be small (even zero!)

One of my favorite papers is
M. Gentleman and S. Johnson,
Analysis of Algorithms, A Case Study: Determinants of Matrices with Polynomial Entries
ACM Transactions on Mathematical Software (TOMS)
Volume 2 , Issue 3 (September 1976)
[DOI] http://doi.acm.org/10.1145/355694.355696
gentleman.pdf

Two models

All elements distinct symbols (the model already discussed).

All elements are univariate polynomials dense to the same degree.

Two distinct algorithms: Gaussian elimination

The Gaussian elimination method, using exact division, is appropriate for computations over the integers; If we ignore pivoting, the algorithm consists of n -1 steps, indexed by a variable k running from 1 to n - 1. The k^{th} step involves the computation of an n -k +1 by n -k +1 matrix, which we shall call $A^{(k+l)}$; the entries will be denoted $a_{ij}^{(k+1)}$ with $k \cdot i, j \cdot n$. The original matrix is identified with $A^{(1)}$.
For each k, the entry $a_{ij}^{(k+1)}$ is computed by the formula

$$a_{ij}^{(k+1)} = (a_{kk}^{(k)}\, a_{ij}^{(k)} - a_{ik}^{(k)}\, a_{kj}^{(k)})/a_{k-1,k-1}^{(k-1)}$$

for $k + l \cdot i, j \cdot n$, where $A^{(0)}$ is taken to be 1. The division is always exact, so each a_{ij}^{k} is a polynomial and $a_{nn}^{(n)}$ is the determinant of $A^{(n)}$

Cost of Gaussian elimination / I

An analysis of this algorithm shows that each $a^{(k)}$ is a determinant (minor) of some k-by-k submatrix of the original matrix. Since we assumed all entries in the original matrix to be of the same size, we may expect that all elements in $A^{(k)}$, for a given i, are the same size.

Cost of Gaussian elimination / II

To compute $a_{ij}^{(k)}$ takes two multiplications, a subtraction, and a division. In general, the cost of multiplying two polynomials will depend on their size; in our situation the size is assumed to depend only on the order of the minor making up the element. Thus we shall use numbers C_{rs} to compute the cost of our algorithm, C_{rs} is the cost of multiplying a minor of order r by a minor of order s. Notice that $C_{1,1}$ is the cost of multiplying two elements from the original matrix. We assume also that an exact division of A by B to yield C has the same cost as a multiplication of B by C, and we ignore the costs of addition and subtraction.

We can now write the cost for Gaussian elimination in terms of the C_{rs}. To compute $a_{ij}^{(k+1)}$ requires two multiplications of cost C_{kk}, a division of cost $C_{k-1,k+1}$, and a subtraction whose cost we do not count. There are, for a given k, $(n-k)^2$ elements $a_{ij}^{(k+1)}$; so the total cost for the Gaussian elimination is

$$G = \sum_{k=1}^{n-1} (n-k)^2(2C_{kk} + C_{k-1,k+1}).$$

Cost of Gaussian elimination / II

Note that, when $C_{rs} = 1$ for all r and s, representing the familiar floating-point or fixed-length integer case, the cost becomes

$$G = 3 \sum_{k=1}^{n-1} (n-k)^2 = n^3 - \tfrac{3}{2}n^2 + \tfrac{1}{2}n.$$

Cost of Expansion by Minors

$$M = \sum_{k=2}^{n} \binom{n}{k} k C_{k-1,1} = n \sum_{k=2}^{n} \frac{(n-1)!}{(k-1)!(n-k)!} C_{k-1,1} = n \sum_{k=2}^{n} \binom{n-1}{k-1} C_{k-1,1}$$

$$= n \sum_{k=1}^{n-1} \binom{n-1}{k} C_{k1}.$$

Once again, we examine the cost when each C_{rs} is 1; we have

$$M = n \sum_{k=1}^{n-1} \binom{n-1}{k} = n(2^{n-1} - 1).$$

this goes a long way toward explaining why numerical analysts don't do this.