AMS526: Numerical Analysis I (Numerical Linear Algebra for Computational and Data Sciences)

Lecture 2: Algorithmic Consideration; Orthogonality

Xiangmin Jiao

Stony Brook University

Outline

Algorithms for Matrix-Vector Multiplication

- Suppose $A \in \mathbb{R}^{m \times n}$ and $x \in \mathbb{R}^n$
- MATLAB-style code for $b = Ax$:

Row oriented	Column oriented

for $i = 1 : m$
  $b(i) = 0$
  for $j = 1 : n$
    $b(i) = b(i) + A(i,j) * x(j)$
  end
end
	$b(:) = 0$
	for $j = 1 : n$
	  for $i = 1 : m$
	    $b(i) = b(i) + A(i,j) * x(j)$
	  end
	end

- Number of operations is $O(mn)$, but big-Oh notation is insufficient
- Number of operations is $2mn$: coefficient of leading-order term is important for comparison

Flop Count

- It is important to assess *efficiency* of algorithms. But how?
  - We could implement different algorithms and do direct comparison, but implementation details can affect true performance
  - We could estimate cost of all operations, but it is very tedious
  - Relatively simple and effective approach is to estimate amount of floating-point operations, or "flops", and focus on asymptotic analysis as sizes of matrices approach infinity
- Idealization
  - Count each operation $+$,$-$, $*$, $/$, and $\sqrt{\ }$ as one flop
  - This estimation is crude, as it omits data movement in memory, which is non-negligible on modern computer architectures (e.g., different loop orders can affect cache performance)
- Matrix-vector product requires about $2mn$ flops
- Suppose $m = n$, it takes quadratic time in n, or $O(n^2)$

Algorithms for Saxpy: Scalar a x plus y

- Saxpy computes $ax + y$ and updates y

$$y = ax + y \Rightarrow y_i = ax_i + y_i$$

- Suppose $x, y \in \mathbb{R}^n$ and $a \in \mathbb{R}$

MATLAB-style code

for $i = 1 : n$
$\quad y(i) = y(i) + a * x(i)$
end

Pseudo-code

for $i = 1 : n$
$\quad y_i \leftarrow y_i + a\,x_i$

- Number of flops is $2n$
- Pseudo-code cannot run on any computer, but are human readable and straightforward to convert into real codes in any programming language (e.g., C, FORTRAN, MATLAB, etc.)
- We use pseudo-code on slides for conciseness

Gaxpy: Generalized saxpy

- Computes $y = y + Ax$, where $A \in \mathbb{R}^{m \times n}$ and $x \in \mathbb{R}^n$

Row oriented	Column oriented

for $i = 1 : m$
    for $j = 1 : n$
        $y_i = y_i + a_{ij} x_j$
	for $j = 1 : n$
	    for $i = 1 : m$
	        $y_i = y_i + a_{ij} x_j$

- Inner loop of column-oriented algorithm can be converted to $y = y + x_j a_{:,j}$
- Number of flops is $2mn$

Matrix Multiplication Update

- Computes $C = C + AB$, where $A \in \mathbb{R}^{m\times r}$, $B \in \mathbb{R}^{r\times n}$, and $C \in \mathbb{R}^{m\times n}$

$$c_{ij} = c_{ij} + \sum_{k=1}^{r} a_{ik} b_{kj}$$

$$\boxed{\begin{array}{l} \textbf{for } i = 1:m \\ \quad \textbf{for } j = 1:n \\ \quad\quad \textbf{for } k = 1:r \\ \quad\quad\quad c_{ij} = c_{ij} + a_{ik} b_{kj} \end{array}}$$

- The operation further generalizes Gaxpy
- Number of flops is $2mnr$
- In BLAS (Basic Linear Algebra Subroutines), functions are grouped into level-1, 2, and 3, depending on whether complexity is linear, quadratic, or cubic

Six Variants of Algorithms

- There are six variants depending on permutation of i, j, and k:

$$ijk, \quad jik, \quad ikj, \quad jki, \quad kij, \quad kji$$

- Inner product version: $c_{ij} = c_{ij} + a_{i,:} b_j$

$$\begin{array}{l} \textbf{for } i = 1 : m \\ \quad \textbf{for } j = 1 : n \\ \quad \quad c_{ij} = c_{ij} + a_{i,:} b_j \end{array}$$

- Saxpy version: computes as $c_j = c_j + Ab_j$

$$\begin{array}{l} \textbf{for } j = 1 : n \\ \quad c_j = c_j + Ab_j \end{array}$$

- Outer product version: computes as $C = C + \sum_{k=1}^{r} a_k b_{:,k}$

$$\begin{array}{l} \textbf{for } k = 1 : r \\ \quad C = C + a_k b_{:,k} \end{array}$$

Outline

Inner Product

- *Euclidean length* of u is square root of inner product of u with itself, i.e., $\sqrt{u^T u}$
- Inner product of two **unit** vectors u and v is cosine of angle α between u and v, i.e., $\cos\alpha = u^T v$
- Inner product is *bilinear*, in the sense that it is linear in each vertex separately:

$$
\begin{aligned}
(u_1 + u_2)^T v &= u_1^T v + u_2^T v \\
u^T (v_1 + v_2) &= u^T v_1 + u^T v_2 \\
(\alpha u)^T (\beta v) &= \alpha\beta u^T v
\end{aligned}
$$

Orthogonal Vectors

Definition

A pair of vectors are *orthogonal* if $x^T y = 0$.

In other words, angle between them is 90 degrees

Definition

Two sets of vectors X and Y are orthogonal if every $x \in X$ is orthogonal to every $y \in Y$.

- Subspace $S^\perp$ is *orthogonal complement* of S if they are orthogonal and complementary subspaces
- For $A \in \mathbb{R}^{m\times n}$, null$(A)$ is *orthogonal complement* of range(A^T)

Definition

A set of nonzero vectors S is *orthogonal* if they are pairwise orthogonal. They are *orthonormal* if it is orthogonal and in addition each vector has unit Euclidean length.

Orthogonal Vectors

Theorem

The vectors in an orthogonal set S are linearly independent.

Proof.

Prove by contradiction. If a vector can be expressed as linear combination of the other vectors in the set, then it is orthogonal to itself. □

Question: If the column vectors of an $m \times n$ matrix A are orthogonal, what is the rank of A?

Orthogonal Vectors

Theorem

The vectors in an orthogonal set S are linearly independent.

Proof.

Prove by contradiction. If a vector can be expressed as linear combination of the other vectors in the set, then it is orthogonal to itself. □

Question: If the column vectors of an $m \times n$ matrix A are orthogonal, what is the rank of A?

Answer: $n = \min\{m, n\}$. In other words, A has full rank.

Components of Vector

- Given an orthonormal set $\{q_1, q_2, \ldots, q_m\}$ forming a basis of $\mathbb{R}^m$, vector v can be decomposed into orthogonal components as $v = \sum_{i=1}^{m}(q_i^T v)q_i$
- Another way to express the decomposition is $v = \sum_{i=1}^{m}(q_i q_i^T)v$
- $q_i q_i^T$ is an *orthogonal projection matrix*
- More generally, given an orthonormal set $\{q_1, q_2, \ldots, q_n\}$ with $n \leq m$, we have

$$v = r + \sum_{i=1}^{n}(q_i^T v)q_i = r + \sum_{i=1}^{n}(q_i q_i^T)v \text{ and } r^T q_i = 0, 1 \leq i \leq n$$

- Let Q be composed of column vectors $\{q_1, q_2, \ldots, q_n\}$. $QQ^T = \sum_{i=1}^{n}(q_i q_i^T)$ is an orthogonal projection matrix (more in Lecture 4)
- Question: What is $\boldsymbol{Q}^T\boldsymbol{Q}$ equal to?

Components of Vector

- Given an orthonormal set $\{q_1, q_2, \ldots, q_m\}$ forming a basis of $\mathbb{R}^m$, vector v can be decomposed into orthogonal components as $v = \sum_{i=1}^{m} (q_i^T v) q_i$
- Another way to express the decomposition is $v = \sum_{i=1}^{m} (q_i q_i^T) v$
- $q_i q_i^T$ is an *orthogonal projection matrix*
- More generally, given an orthonormal set $\{q_1, q_2, \ldots, q_n\}$ with $n \leq m$, we have

$$v = r + \sum_{i=1}^{n} (q_i^T v) q_i = r + \sum_{i=1}^{n} (q_i q_i^T) v \text{ and } r^T q_i = 0, 1 \leq i \leq n$$

- Let Q be composed of column vectors $\{q_1, q_2, \ldots, q_n\}$. $QQ^T = \sum_{i=1}^{n} (q_i q_i^T)$ is an orthogonal projection matrix (more in Lecture 4)
- Question: What is $\boldsymbol{Q}^T \boldsymbol{Q}$ equal to?
- Answer: $\boldsymbol{Q}^T \boldsymbol{Q} = \boldsymbol{I}$.

Orthogonal Matrices

Definition

A matrix is *orthogonal* if $Q^T = Q^{-1}$, i.e., if $Q^T Q = QQ^T = I$.

- Its column vectors are *orthonormal*. In other words, $q_i^T q_j = \delta_{ij}$, the *Kronecker delta*.
- For complex matrices, we say the matrix is *unitary* if $Q^H = Q^{-1}$.

Question: What is the geometric meaning of multiplication by an orthogonal matrix?

Orthogonal Matrices

Definition

A matrix is *orthogonal* if $Q^T = Q^{-1}$, i.e., if $Q^T Q = QQ^T = I$.

- Its column vectors are *orthonormal*. In other words, $q_i^T q_j = \delta_{ij}$, the *Kronecker delta*.
- For complex matrices, we say the matrix is *unitary* if $Q^H = Q^{-1}$.

Question: What is the geometric meaning of multiplication by an orthogonal matrix?

Answer: It preserves angles and Euclidean length. In the real case, multiplication by an orthogonal matrix Q is a rotation (if $\det(Q) = 1$) or reflection (if $\det(Q) = -1$).