Cholesky Factorization

- Cholesky factorization
- existence and uniqueness
- algorithm
- solving positive definite linear systems

Cholesky factorization

- every symmetric positive definite matrix $A \in \mathbb{R}^{n \times n}$ can be factored as

$$A = LL^{\mathsf{T}}$$

  where L is lower triangular with positive diagonal elements.

- L is called the *Cholesky factor* of A
- also common to write $A = R^{\mathsf{T}}R$ where R is upper triangular
- can be interpreted as a matrix *square root*
- we write $L = \mathbf{chol}(A)$

Reverse Cholesky factorization

- Can also factor A into UU^{T} where U is upper triangular with positive diagonal entries.
- This is achieved by reversing rows/columns, applying standard Cholesky, then reversing again.

Let Z be the *reverser matrix* so that $Zx = (x_n, \ldots, x_1)$

- applying Cholesky we have $ZAZ = \tilde{L}\tilde{L}^{\mathsf{T}}$, and so

$$A = Z\tilde{L}Z Z\tilde{L}^{\mathsf{T}}Z = UU^{\mathsf{T}}$$

Uniqueness

- *uniqueness*: for any matrix A, its Cholesky factorization $A = LL^{\mathsf{T}}$ with positive diagonal entries is **unique**.

- first we show this when $A = I$. Then $LL^{\mathsf{T}} = I$ means $L^{-1} = L^{\mathsf{T}}$. Since L is lower triangular, so is L^{-1}, and so L must be diagonal. Since $L_{ii}^2 = 1$ we must have $L = I$.

- for the general case, suppose $LL^T = \tilde{L}\tilde{L}^T$, then

$$\tilde{L}^{-1} L L^T \tilde{L}^{-T} = I.$$

which implies

$$\left(\tilde{L}^{-1} L\right)\left(\tilde{L}^{-1} L\right)^T = I.$$

The matrix $\bar{L} = \tilde{L}^{-1} L$ is lower triangular with positive diagonal entries. Since it satisfies $\bar{L}\bar{L}^T = I$, we have $\bar{L} = I$, which in turn means $L = \tilde{L}$.

Existence

- *existence*: the Cholesky factorization **exists if and only if** A is PD.
- (only if): if $A = LL^T$, then A is symmetric and positive definite, since L is invertible
- (if): we will give an algorithm that constructs it

Cholesky of Gram matrix and inverse

- *Gram matrix*: If $B \in \mathbb{R}^{m \times n}$ has independent columns, $A = B^{\mathsf{T}} B$ is positive definite
  - Using QR factorization $B = QR$, we get $A = R^{\mathsf{T}} R$.
  - R has positive diagonals, so let $L = R^{\mathsf{T}}$
  - using QR is good numerically, since it avoids 'squaring' B

- *inverse*: first, we Cholesky factorize the reversed A, so that $ZAZ = \tilde{L}\tilde{L}^{\mathsf{T}}$, then

$$\begin{aligned} A^{-1} &= Z\tilde{L}^{-T}\tilde{L}^{-1}Z \\ &= (Z\tilde{L}^{-T}Z)(Z\tilde{L}^{-1}Z) \end{aligned}$$

  and $Z\tilde{L}^{-T}Z$ is lower triangular with positive diagonal entries

Cholesky factorization algorithm (block recursive)

- recall completion of squares

$$A = \begin{bmatrix} a & b^T \\ b & D \end{bmatrix} = \begin{bmatrix} I & 0 \\ b^T/a & I \end{bmatrix} \begin{bmatrix} a & 0 \\ 0 & D - bb^T/a \end{bmatrix} \begin{bmatrix} I & b^T/a \\ 0 & I \end{bmatrix}$$

$$= \begin{bmatrix} \sqrt{a} & 0 \\ b^T/\sqrt{a} & I \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & D - bb^T/a \end{bmatrix} \begin{bmatrix} \sqrt{a} & b^T/\sqrt{a} \\ 0 & I \end{bmatrix}$$

- let $D - bb^T/a = \begin{bmatrix} \alpha & \beta^T \\ \beta & \delta \end{bmatrix}$, and repeat, which gives

$$A = \begin{bmatrix} \sqrt{a} & 0 & 0 \\ b_1/\sqrt{a} & \sqrt{\alpha} & 0 \\ b_{2:n-1}/\sqrt{a} & \beta/\sqrt{\alpha} & I \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \delta - \beta\beta^T/\alpha \end{bmatrix} \begin{bmatrix} \sqrt{a} & b_1/\sqrt{a} & b_{2:n-1}^T/\sqrt{a} \\ 0 & \sqrt{\alpha} & \beta^T/\sqrt{\alpha} \\ 0 & 0 & I \end{bmatrix}$$

- repeat until completion

Cholesky factorization algorithm (block recursive)

- after step 1, we have $S = D - bb^{\mathsf{T}}/a = \begin{bmatrix} \alpha & \beta^{\mathsf{T}} \\ \beta & \delta \end{bmatrix}$
- S is positive definite, since it is a Schur complement of A
- therefore
  - $\alpha > 0$, so in step 2, we have that $\sqrt{\alpha}$ is real
  - at the next step we can apply the same process to S
- if during the algorithm we ever attempt to square-root a negative number, that means that the original matrix A was not positive definite
- a practical test for positive definiteness

Cholesky factorization algorithm

- same algorithm as above, written differently
- given $n \times n$ symmetric positive definite matrix A
- For $j = 1, \ldots, n$

  1. Determine diagonal entry:

$$L_{jj} = \left(A_{jj} - \sum_{k=1}^{j-1} L_{jk}^2 \right)^{1/2}$$

  2. For $i = j+1, \ldots, n$ Determine entries below diagonal:

$$L_{ij} = \frac{1}{L_{jj}} \left(A_{ij} - \sum_{k=1}^{j-1} L_{ik} L_{jk} \right)$$

Application: solving symmetric positive definite linear systems

- the equation $Ax = b$ when A is symmetric and positive definite occurs very frequently, *e.g.*, in Newton's method for minimizing a function

- algorithm:

  1. Factor A as $A = LL^{\mathsf{T}}$.
  2. Solve $LL^{\mathsf{T}}x = b$
     - Solve $Ly = b$ for y by forward substitution.
     - Solve $L^{\mathsf{T}}x = y$ for x by back substitution.

- Total: $\frac{1}{3}n^3 + 2n^2 \sim \frac{1}{3}n^3$ flops.
  - Factorization: $\frac{1}{3}n^3$ flops (roughly)
  - Forward and backward substitution: $2n^2$ flops.

Sparse positive definite matrices

- Cholesky factorization of dense matrices
  - $\frac{1}{3}n^3$ flops.
  - practical for n up to several 1000.

- Cholesky factorization of sparse matrices
  - If A is very sparse, R is often (but not always) sparse.
  - If R is sparse, the cost of the factorization is much less than $\frac{1}{3}n^3$.
  - Exact cost depends on n, number of nonzeros, and sparsity pattern (fill-in).
  - Very large sets of equations can be solved by exploiting sparsity.

Sparse Cholesky factorization with permutation

- if A is sparse and positive definite, it is usually factored as

$$A = PLL^{\mathsf{T}}P^{\mathsf{T}}$$

  where P is a permutation matrix
- we permute the rows and columns of A and factor $P^{\mathsf{T}}AP = LL^{\mathsf{T}}$.
- choice of permutation P greatly affects the sparsity of R (minimizing "fill-in").
- heuristic methods exist for choosing good permutations