UBC MATH 307

LU Decompositions

Construction

Use Gaussian elimination to produce an upper triangular matrix, U. The lower triangular matrix L records the inverse of the row reduction operations. No pivoting of rows allowed. If A can be reduced by Gaussian elimination to row echelon form only with operations without scaling rows and without interchanging rows, then A has an LU decomposition of the form:

$$A = LU$$

Where L records the coefficients $c_{i,j}$ for each row reduction operation– add $c_{i,j}$ times row j to row i:

$$L = \begin{bmatrix} 1 & & & & \\ -c_{2,1} & 1 & & & \\ -c_{3,1} & -c_{3,2} & 1 & & \\ \vdots & \vdots & \ddots & \ddots & \\ -c_{m,1} & -c_{m,2} & \cdots & -c_{m,m-1} & 1 \end{bmatrix}$$

Back Substitution

To solve $A\boldsymbol{x} = \boldsymbol{b}$, use the decomposition $LU\boldsymbol{x} = \boldsymbol{b}$

$$\boxed{\text{Solve:} L\boldsymbol{y} = \boldsymbol{b}, \text{ then } U\boldsymbol{x} = \boldsymbol{y}}$$

Useful Properties of LU

$\text{rank}(A) = \text{rank}(U)$

$\det(A) = \det(U) = \Pi(\text{Diagonal entries } U)$

$N(A) = N(U)$

$R(A) = \text{span}\{\boldsymbol{\ell}_1, \ldots, \boldsymbol{\ell}_r\}$for $r = \text{rank}(A)$,

Compiled by Simon Ghyselincks 2022
based off of Math 307 content generously provided by UBC Math at
`https://ubcmath.github.io/MATH307/`

Matrix Norms and Error Analysis

Definitions

$$\|A\| = \max_{\|\boldsymbol{x}\|=1} \|A\boldsymbol{x}\| \quad \text{and} \quad \|A^{-1}\| = \frac{1}{\min\limits_{\|\boldsymbol{x}\|=1} \|A\boldsymbol{x}\|}$$

Matrix norm generally has properties of vector norm.

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

Finding Matrix Norm

Use the SVD decomposition of A, the singular values of the Σ matrix determine the norm:

$$\sigma_{max} = ||A|| \quad \text{and} \quad \frac{1}{\sigma_{min}} = ||A^{-1}||$$

Relative Error Formula

$$A\boldsymbol{x} = \boldsymbol{b} \implies \frac{\|\Delta \boldsymbol{x}\|}{\|\boldsymbol{x}\|} \leq \text{cond}(A)\frac{\|\Delta \boldsymbol{b}\|}{\|\boldsymbol{b}\|}$$

Subspaces

Definitions

$U \subseteq \mathbb{R}^n$ is a **subspace** if:
1. U contains the zero vector $\mathbf{0}$
2. $\boldsymbol{u}_1 + \boldsymbol{u}_2 \in U$ for all $\boldsymbol{u}_1, \boldsymbol{u}_2 \in U$
3. $c\boldsymbol{u} \in U$ for all $c \in \mathbb{R}, \boldsymbol{u} \in U$

span$\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m\}$ = all possible linear combinations of vectors included in the brackets

Basis $\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m\}$ forms basis of U if:
1. $\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m\}$ is a linearly independent set
2. $\text{span}\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m\} = U$

Range of A is $R(A) = \text{span}\{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n\}$ column vecs

Rank-Nullity Theorem:
$A(m \times n \text{ matrix}) \implies \text{rank}(A) + \dim(N(A)) = n$

Orthogonality of Fundamental Subspaces	
$R(A) = N(A^T)^\perp$	$N(A) = R(A^T)^\perp$
$R(A^T) = N(A)^\perp$	$N(A^T) = R(A)^\perp$

Interpolation

Polynomial Interpolation

$$y = p(t) = c_0 + c_1 t + \cdots + c_d t^d$$

$$\underbrace{\begin{bmatrix} 1 & t_0 & \cdots & t_0^d \\ 1 & t_1 & \cdots & t_1^d \\ \vdots & \vdots & \ddots & \vdots \\ 1 & t_d & \cdots & t_d^d \end{bmatrix}}_{\text{Vandermonde Matrix}} \underbrace{\begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_d \end{bmatrix}}_{\text{Coefficients}} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_d \end{bmatrix}$$

$$\det(\text{Vandermonde}) = \prod_{0 \leq i < j \leq d} (t_j - t_i)$$

Determinant is the product of the difference of each unique pair of t

Cubic Spline Interpolation

For $N + 1$ points $p(t)$ is a piecewise cubic function:

$$p_k(t) = a_k(t - t_{k-1})^3 + b_k(t - t_{k-1})^2 + c_k(t - t_{k-1}) + d_k$$

1. **Interpolation at left endpoints**:
$p_k(t_{k-1}) = y_{k-1}$ for $k = 1, \ldots, N$ yields N equations.
2. **Interpolation at right endpoints**:
$p_k(t_k) = y_k$ for $k = 1, \ldots, N$ yields N equations.
3. **Continuity of** $p'(t)$:
$p'_k(t_k) = p'_{k+1}(t_k)$ for $k = 1, \ldots, N-1$ yields $N-1$ eqns.
4. **Continuity of** $p''(t)$:
$p''_k(t_k) = p''_{k+1}(t_k)$ for $k = 1, \ldots, N-1$ yields $N-1$ eqns.

Fundamental Subspaces

Space	Method of Solving
$R(A)$	LU decomp to find $r = \text{rank}(A)$, then pick the first r columns of L
$N(A)$	LU decomp then solve $U\boldsymbol{x} = 0$ using parameters $c_1, c_2, \ldots, c_n$ for each free variable to get $\boldsymbol{x} = c_1\boldsymbol{v_1} + c_2\boldsymbol{v_2} + \ldots + c_n\boldsymbol{v_n}$ and the vectors $\boldsymbol{v}$ are the basis of $N(A)$
$R(A^T)$	transpose then LU, or take the orthogonal complement of $N(A)$
$N(A^T)$	transpose then LU, or take the orthogonal complement of $R(A)$

Orthogonality and Projections

Orthogonal Complement

Vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m \in \mathbb{R}^n$ are orthogonal if $\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle = 0$ for all $i \neq j$, **orthonormal** if they are orthogonal and each is a unit vector, $\|\boldsymbol{x}_k\| = 1, k = 1, \ldots, m$.

$$U^\perp = \{\boldsymbol{x} \in \mathbb{R}^n : \langle \boldsymbol{x}, \boldsymbol{y} \rangle = 0 \text{ for all } \boldsymbol{y} \in U\}$$

To find $U^\perp$ place the basis vectors of U as the columns of matrix A, then:

$$U^\perp = \left\{\boldsymbol{x} \in \mathbb{R}^n : A^T \boldsymbol{x} = 0\right\} = N(A^T)$$

Projections

$$\text{proj}_{\boldsymbol{u}}(\boldsymbol{x}) = \frac{\langle \boldsymbol{x}, \boldsymbol{u} \rangle}{\langle \boldsymbol{u}, \boldsymbol{u} \rangle} \boldsymbol{u} \quad \text{proj}_{\boldsymbol{u}}(\boldsymbol{x}) = P\boldsymbol{x} \quad P = \frac{1}{\|\boldsymbol{u}\|^2} \boldsymbol{u}\boldsymbol{u}^T$$

Where P is a projection matrix with properties:

$$P^2 = P \quad P^T = P \quad \text{rank}(P) = 1$$

Any matrix with the first two properties is an **orthogonal projector**

Gram-Schmidt Orthogonalization Algorithm

Take a basis of vectors for U: $\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m\}$, then:

$$\boldsymbol{v_k} = \boldsymbol{u_k} - \sum_{n=1}^{k-1} \text{proj}_{\boldsymbol{v_n}}(\boldsymbol{u_k})$$

Applied sequentially from $k = 1$ to $k = m$ yields m orthogonal vectors forming a basis of U. Normalize all of the $\boldsymbol{v}$ vectors to form orthonormal basis.

Subspace Projections

$\text{proj}_U(\boldsymbol{x}) = \frac{\langle \boldsymbol{x}, \boldsymbol{u}_1 \rangle}{\langle \boldsymbol{u}_1, \boldsymbol{u}_1 \rangle} \boldsymbol{u}_1 + \cdots + \frac{\langle \boldsymbol{x}, \boldsymbol{u}_m \rangle}{\langle \boldsymbol{u}_m, \boldsymbol{u}_m \rangle} \boldsymbol{u}_m$
For an orthonormal basis, $\text{proj}_U(\boldsymbol{x}) = P_U = AA^T$, where A has columns of $\boldsymbol{u}$

$$P_U + P_{U^\perp} = I$$

The projection of $\boldsymbol{x}$ onto a subspace U gives the closest vector to $\boldsymbol{x}$ in U

QR and Orthogonal Matrices

Orthogonal Matrices

A matrix A is **orthogonal** if $A^T A = AA^T = I$, then:

1. $\|A\boldsymbol{x}\| = \|\boldsymbol{x}\|$ for all $\boldsymbol{x} \in \mathbb{R}^n$
2. the columns and rows of are orthonormal
3. $A^T = A^{-1}$

QR Definition and Construction

$A = QR$ where Q is an orthogonal matrix and R is an upper triangular matrix. Q by definition must be a square matrix, its columns include the orthonormal basis of $R(A)$ via GS Algorithm for the first column entries, Q_1. The orthonormal basis of $R(A)^\perp$ fills the remaining columns Q_2.

$$\underbrace{A}_{n \times m} = QR = \underbrace{\begin{bmatrix} Q_1 & Q_2 \end{bmatrix}}_{n \times n} \underbrace{\begin{bmatrix} R_1 \\ 0 \end{bmatrix}}_{n \times m}$$

$$R_1 = \begin{bmatrix} \langle \boldsymbol{w}_1, \boldsymbol{a}_1 \rangle & \langle \boldsymbol{w}_1, \boldsymbol{a}_2 \rangle & \cdots & \langle \boldsymbol{w}_1, \boldsymbol{a}_m \rangle \\ & \langle \boldsymbol{w}_2, \boldsymbol{a}_2 \rangle & \cdots & \langle \boldsymbol{w}_2, \boldsymbol{a}_m \rangle \\ & & \ddots & \vdots \\ & & & \langle \boldsymbol{w}_m, \boldsymbol{a}_m \rangle \end{bmatrix}$$

The entire structure is simply a projection of the columns of A onto the orthonormal basis of A. The R vector contains the information to reverse the GS algorithm process used for Q:

$$\boldsymbol{a}_1 = \langle \boldsymbol{w}_1, \boldsymbol{a}_1 \rangle \boldsymbol{w}_1$$

$$\boldsymbol{a}_2 = \langle \boldsymbol{w}_1, \boldsymbol{a}_2 \rangle \boldsymbol{w}_1 + \langle \boldsymbol{w}_2, \boldsymbol{a}_2 \rangle \boldsymbol{w}_2$$

$$\vdots$$

$$\boldsymbol{a}_m = \langle \boldsymbol{w}_1, \boldsymbol{a}_m \rangle \boldsymbol{w}_1 + \langle \boldsymbol{w}_2, \boldsymbol{a}_m \rangle \boldsymbol{w}_2 + \cdots + \langle \boldsymbol{w}_m, \boldsymbol{a}_m \rangle \boldsymbol{w}_m$$

As with any orthonormal basis:

$$\text{proj}_{R(A)}(\boldsymbol{x}) = Q_1 Q_1^T \boldsymbol{x}$$

Least Squares Methods

Definition

The least squares method approximates a linear system with no solution by finding the closest vector to $\boldsymbol{b}$ in $R(A)$: $A\boldsymbol{x} = \text{proj}_{R(A)}\boldsymbol{b}$

$$\boxed{\text{A } \boldsymbol{x} \cong \boldsymbol{b}}$$

We assume that A is a $m \times n$ matrix where $m > n$ and also $\text{rank}(A) = n$

Methods of Solving

$$A^T A\boldsymbol{x} = A^T \boldsymbol{b} \qquad R_1 \boldsymbol{x} = Q_1^T \boldsymbol{b} \qquad \boldsymbol{x} = A^+ \boldsymbol{b}$$

Note that the QR method is generally preferred as having a lower condition number involved and not being overly computationally expensive.

Curve Fitting

For m data points (t_i, y_i) fitted to the linear combination of n functions f_k, the coefficients c_k can be found by applying least squares to $A\boldsymbol{c} \cong \boldsymbol{y}$

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \; A = \begin{bmatrix} f_1(t_1) & f_2(t_1) & \cdots & f_n(t_1) \\ f_1(t_2) & f_2(t_2) & \cdots & f_n(t_2) \\ \vdots & & & \vdots \\ f_1(t_m) & f_2(t_m) & \cdots & f_n(t_m) \end{bmatrix} \; \boldsymbol{c} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}$$

PDP Decomposition and Power Method

Definitions

A $(n \times n)$ is diagonalizable if and only if A has n linearly independent eigenvectors, $A = PDP^{-1}$:

$$P = \underbrace{\begin{bmatrix} \boldsymbol{v}_1 & \cdots & \boldsymbol{v}_n \end{bmatrix}}_{\text{Eigenvectors}} \quad D = \underbrace{\begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix}}_{\text{Eigenvalues}}$$

Spectral Theorem

If A is a real symmetric matrix then the eigenvalues of A are real, the eigenvectors for distinct eigenvalues are orthogonal, and A has a

$$PDP^{-1}$$

where P is orthogonal: $A = PDP^T$. To make P orthogonal, make sure to normalize the eigenvectors to length 1.

Power Method

A dominant eigenvalue is a unique eigenvalue that is the largest for a given matrix A.

$$\boldsymbol{x}_{k+1} = \frac{A\boldsymbol{x}_k}{\|A\boldsymbol{x}_k\|_\infty} \qquad \|\boldsymbol{x}_k\|_\infty = \max\{\boldsymbol{x}_k\}$$

Check for convergence to $A\boldsymbol{x_k} = \lambda\boldsymbol{x_k}$ with the Rayleigh quotient:

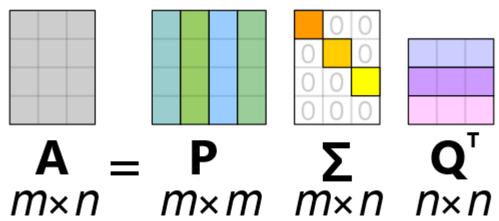
$$\lambda_k = \frac{\boldsymbol{x}_k^T A \boldsymbol{x}_k}{\boldsymbol{x}_k^T \boldsymbol{x}_k}$$

The smallest eigenvalue can be found by iterating the inverse matrix (use LU with back substitution for repeated inverse operations).

SVD Decomposition

Definitions

Any $m \times n$ matrix A has a singular value decomposition $A = P\Sigma Q^T$ where P and Q are orthogonal matrices and Σ is a diagonal matrix ordered high to low $(\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_r)$.



$$A = P\Sigma Q^T \quad \text{where} \quad \Sigma = \left[\begin{array}{ccc|c} \sigma_1 & & & \\ & \ddots & & \mathbf{0} \\ & & \sigma_r & \\ \hline & \mathbf{0} & & \mathbf{0} \end{array}\right]_{m \times n}$$

Construction

σ are known as the singular values of A. We find them as the $\sqrt{\lambda}$ of either AA^T or A^TA. The columns and rows of P and Q are corresponding *normalized* eigenvectors:

$$A^TA\boldsymbol{q}_i = \sigma_i^2\boldsymbol{q}_i \qquad AA^T\boldsymbol{p}_i = \sigma_i^2\boldsymbol{p}_i$$

$$\boldsymbol{p}_i = \frac{1}{\sigma_i}A\boldsymbol{q}_i \qquad \boldsymbol{q}_i = \frac{1}{\sigma_i}A^T\boldsymbol{p}_i$$

Either P or Q may not have a full orthonormal basis defined based on the dimension and rank of A. In these cases simply extend the basis to a full one by finding the remaining orthogonal subspace and applying GS algorithm.

Important Properties of SVD

$$\text{rank}(A) = r$$
$$\|A\| = \sigma_1$$
$$\|A^{-1}\| = 1/\sigma_r$$
$$\text{cond}(A) = \sigma_1/\sigma_r$$
$$\text{null}(A) = \text{span}\{\boldsymbol{q}_{r+1}, \ldots, \boldsymbol{q}_n\}$$
$$\text{range}(A) = \text{span}\{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_r\}$$

PCA and Applications of SVD

Principle Component Analysis

Create a data matrix X composed of row vectors, the same shape as used for least squares fitting. Normalize data so the columns have a mean value of 0. Apply SVD decomposition to the data matrix, the principle weight vectors are the q vectors. We can project X onto its principle components to reduce the dimension of the data while keeping the most significant weight vectors of dimensioning.

Pseudo Inverse

$$A^+ = Q\Sigma^+P^T$$

$$\Sigma^+ = \left[\begin{array}{ccc|c} \sigma_1^{-1} & & & \\ & \ddots & & \mathbf{0} \\ & & \sigma_r^{-1} & \\ \hline & \mathbf{0} & & \mathbf{0} \end{array}\right]_{n \times m}$$

SVD Expansion

$$A = \sum_{i=1}^{r} \sigma_i \boldsymbol{p}_i \boldsymbol{q}_i^T = \sigma_1 \boldsymbol{p}_1 \boldsymbol{q}_1^T + \cdots + \sigma_r \boldsymbol{p}_r \boldsymbol{q}_r^T$$

Complex Linear Algebra

Inner Product

$$< \boldsymbol{v}, \boldsymbol{w} >= v_0\overline{w_0} + v_1\overline{w_1} + \cdots + v_n\overline{w_n}$$

$$< c\boldsymbol{v}, \boldsymbol{w} >= c < \boldsymbol{v}, \boldsymbol{w} > \qquad < \boldsymbol{v}, \boldsymbol{w} >= \overline{< \boldsymbol{w}, \boldsymbol{v} >}$$

$$< \boldsymbol{v}, c\boldsymbol{w} >= \bar{c} < \boldsymbol{v}, \boldsymbol{w} > \qquad < \boldsymbol{v}, \boldsymbol{v} >= ||\boldsymbol{v}||$$

Complex Analogs

Hermitian: $A = \overline{A}^T$

Unitary: $A^{-1} = \overline{A}^T$

If A is hermitian then diagonal entries of A are real

Roots of Unity

The Nth complex roots of 1:

$\omega = e^{\frac{2\pi ik}{N}}$ for $0 < k < N, k \in \mathbb{C}^N$

Discrete Fourier Transform

Fourier Basis

The fourier basis of $\mathbb{C}^N$ is an orthogonal basis defined as $\boldsymbol{f}_0, \ldots, \boldsymbol{f}_{N-1}$:

$$\boldsymbol{f}_k = \begin{bmatrix} 1 \\ \omega_N^k \\ \omega_N^{2k} \\ \vdots \\ \omega_N^{(N-1)k} \end{bmatrix}$$

$$\overline{\boldsymbol{f}}_k = \boldsymbol{f}_{N-k}$$

Fourier Transform

$$\mathrm{DFT}(\boldsymbol{x}) = F_N \boldsymbol{x} \text{ and } F_N =$$

$$\begin{bmatrix} \overline{\boldsymbol{f}}_0^T \\ \overline{\boldsymbol{f}}_1^T \\ \vdots \\ \overline{\boldsymbol{f}}_{N-1}^T \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \overline{\omega}_N & \overline{\omega}_N^2 & \cdots & \overline{\omega}_N^{N-1} \\ 1 & \overline{\omega}_N^2 & \overline{\omega}_N^4 & \cdots & \overline{\omega}_N^{2(N-1)} \\ 1 & \vdots & \vdots & \ddots & \vdots \\ 1 & \overline{\omega}_N^{N-1} & \overline{\omega}_N^{2(N-1)} & \cdots & \overline{\omega}_N^{(N-1)^2} \end{bmatrix}$$

Note that $\frac{1}{\sqrt{N}} F_N$ is unitary

$$\boldsymbol{x} = \frac{1}{N} \begin{bmatrix} \boldsymbol{f}_0 & \cdots & \boldsymbol{f}_{N-1} \end{bmatrix} \begin{bmatrix} \overline{\boldsymbol{f}}_0^T \\ \overline{\boldsymbol{f}}_1^T \\ \vdots \\ \overline{\boldsymbol{f}}_{N-1}^T \end{bmatrix} \boldsymbol{x}$$

Let $\boldsymbol{y} = \mathrm{DFT}(\boldsymbol{x})$ then:

$$\overline{\boldsymbol{y}[k]} = \boldsymbol{y}[N-k]$$

Inverse Transform

$$\mathrm{IDFT}(\boldsymbol{y}) = \frac{1}{N} \overline{F}_N^T \boldsymbol{y}$$

DFT Analysis

Sinusoids

For N number of samples over a time period of 1, we have:

$$\boldsymbol{n} = \begin{bmatrix} 0 \\ 1 \\ 2 \\ \vdots \\ N-1 \end{bmatrix} \qquad \boldsymbol{t} = (1/N)\boldsymbol{n} = \begin{bmatrix} 0 \\ 1/N \\ 2/N \\ \vdots \\ (N-1)/N \end{bmatrix}$$

A discrete sinusoid signal is $\boldsymbol{x} = A\cos(2\pi k \boldsymbol{t} + \phi)$

The kth Fourier vector has the following sinusoidal properties:

$$\boldsymbol{f}_k = \cos(2\pi k \boldsymbol{t}) + i \sin(2\pi k \boldsymbol{t})$$

$$\frac{1}{2}\left(\boldsymbol{f}_k + \overline{\boldsymbol{f}}_k\right) = \cos(2\pi k \boldsymbol{t})$$

$$\frac{1}{2i}\left(\boldsymbol{f}_k - \overline{\boldsymbol{f}}_k\right) = \sin(2\pi k \boldsymbol{t})$$

For $\boldsymbol{x} = A\cos(2\pi k \boldsymbol{t} + \phi)$:

$$\mathrm{DFT}(\boldsymbol{x}) = \frac{AN}{2} e^{i\phi}\, \boldsymbol{e}_k + \frac{AN}{2} e^{-i\phi}\, \boldsymbol{e}_{N-k}$$

Stemplots