Cholesky and QR Decomposition

Chris Tinkler

Cholesky Decomposition

Special factorization of symmetric, positive definite matrices

- Symmetric: $A_{ij} = A_{ji}$
- Positive Definite: v * A * v > 0 for all v
  - Equivalent to A having all positive eigen vectors

Cholesky decomposition is roughly 2x faster than LU decomposition when solving linear equations.

Properties

$L * L^T = A$

- L is a lower triangular matrix

Numerically stable without decomposition

- If decomp fails A is not positive definite

Decomposing (Finding L)

- Cholesky algorithm (modified Gaussian elimination)

$$L_{ji} = \frac{1}{L_{ii}}\left(a_{ij} - \sum_{k=0}^{i-1} L_{ik}L_{jk}\right) \qquad j = i+1, i+2, \ldots, N-1$$

- $L*y = b$
- $L^T*x = y$

Finding A^{-1} with Cholesky

$A = L * L^{-1}$

$A^{-1} = (L^{-1})^T(L^{-1})$

Solving Systems of Linear Eqn with Cholesky

Solve $L * y = b$ with forward substitution

Solve $L^T * x = y$ with forward substitution

QR Decomposition

Exists for all rectangular matrices

- A = Q * R
  - R: Upper Triangular
  - Q: Orthogonal
    - $Q^T * Q = 1$

Roughly twice as many operations as LU Decomp, so only used to solve special cases of linear equations (least squares problem).

Methods of Decomposition

Finding Q:

- Gram-Schmidt
- Householder Reflections

$R = A * Q^T$

Gram-Schmidt

Q is found using the Gram–Schmidt process

- Generates orthogonal set of vectors from a finite, linearly independent set of vectors

Inherently unstable method of orthogonalization

$$
\begin{aligned}
\mathbf{u}_1 &= \mathbf{v}_1, & \mathbf{e}_1 &= \frac{\mathbf{u}_1}{\|\mathbf{u}_1\|} \\
\mathbf{u}_2 &= \mathbf{v}_2 - \mathrm{proj}_{\mathbf{u}_1}(\mathbf{v}_2), & \mathbf{e}_2 &= \frac{\mathbf{u}_2}{\|\mathbf{u}_2\|} \\
\mathbf{u}_3 &= \mathbf{v}_3 - \mathrm{proj}_{\mathbf{u}_1}(\mathbf{v}_3) - \mathrm{proj}_{\mathbf{u}_2}(\mathbf{v}_3), & \mathbf{e}_3 &= \frac{\mathbf{u}_3}{\|\mathbf{u}_3\|} \\
\mathbf{u}_4 &= \mathbf{v}_4 - \mathrm{proj}_{\mathbf{u}_1}(\mathbf{v}_4) - \mathrm{proj}_{\mathbf{u}_2}(\mathbf{v}_4) - \mathrm{proj}_{\mathbf{u}_3}(\mathbf{v}_4), & \mathbf{e}_4 &= \frac{\mathbf{u}_4}{\|\mathbf{u}_4\|} \\
&\;\;\vdots & &\;\;\vdots \\
\mathbf{u}_k &= \mathbf{v}_k - \sum_{j=1}^{k-1} \mathrm{proj}_{\mathbf{u}_j}(\mathbf{v}_k), & \mathbf{e}_k &= \frac{\mathbf{u}_k}{\|\mathbf{u}_k\|}.
\end{aligned}
$$

https://en.wikipedia.org/wiki/Gram%E2%80%93Schmidt_process#Example

Householder Reflections

Def: a transformation that reflects a vector about a plane

Householder Matrix: N x N orthogonal matrix of the form

$$P = I - \frac{2}{v^T v} v v^T, \qquad 0 \neq v \in \mathbb{R}^n.$$

https://towardsdatascience.com/qr-matrix-factorization-15bae43a6b2

Householder matrices can zero the elements of a column under a certain element.

We apply Householder matrices in succession to create a final matrix Q.

- $Q = Q_0 Q_1 \ldots Q_{n-2}$
- $R = Q_{n-2} Q_{n-1} \ldots Q_0 * A$

More stable than Gram-Schmidt, but more costly and non parallelizable

Finding A^{-1} with QR Decomp

$A = Q * R$

$A^{-1} = R^{=1} * Q^{-1} = R^{-1} * Q^T$

Solving Sys of Lin Eqn with QR Decomp

$A * x = b$

Form $Q^T * b$

Solve $R * x = Q^T * b$ by back substitution

Least Squares Problem

Sets of equations in which there are more equations than unknowns

Cannot take inverse

Use another set of parameters

Minimize sum of squares of residuals to find best set

$$X\beta = QR\beta = y$$

$$\hat{\beta} = (QR)^{-1}y = R^{-1}Q^T y$$

Examples:

Cholesky:

https://atozmath.com/example/MatrixEv.aspx?he=e&q=choleskydecomp&ex=1

QR (Householder):

https://atozmath.com/example/MatrixEv.aspx?he=e&q=qrdecomphh