Computation Guidelines

Monday, October 17, 2005
11:07 AM

In performing calculations, it is best to keep #'s on the same side

e.g. e^x, $x = 0.17$

$$e^x = 1 - x^2/2 + x^3/3! + \cdots$$

$$\frac{1}{e^x} = \frac{1}{1 - x^2/2 + x^3/3! + \cdots}$$

e.g. $x^2 + bx + c = 0$

$$x_\pm = \frac{-b \pm \sqrt{b^2 - 4c}}{2}$$

$$x^2 + 1000x = 0$$

$$x_1 = \frac{-b + \mathrm{sgn}(-b)\sqrt{b^2-4c}}{2}$$

$$x_2 = c/2 \qquad ?$$

In statistics, we often need to compute

$$s^2 = \sum_{i=1}^{n} (x_i - \bar{x})^2 , \quad , \quad \bar{x} \quad \frac{\sum x_i}{n}$$

which is often rewritten as

$$\sum (x_i^2 - 2x_i\bar{x} + \bar{x}^2)$$

$$= \sum x_i^2 - 2\bar{x}\sum x_i + n\bar{x}^2$$

$$= \sum x_i^2 - n\bar{x}^2$$

This formula is no good-- if the x_i's are all large, then we're adding a bunch of large #'s, then subtracting a large # -- of course, the advantage for this is that it is a "one-pass" algorithm, whereas the first one was a "two-pass" algorithm

Sherman-Morrison Formula

Monday, October 17, 2005
11:15 AM

Suppose we have A, A^{-1}, want to compute $(A + \mathbf{u}\mathbf{v}^T)^{-1}$

$(A + \mathbf{u}\mathbf{v}^T)^{1/2}$

without re-computing the entire inverse; or maybe we want $(A + \mathbf{u}\mathbf{v}^T)^{1/2}$ or in general $f(A + \mathbf{u}\mathbf{v}^T)$

aside: we can't do this for eigenvalues

we'd like to know how changing only one element changes things:

$$\bar{A} := A + \begin{pmatrix} \bar{a}_{11} - a_{11} & \\ & \emptyset \end{pmatrix}$$

This is the same as

$$A + \begin{pmatrix} \bar{a}_{11} - a_{11} \\ \emptyset \\ \vdots \\ \emptyset \end{pmatrix} (1, \emptyset, \emptyset \, \emptyset)$$

In perturbation theory we had $A + \varepsilon E$

Here we have $(A+\boldsymbol{u}\boldsymbol{v}^T)\boldsymbol{y} = \boldsymbol{b}$; original problem was
$A\boldsymbol{x} = \boldsymbol{b}$

$(I + A^{-1}\boldsymbol{u}\boldsymbol{v}^T)\boldsymbol{y} = A^{-1}\boldsymbol{b} = \boldsymbol{x}$; let $\boldsymbol{w} = A^{-1}\boldsymbol{u}$

So our problem is
$\boldsymbol{y} = (I + \boldsymbol{w}\boldsymbol{v}^T)^{-1}\boldsymbol{x}$

Want a matrix X satisfying $(I + \boldsymbol{w}\boldsymbol{v}^T)X = I$
$(I + \boldsymbol{w}\boldsymbol{v}^T)^{-1} = ?$

Observe that the eigenvalues of a
rank one matrix are easy to compute:
$\boldsymbol{w}\boldsymbol{v}^T\ ? = \lambda ?$ $\qquad\qquad$ $\lambda = \boldsymbol{v}^T\boldsymbol{w}, 0, \ldots, 0$
$\boldsymbol{v}^T\boldsymbol{w}\boldsymbol{v}^T\ ? = \lambda \boldsymbol{v}^T\ ?$

Matrix inversion Corresponds to reciprocating
eigenvalues, so

$X = (I + \sigma\boldsymbol{w}\boldsymbol{v}^T)$
$(I + \boldsymbol{w}\boldsymbol{v}^T)(I + \sigma\boldsymbol{w}\boldsymbol{v}^T) \stackrel{?}{=} I$

multiplying out gives

$$I + \underbrace{\sigma \vec{w}\vec{v}^T + \vec{w}\vec{v}^T + \sigma \vec{w}\vec{v}^T\vec{w}}_{}\vec{v}^T \stackrel{?}{=} I$$

need these to sum
to 0

The three terms above can be expressed as
$(\sigma + 1 + \sigma \boldsymbol{v}^T\boldsymbol{w})\boldsymbol{w}\boldsymbol{v}^T$

So $\sigma(1 + \boldsymbol{v}^T\boldsymbol{w}) = -1$

$$\sigma = \frac{-1}{1 + \vec{v}^T\vec{w}}$$

This gives our algorithm

1) Solve $A\boldsymbol{x} = \boldsymbol{b}$
2) Solve $A\boldsymbol{w} = \boldsymbol{u}$
3) $\boldsymbol{y} = (I + \boldsymbol{w}\boldsymbol{v}^T)^{-1}\boldsymbol{x}$

$= (I + \sigma\mathbf{w}\mathbf{v}^T)\mathbf{x}$
$= \mathbf{x} + \sigma(\mathbf{v}^T\mathbf{x})\mathbf{w}$,

$$\sigma = \frac{-1}{1 + v^T w}$$

The idea: first solve the problem without perturbation, then Compute with the perturbation then compute σ from that, and done

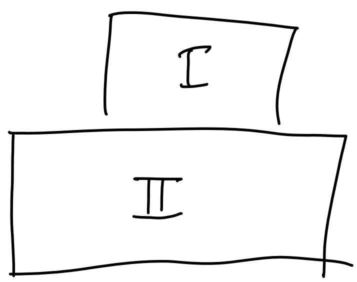
Unfortunately this is extremely prone to numerical inaccuracies, so use with caution

Domain Decomposition

Monday, October 17, 2005
11:32 AM

Over the years, people have applied separation of variables to solve Poisson equation on rectangular domains -- how

about a pair of linear domains?



A_I : linear system for I

A_{II} : " " II

$$\begin{pmatrix} A_I & \\ & A_{II} \end{pmatrix} \rightarrow \begin{pmatrix} 0 \\ 0 \\ x \\ 0 \\ 0 \end{pmatrix}$$

$0 \quad 0\,x\,x\,0 \qquad\qquad 0$

So, this matrix is really like

$$\begin{pmatrix} A_I & & 0 \\ & A_{II} & \\ 0 & & 0 \end{pmatrix} + \begin{pmatrix} & & x \\ & & x \\ & & \end{pmatrix}$$

$$\begin{pmatrix} 0 & Q \end{pmatrix} \quad \begin{pmatrix} \\ xx \end{pmatrix}$$

Called "domain decomposition". -the idea is to break the problem into subdomains, solve on them, then make corrections

Solving linear Equations

Monday, October 17, 2005
11:45 AM

$A\boldsymbol{x} = \boldsymbol{b}$, A: $n \times n$, full rank

Wart to solve the system; one idea is to factor A

$PQ\boldsymbol{x} = \boldsymbol{b}$

1) Solve $P\boldsymbol{y} = \boldsymbol{b}$

2) Solve $Q\boldsymbol{x} = \boldsymbol{y}$

Different kinds of P, Q:

$P = D = \begin{pmatrix} \diagdown & 0 \\ 0 & \diagdown \end{pmatrix}$

$d_i x_i = b_i$, $x_i = b_i / d_i$

or orthogonal:

The one we focus on most is when P is lower triangular:

$p_{11}x_1 = b_1$
$p_{21}x_1 + p_{22}x_2 = b_2$
$p_{n1}x_1 + \ldots + p_{nn}x_n = b_n$

$x_1 = b_1 / p_{11}$

$x_2 = (b_2 - p_{21} x_1) / p_{22}$

$($

This is called "back substitution" and requires *O(n²)* operations

The object is to get these Equations into simple form

Gaussian Elimination

Monday, October 17, 2005
11:52 AM

It turns out that Gaussian Elimination amounts to factorization into *LU*, lower and upper triangular

$a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1$
$a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2$
$a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n = b_n$

Provided $a_{11} \neq 0$, can multiply equation (1) by a_{21}/a_{11} and subtract from 2:

$$a_{11}x_1 + \text{—} + a_{1n}x_n = b_1$$

$$0 + a'_{22}x_2 + \text{—} + a'_{2n}x_n = b'_2$$

where $a'_{2j} = a_{2j} - \left(\frac{a_{21}}{a_{11}}\right) a_{1j}$

WC can do the same thing with the first entry of every equation:

$a_{11}x_1 + \text{—}$

$0 \qquad a'_{n2}x_n$

$\vdots$

0

how many operations? There are *(n-1)* rows and *n* columns so $n^2 - n$

We can continue the whole procedure to knock out all entries below the diagonal

$$A^{(2)} = \begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(1)} \\ 0 & \vdots & \ddots & \vdots \\ \vdots & & & \\ 0 & a_{n2}^{(2)} & \cdots & a_{nn}^{(2)} \end{pmatrix}$$

Provided $a_{ii}^{(i)} \neq 0$

It takes a total of $\sim n^3/6$ operations to get the matrix to this form, and backsubstitution takes $n^2/2$

Continued

Monday, October 17, 2005
12:03 PM

Consider a new matrix

$$M_1 = \begin{pmatrix} 1 & & & \\ -l_{21} & \ddots & & 0 \\ \vdots & & & \\ -l_{n1} & 0 & & 1 \end{pmatrix}$$

$$l_{i1} = a_{i1}/a_{11}$$

$$A_2 = M_1 A$$

$$A_3 = M_2 A_2$$

(we already computed $A_2 = A^{(2)}$)

$$M_2 = \begin{pmatrix} 1 & & & \\ 0 & 1 & & \\ \vdots & -l_{32} & \ddots & \\ \vdots & \vdots & & \\ 0 & -l_{n2} & & 1 \end{pmatrix} \qquad l_{i2} = \frac{a_{i2}^{(2)}}{a_{21}^{(2)}} \quad ?$$

In general
$A_{k+1} = M_k A_k = M_k M_{k-1} A_{k-1}$
$A_n = M_{n-1} M_{n-2} \ldots M_2 M_1 A =: U,$
U upper triangular

$$A = M_1^{-1} M_2^{-1} \cdots M_n^{-1} A_n$$

Recall that

$$M_1 = \begin{pmatrix} 1 & & & \\ -l_{21} & 1 & & 0 \\ \vdots & & \ddots & \\ -l_{n1} & 0 & & 1 \end{pmatrix}$$

Claim that

$$M_1^{-1} = \begin{pmatrix} 1 & & & \\ l_{21} & 1 & & 0 \\ \vdots & & \ddots & \\ l_{n1} & 0 & & 1 \end{pmatrix}$$

how about $M_1^{-1} M_2^{-1}$?

$$\begin{pmatrix} 1 & & \\ l_{21} & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & & \\ 0 & 1 & 0 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & & & \\ l_{21} & 1 & & 0 \\ \vdots & 0 & \ddots & \\ l_{n1} & & & 1 \end{pmatrix} \begin{pmatrix} 1 & & & \\ 0 & 1 & & 0 \\ \vdots & l_{32} & \ddots & \\ 0 & l_{n2} & & 1 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & & & 0 \\ l_{21} & 1 & & \\ l_{31} & l_{32} & \ddots & \\ \vdots & \vdots & & 1 \\ l_{n1} & l_{n2} & & \end{pmatrix}$$

?

$$M_1^{-1} M_2^{-1} \dots M_{n-1}^{-1}$$

$$= \begin{pmatrix} 1 & & 0 \\ & \ddots & \\ l_{ij} & & 1 \end{pmatrix}$$

Theorem: if $a_{ii}^{(i)} \neq 0$, then $A = LU$

det A = det(LU) = det L det U = $u_{11}u_{22}\dots u_{nn}$

Monday, October 17, 2005
12:17 PM

Note if we partition

$A = LU$

$$\left(\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right)$$

then $A_{11} = L_1 U_1$

ie this factorization is a factorization of submatrices

so $u_{jj} \neq 0$ providing $\det\begin{pmatrix} a_{11} & \cdots & a_{1j} \\ \vdots & \ddots & \\ a_{j1} & \cdots & a_{jj} \end{pmatrix}$

non zero

So, LU factorization always exists if

$\det\begin{pmatrix} a_{11} & \cdots \\ \vdots & \ddots & \vdots \\ \cdots & & a_{jj} \end{pmatrix}$ nonzero

But this would break down

$$\begin{pmatrix} 4 & 2 & 7 \\ 2 & 1 & 8 \end{pmatrix} \mapsto \begin{pmatrix} 4 & 2 & 7 \\ 0 & 0 & 4.5 \end{pmatrix}$$

$$\begin{pmatrix} 2 & 7 & 8 \\ 7 & 6 & 9 \end{pmatrix} \mapsto \begin{pmatrix} 0 & 0 & 4.5 \end{pmatrix}$$

Key to remember is that Gaussian elimination is factorization