May 9, 2014

How to compute $x = A^{-1} b$?

In numerical linear algebra, NEVER compute A^{-1} and then $A^{-1} b$. We always consider x as the solution of the equation:

$$A x = b$$

We compute x by solving the equation by Gaussian elimination.

Gaussian Elimination

Big picture of GE:

$$\underset{A}{\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix}} \to \underset{A^{(1)}}{\begin{bmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \end{bmatrix}} \to \underset{A^{(2)}}{\begin{bmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & \times & \times \end{bmatrix}} \to \underset{A^{(3)}}{\begin{bmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times \end{bmatrix}}$$

GE algorithm

for i = 1, 2, …, n-1
  for k = i+1, …, n
    mult = a_{ki} / a_{ii} (a_{ki} = mult)
    for j = i+1, …, n
      $a_{kj} = a_{kj} - \text{mult} \times a_{ij}$ } update row(k)
    end
    $b_k = b_k - \text{mult} \times b_i$ } update RHS
  end
end

At the end, $A^{(n-1)} x = b^{(n-1)}$, is solved by back substitution.

LU factorization

Theorem: A = L U where L = lower Δ, unit diag; U = upper Δ. Moreover

$$U = A^{(n-1)}, \qquad L = \begin{bmatrix} 1 & & 0 \\ & \ddots & \\ mult & & 1 \end{bmatrix}.$$

Solve A x = b

A x = b -> L U x = b

Let y = U x. Then we have L y = b.

(1) Solve L y = b by forward solve

(2) Solve U x = y by back solve

Forward solve algorithm

```
for i = 1, 2, … , n
    y_i = b_i
    for j = 1, 2, … , i-1
        y_i = y_i – l_ij x y_j
    end
end
```

$$\left.\right\} \; y_i = b_i - \sum_{j=1}^{i-1} l_{ij} \, y_j$$

<u>Complexity</u>

- 1 flop = + / - / × / ÷
- Consider forward solve. For each i, the j-loop performs 2(i-1) flops.

  Total flops = sum (2i-2) = 2 sum i – sum 2
  $= 2\, n(n+1)/2 - 2\, n = n^2 - n = O(n^2)$

- flops(back solve) = n^2 (exercise)

- flops(LU) = $2/3\, n^3 + O(n^2)$

For large n, factorization is more expensive than forward or back solves.

Special Linear Systems

- Exploit special structures of linear systems
- More efficient LU factorization

Symmetric systems

- LDM^T factorization, variant of LU.

<u>Theorem</u>: If all the leading principal submatrices of A are nonsingular, then there exist unique unit lower Δ matrices L and M, and a unique diag. matrix D s.t.

$$A = L\,D\,M^T$$

Pf: Factor $A = L\,U$

Define $D = \text{diag}(d_1, \ldots, d_n)$, $d_i = u_{ii}$ $\quad i = 1, \ldots, n$

Let $M^T = D^{-1} U$ = unit upper Δ

(So M = unit lower Δ)

Thus $\quad A = L\,U = L\,D\,(D^{-1} U) = L\,D\,M^T$

Note: flops(LU) = flops(LDM^T)

<u>Theorem</u>: If A is symmetric, then $A = L\,D\,L^T$.

Pf: By previous result, $A = L\,D\,M^T$.

=> $M^{-1} A M^{-T} = M^{-1} L\,D\,M^T M^{-T} = M^{-1} L\,D$

But $M^{-1} A M^{-T}$ is symmetric, so is $M^{-1} L\,D$

Also, $M^{-1} L$ = lower Δ => $M^{-1} L\,D$ = lower Δ

A sym. lower Δ matrix => it is diag.

i.e. $M^{-1} L$ is diag.

But $M^{-1} L$ is unit lower Δ => $M^{-1} L = I$ i.e. $M = L$

Notes

(1) We can save about half the work by computing L and D only.

(2) One way is to compute the U factor only during the LU factorization

Positive definite systems

Def: A is positive definite if $x^T A x > 0$ for all $x \neq 0$.

- A is positive definite => A^{-1} exists.

Theorem: If $A = R^{n\times n}$ is PD and $X = R^{n\times k}$ has rank k, then $B = X^T A X = R^{k\times k}$ is also PD.

Pf: Let $z = R^{k\times 1}$. Then $z^T B z = z^T X^T A X z = (Xz)^T A (Xz)$

If $Xz = 0$, then X is not rank k

Hence $z^T B z > 0$.

Corollary: If A is PD, then all its principal submatrices are PD. In particular, all diag. entries are positive.

Corollary: If A is PD, then $A = L D M^T$, D has positive diag. entries.

Pf: Let $X = L^{-T}$. Then $X^T A X = L^{-1} (L D M^T) L^{-T} = D M^T L^{-T}$ is PD.

By previous corollary, $\mathrm{diag}(D M^T L^{-T})$ has positive entries.

Note that M^T and L^{-T} are unit upper Δ.

=> $M^T L^{-T}$ is also unit upper Δ.

=> $\mathrm{diag}(D M^T L^{-T}) = D$.

Symmetric positive definite systems

Theorem: If A is SPD, then there exists unique lower Δ G such that

$$A = G G^T$$

Pf: $A = L D L^T$ and $D = \mathrm{diag}(d_1, \ldots, d_n)$, $d_i > 0$.

Define $D^{1/2} = \mathrm{diag}(\mathrm{sqrt}\{d_1\}, \ldots, \mathrm{sqrt}\{d_n\})$

Let $G = L D^{1/2}$. Then G is lower Δ.

$$G G^T = L D^{1/2} (L D^{1/2})^T = L D^{1/2} D^{1/2} L^T = L D L^T = A$$

- $A = G G^T$ is called the Cholesky factorization of A and the lower Δ G is called the Cholesky factor.

Cholesky factorization

$$A = \begin{bmatrix} \alpha & v^T \\ v & B \end{bmatrix} = \begin{bmatrix} \sqrt{\alpha} & 0 \\ \dfrac{v}{\sqrt{\alpha}} & I \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & B - \dfrac{vv^T}{\alpha} \end{bmatrix} \begin{bmatrix} \sqrt{\alpha} & \dfrac{v^T}{\sqrt{\alpha}} \\ 0 & I \end{bmatrix}$$

Let $X = \begin{bmatrix} 1 & -\dfrac{v^T}{\alpha} \\ 0 & I \end{bmatrix}$. Then X has full rank.

Also $B - (vv^T)/\alpha = X^T A X$ => SPD

Hence $B - (vv^T)/\alpha = G_1 G_1^T$. Now define

$$G = \begin{bmatrix} \sqrt{\alpha} & 0 \\ \dfrac{v}{\sqrt{\alpha}} & G_1 \end{bmatrix}$$

Then $A = G G^T$.

Algorithm

```
for  k = 1, 2, … , n
    a_kk = sqrt{a_kk}
    for  j = k+1, … , n
        a_jk = a_jk / a_kk
    end
    for  j = k+1, … , n
        for  i = j, … , n
            a_ij = a_ij – a_ik a_jk
        end
    end
end
```

- flops(Cholesky) ~ $n^3/3$