7 Lecture 6, Jan 21

Announcements

- Quiz 1 returned (???).
- HW2 due next Monday.

Last time

- Memory hierarchy, high-level BLAS, effect of data layout.
- Triangular systems.

Today

- Gaussian elimination and LU decomposition.
- Cholesky decomposition.

Gaussian elimination and LU decomposition

Given a system of linear algebraic equations

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

Step 1: Each row times a_{11}/a_{k1},
then use row one to subtract other rows.

$$\Rightarrow \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & \tilde{a}_{22} & \cdots & \tilde{a}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \tilde{a}_{n2} & \cdots & \tilde{a}_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \tilde{b}_2 \\ \vdots \\ \tilde{b}_n \end{bmatrix}$$

Step 2: The second row and down multiply by $\tilde{a}_{22}/\tilde{a}_{k2}$,
then use row two to subtract every row below.

$$\Rightarrow \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & \tilde{a}_{22} & \tilde{a}_{23} & \cdots & \tilde{a}_{2n} \\ 0 & 0 & \tilde{a}_{33} & \cdots & \tilde{a}_{3n} \\ \vdots & \vdots & \cdots & \ddots & \vdots \\ 0 & 0 & \tilde{a}_{n3} & \cdots & \tilde{a}_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \tilde{b}_2 \\ \tilde{b}_3 \\ \vdots \\ \tilde{b}_n \end{bmatrix}$$

Step 3: Similar to the previous two steps, repeat until all
elements in the lower triangle of the matrix A
become zeros.

$$\Rightarrow \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & \tilde{a}_{22} & \cdots & \tilde{a}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \tilde{a}_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \tilde{b}_2 \\ \vdots \\ \tilde{b}_n \end{bmatrix}$$

- History: It is one by-product of Gauss's efforts to re-discover the dwarf planet Ceres using method of least squares. No linear algebra in 1801!
- Solve $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ for a general matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$.
- A toy example: `https://en.wikipedia.org/wiki/Gaussian_elimination#Example_of_the_algorithm`.

- Idea: a series of *elementary operations* that turn $\boldsymbol{A}$ into a triangular system.
- We consider the square $\boldsymbol{A}$ case first.
- *Elementary operator matrix* $\boldsymbol{E}_{jk}(c)$ is the identity with the 0 in position (j,k) replaced by c. For any vector $\boldsymbol{x}$,

$$\boldsymbol{E}_{jk}(c)\boldsymbol{x} = (x_1, \ldots, x_{j-1}, x_j + cx_k, x_{j+1}, \ldots, x_n)^\mathsf{T}.$$

  Applying $\boldsymbol{E}_{jk}(c)$ to both sides of the system $\boldsymbol{Ax} = \boldsymbol{b}$ replaces the j-th equation $\boldsymbol{a}_{j*}^\mathsf{T}\boldsymbol{x} = b_j$ by $\boldsymbol{a}_{j*}^\mathsf{T}\boldsymbol{x} + c\boldsymbol{a}_{k*}^\mathsf{T}\boldsymbol{x} = b_j + cb_k$. For $j > k$, $\boldsymbol{E}_{jk}(c) = \boldsymbol{I} + c\boldsymbol{e}_j\boldsymbol{e}_k^\mathsf{T}$ is unit lower triangular and full rank. $\boldsymbol{E}_{jk}^{-1}(c) = \boldsymbol{E}_{jk}(-c)$.
- Zeroing the first column

$$\begin{aligned}
\boldsymbol{E}_{21}(c_2^{(1)})\boldsymbol{Ax} &= \boldsymbol{E}_{21}(c_2^{(1)})\boldsymbol{b} \\
\boldsymbol{E}_{31}(c_3^{(1)})\boldsymbol{E}_{21}(c_2^{(1)})\boldsymbol{Ax} &= \boldsymbol{E}_{31}(c_3^{(1)})\boldsymbol{E}_{21}(c_2^{(1)})\boldsymbol{b} \\
&\vdots \\
\boldsymbol{E}_{n1}(c_n^{(1)})\cdots\boldsymbol{E}_{31}(c_3^{(1)})\boldsymbol{E}_{21}(c_2^{(1)})\boldsymbol{Ax} &= \boldsymbol{E}_{n1}(c_n^{(1)})\cdots\boldsymbol{E}_{31}(c_3^{(1)})\boldsymbol{E}_{21}(c_2^{(1)})\boldsymbol{b}
\end{aligned}$$

  where $c_i^{(1)} = -a_{i1}/a_{11}$. Denote $\boldsymbol{M}_1 = \boldsymbol{E}_{n1}(c_n^{(1)})\cdots\boldsymbol{E}_{31}(c_3^{(1)})\boldsymbol{E}_{21}(c_2^{(1)})$.
- Then zero the k-th column for $k = 2, \ldots, n-1$ sequentially. This results in a transformed linear system $\boldsymbol{Ux} = \tilde{\boldsymbol{b}}$, where $\boldsymbol{U} = \boldsymbol{M}_{n-1}\cdots\boldsymbol{M}_1\boldsymbol{A}$ is upper triangular and $\tilde{\boldsymbol{b}} = \boldsymbol{M}_{n-1}\cdots\boldsymbol{M}_1\boldsymbol{b}$. $\boldsymbol{M}_k$ has the shape

$$\boldsymbol{M}_k = \boldsymbol{E}_{n,k}^{(k)}\cdots\boldsymbol{E}_{k+1,k}^{(k)} = \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & c_{k+1}^{(k)} & 1 & \\ & & \vdots & & \ddots \\ & & c_n^{(k)} & & 1 \end{pmatrix},$$

  where $c_i^{(k)} = -\tilde{a}_{ik}^{(k-1)}/\tilde{a}_{kk}^{(k-1)}$. $\boldsymbol{M}_k$ is unit lower triangular and full rank. $\boldsymbol{M}_k$ are called the *Gauss transformations*.
- Let $\boldsymbol{L} = \boldsymbol{M}_1^{-1}\cdots\boldsymbol{M}_{n-1}^{-1}$. We have the decomposition

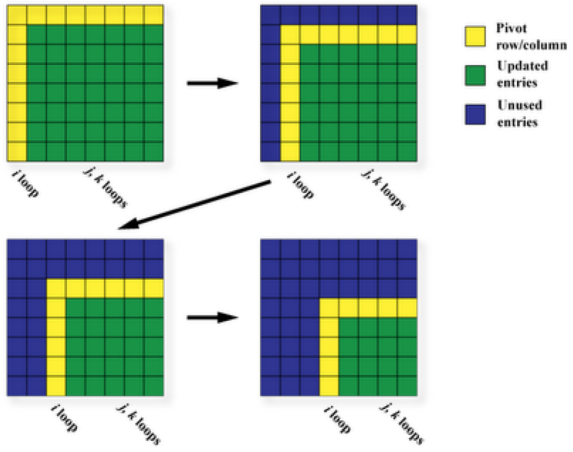
$$\boldsymbol{A} = \boldsymbol{LU}.$$

  $\boldsymbol{M}_k$ is unit upper triangular, so $\boldsymbol{M}_k^{-1}$ and thus $\boldsymbol{L}$ is unit lower triangular.

- Where is $\boldsymbol{L}$? Note $\boldsymbol{M}_k = \boldsymbol{I} + (0, \ldots, 0, c_{k+1}^{(k)}, \ldots, c_n^{(k)})^\mathsf{T} \boldsymbol{e}_k^\mathsf{T}$. By Sherman-Morrison, $\boldsymbol{M}_k^{-1} = \boldsymbol{I} - (0, \ldots, 0, c_{k+1}^{(k)}, \ldots, c_n^{(k)})^\mathsf{T} \boldsymbol{e}_k^\mathsf{T}$. So the entries of $\boldsymbol{L}$ are simply $\ell_{ik} = -c_i^{(k)}$, $i > k$, the negative of multipliers in GE.

- The whole LU procedure is done in place, i.e., $\boldsymbol{A}$ is overwritten by $\boldsymbol{L}$ and $\boldsymbol{U}$.

- Implementation: outer product LU (kij loop), block outer product LU (higher level-3 fraction), Crout's algorithm (jki loop), ...



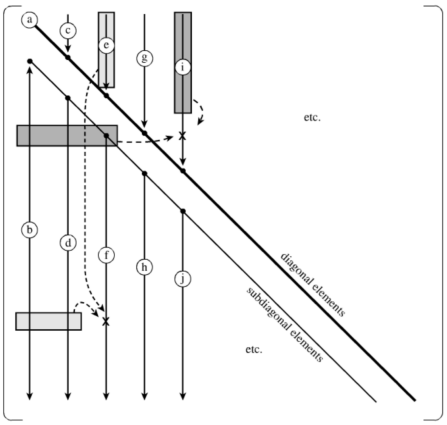


Figure 2.3.1. Crout's algorithm for *LU* decomposition of a matrix. Elements of the original matrix are modified in the order indicated by lower case letters: a, b, c, etc. Shaded boxes show the previously modified elements that are used in modifying two typical elements, each indicated by an "x".

- LU decomposition exists if the principal sub-matrix $A(1:k, 1:k)$ is non-singular for $k = 1, \ldots, n-1$. If the LU decomposition exists and $\boldsymbol{A}$ is non-singular, then the LU decomposition is unique and $\det(\boldsymbol{A}) = \prod_{i=1}^n u_{ii}$.

- This *LU decomposition* costs $2(n-1)^2 + 2(n-2)^2 + \cdots + 2 \cdot 1^2 \approx \frac{2}{3}n^3$ flops ($n^3/3$ multiplications and $n^3/3$ additions).

- Given LU, one right hand side costs $2n^2$ flops (one backward substitution and then one forward substitution).

- For *matrix inversion*, there are n right hand sides $\boldsymbol{e}_i$. However, taking advantage of zeros reduces $2n^3$ flops to $\frac{4}{3}n^3$. So matrix inversion costs $\frac{2}{3}n^3 + \frac{4}{3}n^3 = 2n^3$ flops in total.

- *No inversion* mentality: Whenever we see matrix inverse, we should think in terms of solving linear equations. We do *not* compute matrix inverse unless (i) it is necessary to compute standard errors, (2) number of right hand sides is much larger than n, (3) n is small.

- LU decomposition of a rectangular matrix $\boldsymbol{A} \in \mathbf{R}^{m \times n}$ exists if $\boldsymbol{A}(1:k, 1:k)$ is non-singular for $k = \min\{m, n\}$. For example,

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 3 & 1 \\ 5 & 2 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ 0 & -2 \end{pmatrix}$$

  and

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 4 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 \\ 0 & -3 & -6 \end{pmatrix}.$$

  Slight modification to the algorithm.

Pivoting for LU

- What if we encounter a *pivot* $\tilde{a}_{kk}^{(k-1)}$ being 0 or close to 0 due to underflow?

- Think about $\boldsymbol{A} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$. Does it have a solution for arbitrary $\boldsymbol{b}$? Does GE work?

- Work on the example

$$\begin{aligned} 0.0001x_1 + x_2 &= 1 \\ x_1 + x_2 &= 2, \end{aligned}$$

  which has solution $x_1 = 1.0001$ and $x_2 = 0.9999$. Suppose we have 3 digits of precision. After first step of elimination, we have (catastrophic cancellation happens)

$$\begin{aligned} 0.0001x_1 + x_2 &= 1 \\ -10,000x_2 &= -10,000 \end{aligned}$$

  and the solution by back substitution is $x_2 = 1.000$ and $x_1 = 0.000$.

- Message: zero or very small pivots cause trouble.
  Solution: *pivoting*.

- *Partial pivoting*: at the k-th stage the equation with $\max_{i=k}^{n} |\tilde{a}_{ik}^{(k-1)}|$ is moved into the k-th row. Thus we have $\boldsymbol{M}_{n-1}\boldsymbol{P}_{n-1}\cdots\boldsymbol{M}_1\boldsymbol{P}_1\boldsymbol{A} = \boldsymbol{U}$.

- With partial pivoting, it can be shown that

  $$\boldsymbol{PA} = \boldsymbol{LU},$$

  where $\boldsymbol{P} = \boldsymbol{P}_{n-1}\cdots\boldsymbol{P}_1$, $\boldsymbol{L}$ is unit lower triangular with $|\ell_{ij}| \le 1$, and $\boldsymbol{U}$ is upper triangular.

- $\det(\boldsymbol{P})\det(\boldsymbol{A}) = \det(\boldsymbol{U}) = \prod_{i=1}^{n} u_{ii}$.

- To solve $\boldsymbol{Ax} = \boldsymbol{b}$, we solve two triangular systems

  $$\boldsymbol{Ly} = \boldsymbol{Pb} \quad \text{and} \quad \boldsymbol{Ux} = \boldsymbol{y},$$

  costing $2n^2$ flops.

- *Complete pivoting*: Do both row and column interchanges so that the largest entry in the sub matrix $\boldsymbol{A}(k:n, k:n)$ is permuted to the (k,k)-th entry. This yields the decomposition $\boldsymbol{PAQ} = \boldsymbol{LU}$, where $|\ell_{ij}| \le 1$.

- Warning: In the actual LAPACK implementation, we do not really need to interchange rows/columns for pivoting. Just keep track of the indices we have interchanged.

- Gaussian elimination with partial pivoting is one of the most commonly used methods for solving general linear systems. Complete pivoting is stable but costs more computation. Partial pivoting is stable most of times.

- LAPACK: `?GETRF` does $\boldsymbol{PA} = \boldsymbol{LU}$ (LU decomposition with partial pivoting) in place.

- In R, `solve()` implicitly performs LU decomposition (wrapper of LAPACK routine `DGESV`). `solve()` allows specifying a single or multiple right hand sides. If none, it computes the matrix inverse. The `matrix` package contains `lu()` function that outputs `L`, `U`, and `pivot`.

- In Julia, `lu()` and `lufact()` perform LU decomposition with partial pivoting, or call LAPACK function directly using `getrf!()`.

Cholesky decomposition (symmetric LU)

Reference: KL 7.7.

Archives de l'Ecole polytechnique (Fonds A. Cholesky)

- A basic tenet in numerical analysis:

  > The structure should be exploited whenever solving a problem.

  Common structures include: symmetry, definiteness, sparsity, Kronecker product, low rank, ...

- Consider solving the normal equation $\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\beta} = \boldsymbol{X}^T\boldsymbol{y}$ for linear regression. The coefficient matrix $\boldsymbol{X}^T\boldsymbol{X}$ is symmetric and positive semidefinite, how to exploit this structure?

- Theorem: Let $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ be symmetric and positive definite. Then $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{L}^\intercal$, where $\boldsymbol{L}$ is lower triangular with positive diagonal entries and is unique.

  Proof by induction. If $n = 1$, then $\ell = \sqrt{a}$. For $n > 1$, the block equation

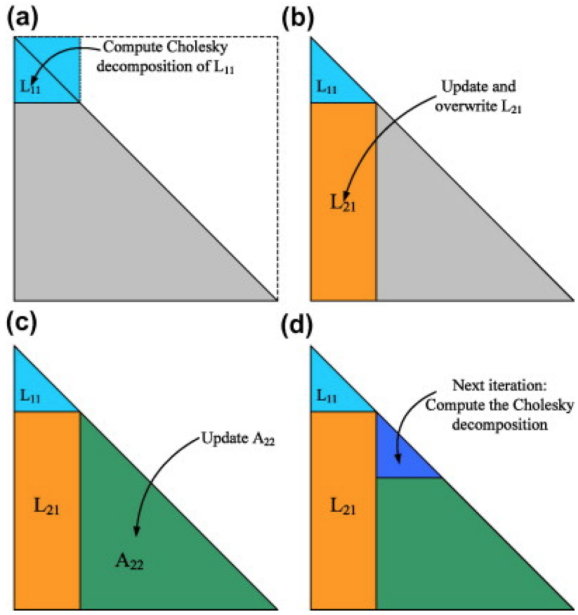
$$\begin{pmatrix} a_{11} & \boldsymbol{a}^\intercal \\ \boldsymbol{a} & \boldsymbol{A}_{22} \end{pmatrix} = \begin{pmatrix} \ell_{11} & \boldsymbol{0}_{n-1}^\intercal \\ \boldsymbol{l} & \boldsymbol{L}_{22} \end{pmatrix} \begin{pmatrix} \ell_{11} & \boldsymbol{l}^\intercal \\ \boldsymbol{0}_{n-1} & \boldsymbol{L}_{22}^\intercal \end{pmatrix}.$$

has solution

$$\begin{aligned} \ell_{11} &= \sqrt{a_{11}} \\ \boldsymbol{l} &= \ell_{11}^{-1}\boldsymbol{a} \\ \boldsymbol{L}_{22}\boldsymbol{L}_{22}^{\mathsf{T}} &= \boldsymbol{A}_{22} - \boldsymbol{l}\boldsymbol{l}^{\mathsf{T}} = \boldsymbol{A}_{22} - a_{11}^{-1}\boldsymbol{a}\boldsymbol{a}^{\mathsf{T}}. \end{aligned}$$

Now $a_{11} > 0$ (why?), so ℓ_{11} and $\boldsymbol{l}$ are uniquely determined. $\boldsymbol{A}_{22} - a_{11}^{-1}\boldsymbol{a}\boldsymbol{a}^{\mathsf{T}}$ is positive definite because $\boldsymbol{A}$ is positive definite (why?). By induction hypothesis, $\boldsymbol{L}_{22}$ exists and is unique. □

- The constructive proof completely specifies the algorithm.



- Computational cost: $\frac{1}{2}[2(n-1)^2 + 2(n-2)^2 + \cdots + 2\cdot 1^2] \approx \frac{1}{3}n^3$ (half the cost of LU decomposition due to symmetry) plus n square roots.
- Avoid square roots: $\boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^{\mathsf{T}}$ decomposition.
- In general Cholesky decomposition is very stable. Failure of the decomposition simply means $\boldsymbol{A}$ is not positive definite. It is an efficient way to test positive definiteness.
- Any matrix $\boldsymbol{X} \in \mathbb{R}^{n\times n}$ is a *square root* of $\boldsymbol{A} \succeq \boldsymbol{0}_{n\times n}$ if $\boldsymbol{A} = \boldsymbol{X}\boldsymbol{X}$. Note that Cholesky factor is *not* a square root of $\boldsymbol{A}$ unless $\boldsymbol{A}$ is diagonal.

Cholesky with symmetric pivoting

Assume $\boldsymbol{A} \succeq \boldsymbol{0}_{n \times n}$ (p.s.d.)

- When $\boldsymbol{A}$ does not have full rank, e.g., $\boldsymbol{X}^T\boldsymbol{X}$ with a non-full column rank $\boldsymbol{X}$, we encounter $\tilde{a}_{kk} = 0$ during the procedure.
- *Symmetric pivoting.* At each stage k, we permute both row and column such that $\max_{i=k}^{n} \tilde{a}_{kk}$ becomes the pivot. If we encounter $\max_{i=k}^{n} \tilde{a}_{kk} = 0$, then $A(k : n, k : n) = \boldsymbol{0}$ (why?) and the algorithm terminates.
- With symmetric pivoting: $\boldsymbol{P}\boldsymbol{A}\boldsymbol{P}^T = \boldsymbol{L}\boldsymbol{L}^T$, where $\boldsymbol{P}$ is a permutation matrix and $\boldsymbol{L} \in \mathbb{R}^{n \times r}$, $r = \text{rank}(\boldsymbol{A})$.
- In R, `chol()` is a wrapper function for LAPACK routines `DPOTRF` (p.d. no pivoting) and `DPSTRF` (p.s.d. with pivoting).
  - Option `pivot = FALSE` calls `DPOTRF`. It does $\boldsymbol{A} = \boldsymbol{R}^T\boldsymbol{R}$ and gives error message if $\boldsymbol{A}$ is not full rank.
  - Option `pivot = TRUE` calls `DPSTRF`. It does symmetric pivoting $\boldsymbol{P}\boldsymbol{A}\boldsymbol{P}^T = \boldsymbol{R}^T\boldsymbol{R}$ and yields `rank` and `pivot`.
  - Option `tol` passes the tolerance to LAPACK for deciding zero pivots. Default is $n \cdot \text{machine epsilon} \cdot \max(\text{diag}(\boldsymbol{A}))$.
- In Julia, `chol()`, `cholfact()`, `ldlfact()`, or call LAPACK functions directly.

Applications of Cholesky decomposition

There are numerous applications of Cholesky decomposition.

- *No inversion* mentality: Whenever we see matrix inverse, we should think in terms of solving linear equations. If the matrix is positive (semi)definite, Cholesky decomposition applies.
- Example: multivariate normal density $N_n(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, $\boldsymbol{\Sigma}$ is p.d.

$$-\frac{n}{2}\ln(2\pi) - \frac{1}{2}\ln\det\boldsymbol{\Sigma} - \frac{1}{2}(\boldsymbol{y}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{y}-\boldsymbol{\mu}).$$

  - Method 1: (a) compute explicit inverse $\boldsymbol{\Sigma}^{-1}$ ($2n^3$ flops), (b) compute quadratic form ($2n^2 + 2n$ flops), (c) compute determinant ($2n^3/3$ flops).
  - Method 2: (a) Cholesky decomposition $\boldsymbol{\Sigma} = \boldsymbol{L}\boldsymbol{L}^T$ ($n^3/3$ flops), (b) Solve $\boldsymbol{L}\boldsymbol{x} = \boldsymbol{y} - \boldsymbol{\mu}$ by forward substitutions (n^2 flops), (c) compute quadratic form $\boldsymbol{x}^T\boldsymbol{x}$ ($2n$ flops), and (d) compute determinant from Cholesky factor (n flops).

  Which method is better?

- Compute Moore-Penrose inverse $\boldsymbol{A}^+$.
- Linear regression.

Linear regression by Cholesky (method of normal equations)

Assume $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ has full column rank. (For rank deficient $\boldsymbol{X}$, use Cholesky with symmetric pivoting.)

- It is easier to work on the augmented matrix

$$\begin{pmatrix} \boldsymbol{X}^\top \boldsymbol{X} & \boldsymbol{X}^\top \boldsymbol{y} \\ \boldsymbol{y}^\top \boldsymbol{X} & \boldsymbol{y}^\top \boldsymbol{y} \end{pmatrix} = \begin{pmatrix} \boldsymbol{L} & \boldsymbol{0} \\ \boldsymbol{l}^\top & d \end{pmatrix} \begin{pmatrix} \boldsymbol{L}^\top & \boldsymbol{l} \\ \boldsymbol{0}^\top & d \end{pmatrix} = \begin{pmatrix} \boldsymbol{L}\boldsymbol{L}^\top & \boldsymbol{L}\boldsymbol{l} \\ \boldsymbol{l}^\top \boldsymbol{L}^\top & \|\boldsymbol{l}\|_2^2 + d^2 \end{pmatrix}.$$

  Normal equation implies the equation

$$\boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{\beta} = \boldsymbol{L}\boldsymbol{L}^\top \boldsymbol{\beta} = \boldsymbol{X}^\top \boldsymbol{y} = \boldsymbol{L}\boldsymbol{l} \text{ or } \boldsymbol{L}^\top \boldsymbol{\beta} = \boldsymbol{l},$$

  which we can solve for $\boldsymbol{\beta}$ in p^2 flops. Since $\boldsymbol{l} = \boldsymbol{L}^{-1}\boldsymbol{X}^\top \boldsymbol{y}$, we have

$$\boldsymbol{l}^\top \boldsymbol{l} = \boldsymbol{y}\boldsymbol{X}(\boldsymbol{L}\boldsymbol{L}^\top)^{-1}\boldsymbol{X}^\top \boldsymbol{y} = \boldsymbol{y}^\top \boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{X})^{-1}\boldsymbol{X}^\top \boldsymbol{y} = \boldsymbol{y}^\top \boldsymbol{P}_{\boldsymbol{X}} \boldsymbol{y} = \|\hat{\boldsymbol{y}}\|_2^2$$

  and

$$d^2 = \boldsymbol{y}^\top \boldsymbol{y} - \boldsymbol{l}^\top \boldsymbol{l} = \boldsymbol{y}^\top (\boldsymbol{I} - \boldsymbol{P}_{\boldsymbol{X}})\boldsymbol{y} = \|\boldsymbol{y} - \hat{\boldsymbol{y}}\|_2^2 = \text{SSE}.$$

  If standard errors are needed, we do inversion $(\boldsymbol{X}^T\boldsymbol{X})^{-1} = (\boldsymbol{L}\boldsymbol{L}^T)^{-1} = \boldsymbol{L}^{-T}\boldsymbol{L}^{-1}$. Use `chol2inv()` in R function for this purpose.

- In summary, linear regression by Cholesky, aka the method of normal equations:
  - Form the lower triangular part of $(\boldsymbol{X}, \boldsymbol{y})^T(\boldsymbol{X}, \boldsymbol{y})$ ($n(p+1)^2$ flops)

- Cholesky decomposition of the augmented system $\begin{pmatrix} \boldsymbol{X}^\top \boldsymbol{X} & \boldsymbol{X}^\top \boldsymbol{y} \\ \boldsymbol{y}^\top \boldsymbol{X} & \boldsymbol{y}^\top \boldsymbol{y} \end{pmatrix}$ ($(p+1)^3/3$ flops)
- Solve $\boldsymbol{L}^T \boldsymbol{\beta} = \boldsymbol{l}$ for regression coefficients $\hat{\boldsymbol{\beta}}$ (p^2 flops)
- If want standard errors, estimate σ^2 by $\hat{\sigma}^2 = d^2/(n-p)$ and compute $\hat{\sigma}^2(\boldsymbol{X}^T\boldsymbol{X})^{-1} = \hat{\sigma}^2(\boldsymbol{L}\boldsymbol{L}^T)^{-1}$ ($4p^3/3$ flops)

Total cost is $p^3/3 + np^2$ flops (without s.e.) or $5p^3/3 + np^2$ flops (with s.e.).

- Remark: The Cholesky decomposition approach is incremental (or online) by nature and works for huge data sets with large n and moderate p. We can keep updating the Gram matrix $\boldsymbol{X}^T\boldsymbol{X}$ with newest data points and then re-do Cholesky for updated regression coefficients.