IE170: Algorithms in Systems Engineering: Lecture 28

Jeff Linderoth

Department of Industrial and Systems Engineering
Lehigh University

April 11, 2007

Another Look at Matrix Multiplication

Important Notation

- If $A \in \mathbb{R}^{m \times n}$, then A_j is the j^{th} column, and a_j is the j^{th} row.

- If $A \in \mathbb{R}^{m \times k}, B \in \mathbb{R}^{k \times n}$, then $[AB]_{ij} = a_i^T B_j$.
- That is, you find the i, j element of the matrix AB, by taking the inner product of the i^{th} row of A with the j^{th} column of B.
- Naturally this is only defined if $A \in \mathbb{R}^{m \times \ell}$ and $B \in \mathbb{R}^{\ell \times n}$, wherein also

$$[AB]_{ij} = \sum_{k=1}^{\ell} a_{ik} b_{kj}.$$

Matrix Multiplication: Linear Combinations of Columns

- Looking at it another way, write B as its columns:

$$B = (B_1 \quad B_2 \cdots B_n)$$

Then the j^{th} column of AB is AB_j, or

$$AB = A\,(B_1 \quad B_2 \cdots B_n) = (AB_1 \quad AB_2 \cdots AB_n)$$

so that each column of AB is a linear combination of the columns of A, and the multipliers for the linear combination are given in the column B_j

Matrix Multiplication: Linear Combinations of Rows

- We can also express the relationship in terms of the rows of A

$$A = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{pmatrix}, AB = \begin{pmatrix} a_1 B \\ a_2 B \\ \vdots \\ a_m B \end{pmatrix}$$

So that the i^{th} row of AB is a linear combination of the rows of B, with the weights in the combination coming from the weights in a_i.

Another Nice Formula

$$(AB)^T = B^T A^T.$$

Some Definitions You Already Knew

- Vectors $\{A_1, A_2 \ldots A_n\}$ are said to be linearly dependent if the zero vector can be written as a non-trivial linear combination of the vectors, or $\exists \alpha_1, \alpha_2, \ldots, \alpha_n$ not all equal to zero such that

$$\sum_{j=1}^{n} \alpha_j A_j = 0.$$

- Alternatively, if A_j are columns of A, then the A_j are linearly dependent if $Az = 0$ for some $z \neq 0$

More Definitions You Already Knew

- Vectors $\{A_1, A_2 \ldots A_n\}$ are said to be linearly independent if the zero vector cannot be written as a non-trivial linear combination of the vectors, or

$$\sum_{j=1}^{n} \alpha_j A_j = 0 \Rightarrow \alpha_1 = \alpha_2 = \cdots = \alpha_n = 0$$

- Alternatively, if A_j are columns of A, then the A_j are linearly independent if $Az = 0 \rightarrow z = 0$. (0 is the only solution to $Az = 0$).

More Definitions

- The Range of a matrix $A \in \mathbb{R}^{m \times n}$, denoted $\mathcal{R}(A)$ is the set of all linear combinations of the columns of A. Thus, $\mathcal{R}(A) \subseteq \mathbb{R}^m$.
- $b \in \mathcal{R}(A) \Leftrightarrow \exists x \in \mathbb{R}^n$ with $Ax = b$
- The range is sometimes called the column space of A
- $\mathcal{R}(A^T) \subseteq \mathbb{R}^n$ is sometimes called the row space of A.
- Two vectors x, y are orthogonal if $x^T y = 0$
- The set of all (m-dimensional) vectors orthogonal to vectors in $\mathcal{R}(A)$ is the null space of A^T
- $z \in \mathcal{N}(A^T) \Leftrightarrow A^T z = 0$
- $z^T b = z^T (Ax) = x^T A^T z = 0$ if $b \in \mathcal{R}(A), z \in \mathcal{N}(A^T)$
- Likewise, the set of all n-dimensional vectors orthogonal to the vectors in $\mathcal{R}(A^T)$ is the null space of A

Matrix Identities

- A square matrix $A \in \mathbb{R}^{n \times n}$ whose columns are linearly independent is called nonsingular.

- A is nonsingular if $Ax = 0$ only when $x = 0$
- If A is nonsingular, then $Ax = b$ has a unique solution.
  - Proof: $Ax = b$, $Ay = b$, $A(x - y) = 0$, so $x - y = 0$ if A is nonsingular
- For every nonsingular matrix, there exists a unique matrix A^{-1} ("A inverse") such that $A^{-1}Ax = x \ \forall x \in \mathbb{R}^n$, or $AA^{-1} = I = A^{-1}A$
  - Uniqueness Proof: If B and C are both inverses: $B = BI = B(AC) = (BA)C = C$
- A square matrix whose columns all have length (norm) 1, and that are (pairwise) orthogonal is called orthogonal.
- If $Q \in \mathbb{R}^{n \times n}$ is orthogonal then (by definition) $Q^T Q = I$, so then $Q^T = Q^{-1}$.

Linear Equations

- A linear equation in n variables $x_1, \ldots, x_n$ is an equation of the form

$$a_1 x_1 + a_2 x_2 + \cdots + a_n x_n = b$$

where $a_1, a_2, \ldots, a_n$ and b are constants.
- A solution to the equation is an assignment of values to the variables such that the equation is satisfied.
- Suppose we interpret the constants $a_1, a_2, \ldots a_n$ as the entries of an n-dimensional vector a.
- Let's also make a vector x out of the variables $x_1, x_2, \ldots, x_n$.
- Then we can rewire the above equation as simply $a^T x = b$.

Systems of Linear Equations

- Suppose we are given a set of n variables whose values must satisfy more than one equation.
- In this case, we have a system of equations, such as

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \quad (1)\\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \quad (2)\\ \vdots \quad &\quad \vdots \quad (3)\\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &= b_m \quad (4) \end{aligned}$$

where a_{ij} is a constant for all $1 \le i \le m$ and $1 \le j \le n$ and $b_1, \ldots, b_m$ are constants.
- As before, a solution to this system of equations is an assignment of values to the variables such that all equations are satisfied.

Matrix Notation

- Now we can interpret the constants a_{ij} as the entries of a matrix A and the constants $b_1, \ldots, b_m$ as the entries of a vector b.
- Interpreting the variables $x_1, \ldots, x_n$ as a vector, we can again write the system of equation simply as $Ax = b$.
- From our previous discussion, we know that the system of equations $Ax = b$ has a unique solution if and only if the matrix A is square and invertible, (if the columns A_j are linearly independent).
- From now on, we will consider only such systems.
- How do we solve a systems of equations?

Special Matrices

- A square matrix D is diagonal if $d_{ij} = 0$ whenever $i \neq j$.
- A square matrix L is lower triangular if $l_{ij} = 0$ whenever $j > i$.
- A square matrix U is upper triangular if $u_{ij} = 0$ whenever $j < i$.
- A square matrix P is a permutation matrix if there is a single 1 in each row and column.
- The identity matrix, usually denoted I is a diagonal matrix that is also a permutation matrix.
- What effect does (right)-multiplying by a permutation matrix have?
- What effect does (left)-multiplying by a permutation matrix have?

The LUP Decomposition

- Let's suppose that we are able to find three $n \times n$ matrices L, U, and P such that

$$PA = LU$$

where
  - L is upper triangular.
  - U is lower triangular with 1's on the diagonal.
  - P is a permutation matrix.
- This is called an LUP decomposition of A.
- How could use such a decomposition to solve the system $Ax = b$?

Using the LUP Decomposition

- Once we have an LUP decomposition, we can use it to easily solve the system $Ax = b$.
- Note that the system $PAx = Pb$ is equivalent to the original system, which is then equivalent to $LUx = Pb$.
- We can solve the system in two steps:
  - First solve the system $Ly = Pb$ (forward substitution).
  - Then solve the system $Ux = y$ (backward substitution).

How Does This Help Us

- First, let's convince ourselves that Pb is really nothing more than a "permuted" version of b.
- Typically permutation matrices P are (compactly) represented by an array $\pi[1, \ldots, n]$.
- $\pi[i] = 1 \Rightarrow P_{i,\pi[i]} = 1, P_{ij} = 0 \forall j \neq \pi[i]$
- Recall: left multiply just takes linear combinations of the rows.
  - PA has (i, j) entry of $a_{\pi[i],j}$ and Pb has $b_{\pi[i]}$ in the i^{th} position.
- You will see (28.1-5) that PA is A with rows permuted, and AP is A with columns permuted

Forward Substitution

$$\begin{array}{ccccccccc}
y_1 & & & & & & & = & b_{\pi[1]} \\
l_{21}y_1 & + & y_2 & + & \cdots & & = & b_{\pi[2]} & \\
l_{n1}y_1 & + & l_{n2}y_2 & + & l_{n3}y_3 + & + & \cdots & = & b_{\pi[n]}
\end{array}$$

- Just substitute forward into:

$$y_i = b_{\pi[i]} - \sum_{j=1}^{i-1} l_{ij} y_j$$

Finding the LU Decomposition

- Let's assume for now that $P = I$ and concentrate on finding L and U.
- We can find the these two matrices using a procedure similar to Gaussian elimination.
- In fact, we will implement the algorithm recursively.
- First we'll divide the matrix A into four pieces:

Breaking up A

$$A = \left[\begin{array}{c|ccc} a_{11} & a_{12} & \cdots & a_{1n} \\ \hline a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{array}\right] \quad (5)$$

$$= \begin{bmatrix} a_{11} & w^T \\ v & A' \end{bmatrix} \quad (6)$$

- Next, we'll use use row operations to change v into the zero vector and record the operations in another matrix.

Finding the LU Decomposition (cont.)

- By simple multiplication, you can verify the following factorization of A:

$$A = \begin{bmatrix} a_{11} & w^T \\ v & A' \end{bmatrix} \quad (7)$$

$$= \begin{bmatrix} 1 & 0 \\ v/a_{11} & I \end{bmatrix} \begin{bmatrix} a_{11} & w^T \\ 0 & A' - vw^T/a_{11} \end{bmatrix} \quad (8)$$

- We can show that if A is nonsingular, then so is $A' - vw^T/a_{11}$.
- So we can recursively call the method to factor the $(n-1) \times (n-1)$ matrix $A' - vw^T/a_{11}$.
- Applying this recursion n times yields the desired factorization

Finding the LU Decomposition (cont.)

- To see how to get the factorization from the recursive application of the algorithm, we have the following.

$$A = \begin{bmatrix} 1 & 0 \\ v/a_{11} & I \end{bmatrix} \begin{bmatrix} a_{11} & w^T \\ 0 & A' - vw^T/a_{11} \end{bmatrix} \quad (9)$$

$$= \begin{bmatrix} 1 & 0 \\ v/a_{11} & I \end{bmatrix} \begin{bmatrix} a_{11} & w^T \\ 0 & L'U' \end{bmatrix} \quad (10)$$

$$= \begin{bmatrix} 1 & 0 \\ v/a_{11} & L' \end{bmatrix} \begin{bmatrix} a_{11} & w^T \\ 0 & U' \end{bmatrix} \quad (11)$$

- This shows how to obtain the factorization recursively.
- This can also be done iteratively and "in place."

Finding the LUP Decomposition

- The element a_{11} is called the pivot element.
- Note that the above decomposition method fails whenever the pivot element is zero.
- In this case, we can permute the rows of A to obtain a new pivot element.
- In fact, for numerical stability, it is desirable to have the pivot element be as large as possible in absolute value.
- If no nonzero pivot is available, A is singular.
- This leads to the following modified factorization.

$$QA = \begin{bmatrix} a_{k1} & w^T \\ v & A' \end{bmatrix} \tag{12}$$

$$= \begin{bmatrix} 1 & 0 \\ v/a_{k1} & I \end{bmatrix} \begin{bmatrix} a_{k1} & w^T \\ 0 & A' - vw^T/a_{k1} \end{bmatrix} \tag{13}$$

Finding the LUP Decomposition (cont.)

- As before, we obtain L', U', and P' and we get

$$PA = \begin{bmatrix} 1 & 0 \\ 0 & P' \end{bmatrix} QA \tag{14}$$

$$= \begin{bmatrix} 1 & 0 \\ 0 & P' \end{bmatrix} \begin{bmatrix} 1 & 0 \\ v/a_{k1} & I \end{bmatrix} \begin{bmatrix} a_{k1} & w^T \\ 0 & A' - vw^T/a_{k1} \end{bmatrix} \tag{15}$$

$$= \begin{bmatrix} 1 & 0 \\ P'v/a_{k1} & I \end{bmatrix} \begin{bmatrix} a_{k1} & w^T \\ 0 & P'(A' - vw^T/a_{k1}) \end{bmatrix} \tag{16}$$

$$= \begin{bmatrix} 1 & 0 \\ P'v/a_{k1} & I \end{bmatrix} \begin{bmatrix} a_{k1} & w^T \\ 0 & L'U' \end{bmatrix} \tag{17}$$

$$= \begin{bmatrix} 1 & 0 \\ P'v/a_{k1} & L' \end{bmatrix} \begin{bmatrix} a_{k1} & w^T \\ 0 & U' \end{bmatrix} \tag{18}$$

- What is the running time of finding the LUP decomposition?

Using the LUP Decomposition

- Note that finding the decomposition has the same running time as Gaussian elimination.
- The decomposition can be stored in almost the same space as the original matrix.
- Once we have an LUP decomposition, we can solve $Ax = b$ with various right hand sides in time $\Theta(n^2)$.