Cholesky decomposition

Let a real matrix $\mathbf{A}$ is

symmetric: $\mathbf{A}^T = \mathbf{A}$

positive definite: $\mathbf{x}^T \mathbf{A} \mathbf{x} > 0 \quad \forall \mathbf{x} \in \mathbb{R}^m$

Then

$$\mathbf{A} = \mathbf{L}\mathbf{L}^T$$

where $\mathbf{L}$ is a lower triangular matrix.

Cholesky decomposition

$$\mathbf{A} = \mathbf{L}\mathbf{L}^T$$

$$= \begin{pmatrix} l_{11} & 0 & 0 & \cdots & 0 \\ l_{21} & l_{22} & 0 & \cdots & 0 \\ l_{31} & l_{32} & l_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \\ l_{m1} & l_{m2} & l_{m3} & \cdots & l_{mm} \end{pmatrix} \begin{pmatrix} l_{11} & l_{21} & l_{31} & \cdots & l_{m1} \\ 0 & l_{22} & l_{32} & \cdots & l_{m2} \\ 0 & 0 & l_{33} & \cdots & l_{m3} \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & l_{mm} \end{pmatrix}$$

1st row of $\mathbf{A}$

$$a_{11} = l_{11}^2$$

$$a_{12} = l_{11} l_{21}, \quad \cdots, \quad a_{1k} = l_{11} l_{k1}, \quad k = 2, \ldots, m$$

NB: The square root is OK because $\mathbf{A}$ is positive definite.

Cholesky decomposition

$$\mathbf{A} = \mathbf{L}\mathbf{L}^T$$

$$= \begin{pmatrix} l_{11} & 0 & 0 & \cdots & 0 \\ l_{21} & l_{22} & 0 & \cdots & 0 \\ l_{31} & l_{32} & l_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \\ l_{m1} & l_{m2} & l_{m3} & \cdots & l_{mm} \end{pmatrix} \begin{pmatrix} l_{11} & l_{21} & l_{31} & \cdots & l_{m1} \\ 0 & l_{22} & l_{32} & \cdots & l_{m2} \\ 0 & 0 & l_{33} & \cdots & l_{m3} \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & l_{mm} \end{pmatrix}$$

2nd row of $\mathbf{A}$

$$a_{21} = l_{21} l_{11}$$

$$a_{22} = l_{21}^2 + l_{22}^2, \quad \cdots, \quad a_{2k} = l_{21} l_{k1} + l_{22} l_{k2}, \quad k = 2, \ldots, m$$

NB: The square root is OK because $\mathbf{A}$ is positive definite.

Cholesky decomposition

Compared to a general $\mathbf{LU}$ factorization, Cholesky decomposition:

- requires $1/2$ memory
- requires $\sim 1/2$ less operations
- has better stability, and does *not* require pivoting
- fails if $\mathbf{A}$ is not PD.

Systems of linear equations

If $\mathbf{A}$ is symmetric and positive definite, then

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

is solved via $\mathbf{A} = \mathbf{L}\mathbf{L}^T$, and

$$\mathbf{L}\mathbf{y} = \mathbf{b} \qquad \text{(forward substitution)}$$
$$\mathbf{L}^T\mathbf{x} = \mathbf{y} \qquad \text{(back substitution)}$$

Applications of Cholesky decomposition

Quantum mechanics

Observables are represented by Hermitian operators. ($\left(\mathbf{A}^T\right)^* = \mathbf{A}$)

Applications of Cholesky decomposition

Numerical optimization

The Hessian matrix of a multivariate function $F(\mathbf{x})$

$$H_{jk} = \frac{\partial^2 F}{\partial x_j \partial x_k}$$

is symmetric and is in some cases positive (semi-)definite.

Applications of Cholesky decomposition

Monte Carlo simulations

Generation of correlated Gaussian random variables: decompose the correlation matrix $\mathbf{C} = \mathbf{L}\mathbf{L}^T$, generate a vector of uncorrelated values $\mathbf{x}$, then

$$\mathbf{z} = \mathbf{L}\mathbf{x}$$

has the correlation matrix $\mathbf{C}$.

QR decomposition

Any real $m \times n$ matrix $\mathbf{A}$ can be decomposed into

$$\mathbf{A} = \mathbf{QR}$$

where $\mathbf{Q}$ is an $m \times m$ orthogonal matrix ($\mathbf{Q}^T\mathbf{Q} = 1$) and $\mathbf{R}$ is an $m \times n$ upper triangular matrix.

IOW, a rectangular matrix can be reduced to an upper triangular form by an orthogonal transformation $\mathbf{Q}^T$

$$\mathbf{Q}^T\mathbf{A} = \mathbf{R}$$

Thin $\mathbf{QR}$ factorization

If $m > n$

$$\begin{aligned}\mathbf{A} &= \mathbf{Q}\begin{pmatrix}\mathbf{R}_1\\ 0\end{pmatrix}\\ &= \left(\mathbf{Q}_1\mathbf{Q}_2\right)\begin{pmatrix}\mathbf{R}_1\\ 0\end{pmatrix}\\ &= \mathbf{Q}_1\mathbf{R}_1\end{aligned}$$

where $\mathbf{R}_1$ is $m \times m$ uppper triangular, $\mathbf{Q}_1$ is $m \times n$ and has orthonormal columns.

This is a *thin* $\mathbf{QR}$ factorization (or *economic*, or *reduced* factorization).

Thin QR factorization

If $\mathbf{A}$ has full column rank (i.e., columns of $\mathbf{A}$ are all linearly independent), then

- The thin factorization $\mathbf{A} = \mathbf{Q}_1\mathbf{R}_1$ is unique
- Diagonal elements of $\mathbf{R}_1$ are positive
- $\mathbf{R}_1^T$ is a lower triangular Cholesky factor of $\mathbf{A}^T\mathbf{A}$

Constructing the QR factorization

- Householder reflections
- Givens rotations

Both reduce $\mathbf{A}$ to $\mathbf{R}$ column by column, and construct $\mathbf{Q}$ as a product of orthogonal matrices.

Householder reflections

Given a vector $\mathbf{x} \in \mathbb{R}^m$, reflect it across a hyperplane with the normal vector $\mathbf{u}$ ($\|\mathbf{u}\|_2 = 1$)

$$\mathbf{x} = \mathbf{x}_{\|} + \mathbf{x}_{\perp}, \qquad \begin{array}{l} \mathbf{x}_{\perp} \perp \mathbf{x}_{\|} \\ \mathbf{x}_{\perp} \parallel \mathbf{u} \end{array}$$

The perp component is given by $\mathbf{x}_{\perp} = \mathbf{u}\langle\mathbf{u} \cdot \mathbf{x}\rangle$

$$\mathbf{y} = \mathbf{x}_{\|} - \mathbf{x}_{\perp} = (\mathbf{x}_{\|} + \mathbf{x}_{\perp}) - 2\mathbf{x}_{\perp} = \mathbf{x} - 2\mathbf{u}\langle\mathbf{u} \cdot \mathbf{x}\rangle$$

Householder reflections

In the matrix form, $\langle \mathbf{u} \cdot \mathbf{x} \rangle \equiv \mathbf{u}^T \mathbf{x}$, and the Householder transformation is

$$\mathbf{y} = \mathbf{H}\mathbf{x} = \left(\mathbf{1} - 2\mathbf{u}\mathbf{u}^T\right)\mathbf{x}$$

The Householder matrices are

- Symmetric, $\mathbf{H}^T = \mathbf{H}$
- Orthogonal, $\mathbf{H}^T\mathbf{H} = \mathbf{1}$

Householder reflections

Given two vectors $\mathbf{x}$ and $\mathbf{y}$ with $\|\mathbf{x}\|_2 = \|\mathbf{y}\|_2$, construct a $\mathbf{H}$ which converts $\mathbf{x}$ to $\mathbf{y}$.

Reflect $\mathbf{x}$ across the hyperplane which bisects the angle between $\mathbf{x}$ and $\mathbf{y}$.
The Householder transformation with

$$\mathbf{u} = (\mathbf{x} - \mathbf{y})/\|\mathbf{x} - \mathbf{y}\|_2$$

Householder reflections

Given two vectors $\mathbf{x}$ and $\mathbf{y}$ with $\|\mathbf{x}\|_2 = \|\mathbf{y}\|_2$, construct a $\mathbf{H}$ which converts $\mathbf{x}$ to $\mathbf{y}$.

Reflect $\mathbf{x}$ across the hyperplane which bisects the angle between $\mathbf{x}$ and $\mathbf{y}$.
The Householder transformation with

$$\mathbf{u} = (\mathbf{x} - \mathbf{y}) / \|\mathbf{x} - \mathbf{y}\|_2$$

$$\mathbf{x} = \begin{pmatrix} \times \\ \times \\ \vdots \\ \times \end{pmatrix} \qquad \mathbf{y} = \begin{pmatrix} \times \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Householder reflections $\mathbf{QR}$

$$\mathbf{H}_1\mathbf{A} = \begin{pmatrix} \times & \times & \times & \cdots & \times \\ 0 & \times & \times & \cdots & \times \\ 0 & \times & \times & \cdots & \times \\ & & \cdots & & \\ 0 & \times & \times & \cdots & \times \end{pmatrix}$$

Householder reflections QR

$$\mathbf{H}_1\mathbf{A} = \begin{pmatrix} \times & \times & \times & \cdots & \times \\ 0 & \times & \times & \cdots & \times \\ 0 & \times & \times & \cdots & \times \\ & & \cdots & & \\ 0 & \times & \times & \cdots & \times \end{pmatrix}$$

$$\mathbf{H}_2\mathbf{H}_1\mathbf{A} = \begin{pmatrix} \times & \times & \times & \cdots & \times \\ 0 & \times & \times & \cdots & \times \\ 0 & 0 & \times & \cdots & \times \\ & & \cdots & & \\ 0 & 0 & \times & \cdots & \times \end{pmatrix}$$

Householder reflections $\mathbf{QR}$

After n steps: $$\mathbf{H}_n \cdots \mathbf{H}_2\mathbf{H}_1\mathbf{A} = \mathbf{R}$$
so that
$$\mathbf{A} = \mathbf{QR}$$
with
$$\mathbf{Q} = \mathbf{H}_1\mathbf{H}_2 \cdots \mathbf{H}_n$$

Householder reflections $\mathbf{QR}$

After n steps: $$\mathbf{H}_n \cdots \mathbf{H}_2 \mathbf{H}_1 \mathbf{A} = \mathbf{R}$$
so that
$$\mathbf{A} = \mathbf{QR}$$

with
$$\mathbf{Q} = \mathbf{H}_1 \mathbf{H}_2 \cdots \mathbf{H}_n$$

Computational complexity strongly depends on the order of calculations.

$$\begin{array}{lll} \left(\mathbf{u}\mathbf{u}^T\right)\mathbf{x} & \text{is} & O(m^2) \\ \mathbf{u}\left(\mathbf{u}^T\mathbf{x}\right) & \text{is} & O(m) \end{array}$$

In practice, *never* form the $\mathbf{H}$ matrices explicitly.

Householder reflections: avoiding the roundoff

$$\mathbf{H}\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix} = \begin{pmatrix} \|\mathbf{x}\|_2 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

involves computing $x_1 - \|\mathbf{x}\|_2$ which is prone to a catastrophic cancellation.

For $x_1 > 0$, write

$$\begin{aligned} x_1 - \|\mathbf{x}\|_2 &= \frac{x_1^2 - \|\mathbf{x}\|_2^2}{x_1 + \|\mathbf{x}\|_2} \\ &= \frac{-x_2^2 - \cdots - x_m^2}{x_1 + \|\mathbf{x}\|_2} \end{aligned}$$

Householder reflections $\mathbf{QR}$

The Householder $\mathbf{QR}$ algorithm:

- has excellent stability
- for square matrices involves only several times more work than $\mathbf{LU}$
- is not easy to parallelize

Givens rotations

$$\begin{pmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1m} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2m} \\ & & \cdots & & \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mm} \end{pmatrix}$$

Find ϕ such that

$$\begin{pmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{pmatrix} \begin{pmatrix} a_{11} \\ a_{21} \end{pmatrix} = \begin{pmatrix} \times \\ 0 \end{pmatrix}$$

Givens rotations

$$\begin{pmatrix} \cos\phi & 0 & -\sin\phi \\ 0 & 1 & 0 \\ \sin\phi & 0 & \cos\phi \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1m} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2m} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3m} \\ & & \cdots & & \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mm} \end{pmatrix}$$

Find ϕ such that

$$\begin{pmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{pmatrix} \begin{pmatrix} a_{11} \\ a_{31} \end{pmatrix} = \begin{pmatrix} \times \\ 0 \end{pmatrix}$$

$\mathbf{QR}$ decomposition via Givens rotations

- Complexity is $3/2$ of the Householder $\mathbf{QR}$ algorithm
- There is flexibility in selecting the order of introducing zeros.

Sherman-Morrison formula

Suppose we solved

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

Now, we “slighly” change $\mathbf{A}$. Can we update $\mathbf{x}$?

Sherman-Morrison formula

Suppose we solved

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

Now, we "slighly" change $\mathbf{A}$. Can we update $\mathbf{x}$?

Yes, for an outer product update:

$$\left(\mathbf{A} + \mathbf{u}\mathbf{v}^T\right)\mathbf{x} = \mathbf{b}$$

where $\mathbf{u}$ and $\mathbf{v}$ are known column vectors.

If $u_j = \delta_{jp}$, this updates the p-th row of $\mathbf{A}$.
If $v_j = \delta_{jp}$, this updates the p-th column of $\mathbf{A}$

Sherman-Morrison formula

$$\left(\mathbf{A} + \mathbf{u}\,\mathbf{v}^T\right)^{-1} = \left(\mathbf{1} + \mathbf{A}^{-1}\mathbf{u}\,\mathbf{v}^T\right)^{-1}\mathbf{A}^{-1}$$

Sherman-Morrison formula

$$\left(\mathbf{A} + \mathbf{u}\,\mathbf{v}^T\right)^{-1} = \left(\mathbf{1} + \mathbf{A}^{-1}\mathbf{u}\,\mathbf{v}^T\right)^{-1}\mathbf{A}^{-1}$$

(expand into a formal power series)

$$= \left(\mathbf{1} - \mathbf{A}^{-1}\mathbf{u}\,\mathbf{v}^T + \mathbf{A}^{-1}\mathbf{u}\mathbf{v}^T\mathbf{A}^{-1}\mathbf{u}\mathbf{v}^T + \cdots\right)\mathbf{A}^{-1}$$

Sherman-Morrison formula

$$\left(\mathbf{A}+\mathbf{u}\,\mathbf{v}^T\right)^{-1}=\left(\mathbf{1}+\mathbf{A}^{-1}\mathbf{u}\,\mathbf{v}^T\right)^{-1}\mathbf{A}^{-1}$$

(expand into a formal power series)

$$=\left(\mathbf{1}-\mathbf{A}^{-1}\mathbf{u}\,\mathbf{v}^T+\mathbf{A}^{-1}\mathbf{u}\mathbf{v}^T\mathbf{A}^{-1}\mathbf{u}\mathbf{v}^T+\cdots\right)\mathbf{A}^{-1}$$

(define $\lambda=\mathbf{v}^T\mathbf{A}\mathbf{u}$)

$$=\mathbf{A}^{-1}-\left(\mathbf{A}^{-1}\mathbf{u}\right)\left(\mathbf{v}^T\mathbf{A}^{-1}\right)\left(1-\lambda+\lambda^2+\cdots\right)$$

Sherman-Morrison formula

$$\left(\mathbf{A} + \mathbf{u}\,\mathbf{v}^T\right)^{-1} = \left(\mathbf{1} + \mathbf{A}^{-1}\mathbf{u}\,\mathbf{v}^T\right)^{-1}\mathbf{A}^{-1}$$

(expand into a formal power series)

$$= \left(\mathbf{1} - \mathbf{A}^{-1}\mathbf{u}\,\mathbf{v}^T + \mathbf{A}^{-1}\mathbf{u}\mathbf{v}^T\mathbf{A}^{-1}\mathbf{u}\mathbf{v}^T + \cdots\right)\mathbf{A}^{-1}$$

(define $\lambda = \mathbf{v}^T\mathbf{A}\mathbf{u}$)

$$= \mathbf{A}^{-1} - \left(\mathbf{A}^{-1}\mathbf{u}\right)\left(\mathbf{v}^T\mathbf{A}^{-1}\right)\left(1 - \lambda + \lambda^2 + \cdots\right)$$

(sum the geometric series)

$$= \mathbf{A}^{-1} - \frac{\left(\mathbf{A}^{-1}\mathbf{u}\right)\left(\mathbf{v}^T\mathbf{A}^{-1}\right)}{1 + \lambda}$$

Sherman-Morrison update

Let $\mathbf{y}$ is the (known) solution of $\mathbf{A}\mathbf{y} = \mathbf{b}$.
We want to find the solution of $\left(\mathbf{A} + \mathbf{u}\,\mathbf{v}^T\right)\mathbf{x} = \mathbf{b}$.

Define z as the solution of $\mathbf{A}\mathbf{z} = \mathbf{u}$.
Note that $\lambda = \mathbf{v}^T\mathbf{z}$.

Then the solution $\mathbf{x}$ of $\left(\mathbf{A} + \mathbf{u}\,\mathbf{v}^T\right)\mathbf{x} = \mathbf{b}$ is

$$\begin{aligned} \mathbf{x} &= \mathbf{A}^{-1}\mathbf{b} - \frac{\left(\mathbf{A}^{-1}\mathbf{u}\right)\left(\mathbf{v}^T\mathbf{A}^{-1}\right)}{1+\lambda}\mathbf{b} \\ &= \mathbf{y} - \mathbf{z}\frac{\left(\mathbf{v}^T\mathbf{y}\right)}{1+\left(\mathbf{v}^T\mathbf{z}\right)} \end{aligned}$$

The complexity of the Sherman-Morrison update is $O(m^2)$.

Linear systems with banded left-hand sides

A matrix $\mathbf{A}$ is *banded* if $a_{ij} = 0$ for $|i - j| > p$. Then p is the *bandwidth*.

For banded matrices, constructing the LU factorization is $O(m)$.

Memory requirements for storing banded matrices is also $O(m)$.

Tridiagonal systems: Thomas algorithm

$$\begin{pmatrix} b_1 & c_1 & & & \\ a_2 & b_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & a_{m-1} & b_{m-1} & c_{m-1} \\ & & & a_m & b_m \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \cdots \\ x_{m-1} \\ x_m \end{pmatrix} = \begin{pmatrix} d_1 \\ d_2 \\ \cdots \\ d_{m-1} \\ d_m \end{pmatrix}$$

Forward sweep: bidiagonalization

row 1: $b_1 x_1 + c_1 x_2 = d_1 \qquad \Rightarrow x_1 = \alpha_1 x_2 + \beta_1$

row2: $a_2 x_1 + b_2 x_2 + c_2 x_3 = d_2 \qquad \Rightarrow x_2 = \alpha_2 x_3 + \beta_2$

$\ldots$

Tridiagonal systems: Thomas algorithm

Forward sweep: bidiagonalization

$$\begin{pmatrix} \times & \times & & & \\ & \times & \times & & \\ & & \ddots & \ddots & \\ & & & \times & \times \\ & & & & \times \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{m-1} \\ x_m \end{pmatrix} = \begin{pmatrix} \times \\ \times \\ \vdots \\ \times \\ \times \end{pmatrix}$$

Then do backsubstitution starting from x_m.

The total computational complexity is $O(m)$.

Tridiagonal systems: Thomas algorithm

The corresponding LU factorization is

$$\mathbf{A} = \begin{pmatrix} \times & \times & & & \\ & \times & \times & & \\ & & \ddots & \ddots & \\ & & & \times & \times \\ & & & & \times \end{pmatrix} \begin{pmatrix} \times & & & & \\ \times & \times & & & \\ & \ddots & \ddots & & \\ & & & \times & \\ & & & \times & \times \end{pmatrix}$$

Tridiagonal systems: Thomas algorithm

The corresponding LU factorization is

$$\mathbf{A} = \begin{pmatrix} \times & \times & & & \\ & \times & \times & & \\ & & \ddots & \ddots & \\ & & & \times & \times \\ & & & & \times \end{pmatrix} \begin{pmatrix} \times & & & & \\ \times & \times & & & \\ & \ddots & \ddots & & \\ & & & \times & \\ & & & \times & \times \end{pmatrix}$$

Stability

The algorithm is stable if $\mathbf{A}$ is *diagonally dominant*:

$$|b_k| \geq |a_k| + |c_k|, \qquad k = 1, \ldots, m$$