Computational Methods

CMSC/AMSC/MAPL 460

Vectors, Matrices, Linear Systems, LU Decomposition,

Ramani Duraiswami,

Dept. of Computer Science

Some special matrices

Example: A **Vandermonde** matrix A is defined by a vector of elements $x_1, \ldots, x_n$. Its first column is all ones. Each later column is the preceding one times this vector.

- Matlab code
- How many operations and memory does this take?
- Vectorized operations

```
n = length(x);
V(:,1) = ones(n,1);
for j=2:n,
   V(:,j) = x.*V(:,j-1);
end
```

- Matrix may be sparse, i.e. most elements are zero.
- How many operations/memory?
- Answer still N^2 unless we avoid referring to the zero elements altogether

$$D = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 6 \end{bmatrix}$$

```
D = diag([1 2 4 6]);
```

Some special matrices

Example: A **tridiagonal** matrix

$$T = \begin{bmatrix} 1 & 3 & 0 & 0 \\ 5 & 2 & 7 & 0 \\ 0 & 9 & 4 & 8 \\ 0 & 0 & 6 & 6 \end{bmatrix}$$

can be defined by

```
T = diag([1 2 4 6]) + diag([5 9 6],-1) + diag([3 7 8], 1);
```

- Matrices may be built up from "blocks" of smaller matrices

Matrix-vector product

- Matrix-vector multiplication applies a linear transformation to a vector:

$$\mathbf{M} \bullet \mathbf{v} = \begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} & \mathbf{M}_{13} \\ \mathbf{M}_{21} & \mathbf{M}_{22} & \mathbf{M}_{23} \\ \mathbf{M}_{31} & \mathbf{M}_{32} & \mathbf{M}_{33} \end{bmatrix} \begin{bmatrix} \mathbf{v}_x \\ \mathbf{v}_y \\ \mathbf{v}_z \end{bmatrix}$$

- Recall how to do matrix multiplication
- How many operations does a matrix vector product take?

Matrix-vector product

Matrix Transformations

- A *sequence* or *composition* of linear transformations corresponds to the product of the corresponding matrices
  - Note: the matrices to the *right* affect vector first
  - Note: order of matrices matters!
- The *identity matrix* **I** has no effect in multiplication
- Some (not all) matrices have an inverse:

$$\boxed{\mathbf{M}^{-1}(\mathbf{M}(\mathbf{v})) = \mathbf{v}}$$

Ways to implement a matrix vector product

- Access matrix
  - Element-by-element along rows
  - Element-by-element along columns
  - As column vectors
  - As row vectors
- Discuss advantages

```
[m,n]=size(A);
y = zeros(m,1);
for i=1:m,
   for j=1:n,
        y(i) = y(i) + A(i,j)*x(j);
   end
end
```

```
[m,n]=size(A);
y = zeros(m,1);
for i=1:m,
     y(i) = A(i,:) * x;
end
```

```
[m,n]=size(A);
y = zeros(m,1);
for j=1:n,
    y = y + A(:,j)*x(j);
end
```

Matrix norms

- Can be defined using corresponding vector norms
  - Two norm
  - One norm
  - Infinity norm
- Two norm is hard to define … need to find maximum singular value
  - related to idea that matrix acting on unit sphere converts it in to an ellipsoid
- Frobenius norm is defined just using matrix elements

$$||A||_2 = \max_{||x||_2=1} ||Ax||_2$$

$$||A||_1 = \max_{||x||_1=1} ||Ax||_1$$

$$= \max_{j=1,\dots,n} \sum_{i=1}^{m} |a_{ij}|$$

$$||A||_\infty = \max_{||x||_1=1} ||Ax||_\infty$$

$$= \max_{i=1,\dots,n} \sum_{j=1}^{m} |a_{ij}|$$

$$||A||_F = \left(\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{i,j}|^2 \right)^{1/2}$$

Condition Number of a Matrix

A measure of how close a matrix is to singular

$$\text{cond}(A) = \kappa(A) = \|A\| \cdot \|A^{-1}\|$$

$$= \frac{\text{maximum stretch}}{\text{maximum shrink}} = \frac{\max_i |\lambda_i|}{\min_i |\lambda_i|}$$

- cond(I) = 1
- cond(singular matrix) = ∞

Solving Linear Systems

- One idea compute inverse
- Not usually a good idea
  - (unless inverse is computable easily and accurately using some matrix property)
- Leads to increased errors, and is more expensive usually

$$Ax = b$$

$$7x = 21$$

$$x = \frac{21}{7} = 3$$

$$x = 7^{-1} \times 21$$

$$= .142857 \times 21 = 2.99997$$

Representing linear systems as matrix-vector equations

$$\begin{aligned} 10x_1 - 7x_2 \quad\quad &= 7 \\ -3x_1 + 2x_2 + 6x_3 &= 4 \\ 5x_1 - \ x_2 + 5x_3 &= 6 \end{aligned}$$

- Represent it as a matrix-vector equation (linear system)
- We will apply the familiar elimination technique, and then see its matrix equivalent

$$\begin{pmatrix} 10 & -7 & 0 \\ -3 & 2 & 6 \\ 5 & -1 & 5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 7 \\ 4 \\ 6 \end{pmatrix}$$

Easy systems to solve

- Diagonal system
- Triangular system
- On board and then matlab

Solving a triangular system

```
x = zeros(n,1);
for k = n:-1:1
   x(k) = b(k)/U(k,k);
   i = (1:k-1)';
   b(i) = b(i) - x(k)*U(i,k);
end
```

```
x = zeros(n,1);
for k = n:-1:1
   j = k+1:n;
   x(k) = (b(k) - U(k,j)*x(j))/U(k,k);
end
```

Gaussian Elimination

- Zero elements of first column below 1st row
  - multiplying 1st row by 0.3 and add to 2nd row
  - multiplying 1st row by -0.5 and add to 3rd row
  - Results in
- Zero elements of first column below 2nd row
  - Swap rows
  - Multiply 2nd row by 0.04 and add to 3rd

$$\begin{pmatrix} 10 & -7 & 0 \\ -3 & 2 & 6 \\ 5 & -1 & 5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 7 \\ 4 \\ 6 \end{pmatrix}$$

$$\begin{pmatrix} 10 & -7 & 0 \\ 0 & -0.1 & 6 \\ 0 & 2.5 & 5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 7 \\ 6.1 \\ 2.5 \end{pmatrix}$$

$$\begin{pmatrix} 10 & -7 & 0 \\ 0 & 2.5 & 5 \\ 0 & -0.1 & 6 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 7 \\ 2.5 \\ 6.1 \end{pmatrix}$$

$$\begin{pmatrix} 10 & -7 & 0 \\ 0 & 2.5 & 5 \\ 0 & 0 & 6.2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 7 \\ 2.5 \\ 6.2 \end{pmatrix}$$

Solution

- Start from last equation which can be solved by division
- Next substitute in the previous line and continue
- This describes the way to do the algorithm by hand
- How to represent it using matrices?
- Also, how do we solve another system that has the same matrix?
  – Upper triangular matrix we end up with will be the same, but the sequence of operations on the r.h.s needs to be repeated

$$6.2x_3 = 6.2$$

$$2.5x_2 + (5)(1) = 2.5.$$

$$10x_1 + (-7)(-1) = 7$$

$$x = \begin{pmatrix} 0 \\ -1 \\ 1 \end{pmatrix}$$

Gaussian Elimination: LU Matrix decomposition

- It turns out that Gaussian elimination corresponds to a particular matrix decomposition …
  - Product of permutation, lower triangular and upper triangular matrices
- What is a permutation matrix?
  - It rearranges a system of equations and changes the order.
  - Multiplying by it swaps the order of rows in a matrix
  - Essentially a rearrangement of the identity
  - Nice property: transpose is its inverse: $PP^T=I$

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

$$Px = b$$

$$x = P^T b$$

LU Decomposition

- What is an upper triangular matrix?
  - Elements below diagonal are zero

$$U = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 0 & 5 & 6 & 7 \\ 0 & 0 & 8 & 9 \\ 0 & 0 & 0 & 10 \end{pmatrix}$$

- Lower triangular matrix
- Elements above diagonal are zero
- Unit lower triangular matrix
- Elements along diagonal are one

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 3 & 5 & 1 & 0 \\ 4 & 6 & 7 & 1 \end{pmatrix}$$

- Upper triangular part of Gauss Elimination is clear …
  - final matrix we end up with
- What about lower triangular and permutation?

$LU=PA$

$$L = \begin{pmatrix} 1 & 0 & 0 \\ 0.5 & 1 & 0 \\ -0.3 & -0.04 & 1 \end{pmatrix} \quad U = \begin{pmatrix} 10 & -7 & 0 \\ 0 & 2.5 & 5 \\ 0 & 0 & 6.2 \end{pmatrix} \quad P = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

- Identify the elements of L and P?
- L has the multipliers we used in the elimination steps
- P has a record of the row swaps we did to avoid dividing by small numbers
- In fact we can write each step of Gaussian elimination in matrix form

$$\begin{aligned} U &= M_{n-1}P_{n-1}\cdots M_2P_2M_1P_1A \\ L_1L_2\cdots L_{n-1}U &= P_{n-1}\cdots P_2P_1A \end{aligned}$$

$$A = \begin{pmatrix} 10 & -7 & 0 \\ -3 & 2 & 6 \\ 5 & -1 & 5 \end{pmatrix}$$

$$L = L_1 L_2 \cdots L_{n-1}$$

$$P = P_{n-1} \cdots P_2 P_1$$

the matrices defined during the elimination are

$$P_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad M_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0.3 & 1 & 0 \\ -0.5 & 0 & 1 \end{pmatrix},$$

$$P_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \quad M_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0.04 & 1 \end{pmatrix},$$

$$A = \begin{pmatrix} 10 & -7 & 0 \\ -3 & 2 & 6 \\ 5 & -1 & 5 \end{pmatrix}$$

$$P_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \ M_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0.3 & 1 & 0 \\ -0.5 & 0 & 1 \end{pmatrix},$$

$$P_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \ M_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0.04 & 1 \end{pmatrix},$$

$$L_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0.5 & 1 & 0 \\ -0.3 & 0 & 1 \end{pmatrix}, \ L_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -0.04 & 1 \end{pmatrix},$$

Solving a system with the LU decomposition

$Ax=b$

$LU=PA$

$P^T LUx=b$

$L[Ux]=Pb$

Solve $Ly=Pb$

Then $Ux=y$