

Boardwork for “Approximate Gaussian process inference”

Stephen Huan

April 8, 2025

Throughout, we'll assume $\Theta \in \mathbb{R}^{N \times N}$ is (symmetric) positive semi-definite (psd).

1 Linear algebraic quantities

Matrix square root $\Theta = LL^\top$. Not unique, take orthonormal Q ($Q^\top Q = QQ^\top = \text{Id}$),

$$(LQ)(LQ)^\top = L(QQ^\top)L^\top = LL^\top = \Theta,$$

so if L is a square root then LQ is as well. The Cholesky factor is the unique lower triangular square root (with positive diagonal). Another convention is $\Theta = LL = L^2$. Take eigendecomposition $\Theta = U\Lambda U^\top$, $L = U\Lambda^{1/2}U^\top$ suffices as

$$L^2 = (U\Lambda^{1/2}U^\top)(U\Lambda^{1/2}U^\top) = U\Lambda^{1/2}(U^\top U)\Lambda^{1/2}U^\top = U\Lambda^{1/2}\Lambda^{1/2}U^\top = U\Lambda U^\top = \Theta.$$

But also $U\Lambda^{1/2}SU^\top$ works where S is a diagonal matrix with ± 1 on the diagonal. There is, however, a unique psd square root $\Theta = L^2$, which can be computed by $U\Lambda^{1/2}U^\top$, called the *principle* or *positive* square root of Θ . Note that by symmetry, $\Theta = L^2 = LL^\top$ so it is also a square root in the first sense.

2 Schur complement

Recall: LU factorization. Perform row operations to reduce to reduced row echelon form, collect row operations into matrices, resulting factorization lower and upper triangular.

For Θ blocked as $\Theta = \begin{pmatrix} \Theta_{1,1} & \Theta_{1,2} \\ \Theta_{2,1} & \Theta_{2,2} \end{pmatrix}$, multiply the first row by $-\Theta_{2,1}\Theta_{1,1}^{-1}$ and add to the second row to eliminate the first entry in the second row ($\Theta_{2,1}$). Resulting matrix is

$$\begin{pmatrix} \Theta_{1,1} & \Theta_{1,2} \\ \mathbf{0} & \Theta_{2,2} - \Theta_{2,1}\Theta_{1,1}^{-1}\Theta_{1,2} \end{pmatrix}.$$

Now multiply the first column by $-\Theta_{1,1}^{-1}\Theta_{1,2}$ and add to the second column to eliminate the first entry of the second column ($\Theta_{1,2}$). The resulting matrix is now (block) diagonal,

$$\begin{pmatrix} \Theta_{1,1} & \mathbf{0} \\ \mathbf{0} & \Theta_{2,2|1} \end{pmatrix},$$

where we denote the *Schur complement* with the notation $\Theta_{2,2|1} := \Theta_{2,2} - \Theta_{2,1}\Theta_{1,1}^{-1}\Theta_{1,2}$. Collecting the row and column operations into lower and upper triangular matrices,

$$\Theta = \begin{pmatrix} \text{Id} & \mathbf{0} \\ \Theta_{2,1}\Theta_{1,1}^{-1} & \text{Id} \end{pmatrix} \begin{pmatrix} \Theta_{1,1} & \mathbf{0} \\ \mathbf{0} & \Theta_{2,2|1} \end{pmatrix} \begin{pmatrix} \text{Id} & \Theta_{1,1}^{-1}\Theta_{1,2} \\ \mathbf{0} & \text{Id} \end{pmatrix}$$

which is a (block) LDL^\top factorization. Recursing finishes the construction,

$$\begin{aligned} \text{chol}(\Theta) &= \begin{pmatrix} \text{Id} & \mathbf{0} \\ \Theta_{2,1}\Theta_{1,1}^{-1} & \text{Id} \end{pmatrix} \begin{pmatrix} \text{chol}(\Theta_{1,1}) & \mathbf{0} \\ \mathbf{0} & \text{chol}(\Theta_{2,2|1}) \end{pmatrix} \\ &= \begin{pmatrix} \text{chol}(\Theta_{1,1}) & \mathbf{0} \\ \Theta_{2,1}\text{chol}(\Theta_{1,1})^{-\top} & \text{chol}(\Theta_{2,2|1}) \end{pmatrix} \end{aligned}$$

where we use that $\Theta_{1,1} = \text{chol}(\Theta_{1,1})\text{chol}(\Theta_{1,1})^\top$ by definition of the Cholesky factor so $\Theta_{1,1}^{-1} = \text{chol}(\Theta_{1,1})^{-\top}\text{chol}(\Theta_{1,1})^{-1}$, and so

$$(\Theta_{2,1}\Theta_{1,1}^{-1})\text{chol}(\Theta_{1,1}) = \Theta_{2,1}(\text{chol}(\Theta_{1,1})^{-\top}\text{chol}(\Theta_{1,1})^{-1})\text{chol}(\Theta_{1,1}) = \Theta_{2,1}\text{chol}(\Theta_{1,1})^{-\top}.$$

3 Our desired quantities, revisited

Some quick calculations from $\Theta = LL^\top$:

$$\begin{aligned} \Theta \mathbf{x} &= (LL^\top)\mathbf{x} = L(L^\top \mathbf{x}) \\ \Theta^{-1}\mathbf{x} &= (LL^\top)^{-1}\mathbf{x} = (L^{-\top}L^{-1})\mathbf{x} = L^{-\top}(L^{-1}\mathbf{x}) \end{aligned}$$

$$\log\det(\Theta) = \log\det(LL^\top) = \log(\det(L)\det(L^\top)) = 2\log\det(L) = 2\sum_{i=1}^N \log(L_{i,i})$$

where we used associativity of matrix multiplication, the definition of the log determinant $\log\det(\cdot) := \log(\det(\cdot))$, the fact that the determinant of the product is the product of the determinants, that the determinant of the transpose is the same as the original matrix, and that the determinant of a triangular matrix is the product of its diagonal entries.

4 Sampling & determinant

How to define log of a matrix? In general, scalar functions of a matrix defined by their Taylor series, say. For example,

$$\exp(x) = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$$

Natural generalization to matrices via

$$\exp(A) = \text{Id} + A + \frac{A^2}{2!} + \frac{A^3}{3!} + \dots$$

as addition of matrices, product of matrices, and division by a scalar still well-defined (with additive and multiplicative identity Id instead of 1). Another perspective is from the theory of linear ordinary differential equations. Consider the initial value problem

$$\frac{dx}{dt} = ax, \quad x(0) = x_0.$$

The unique solution is

$$x(t) = \exp(at)x_0.$$

Generalizing to matrices, we have the multivariate problem

$$\frac{d\mathbf{x}}{dt} = A\mathbf{x}, \quad \mathbf{x}(0) = \mathbf{x}_0$$

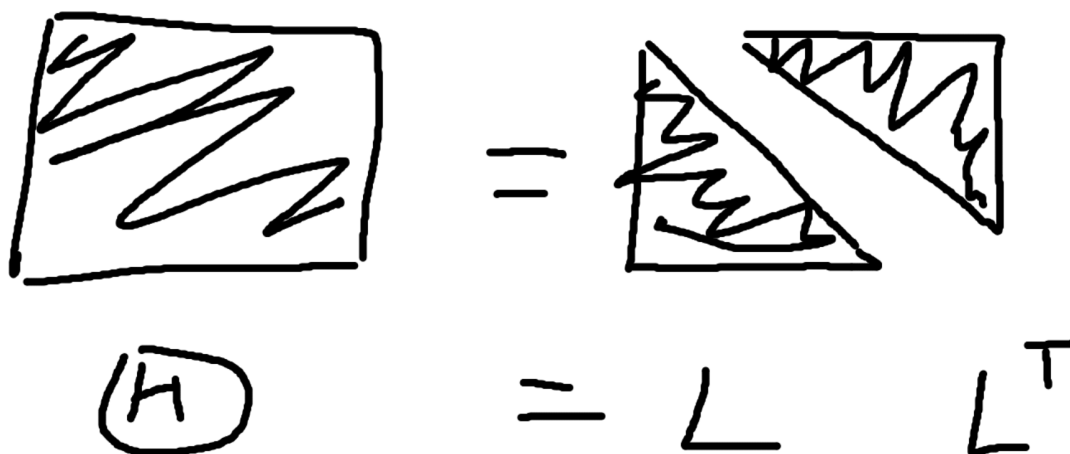
with unique solution

$$\mathbf{x}(t) = \exp(At)\mathbf{x}_0,$$

where the matrix exponential $\exp(At)$ is defined in the Taylor series sense above. Finally, we can define the matrix logarithm as the matrix whose exponential is the original matrix. For a psd matrix, this always exists and is unique. To show $\text{logdet}(\Theta) = \text{trace}(\text{log}(\Theta))$, we can use Jacobi's formula $\det(\exp(A)) = \exp(\text{trace}(A))$ for any A (not necessarily psd). Substituting $A = \text{log}(\Theta)$, we have $\det(\exp(\text{log}(\Theta))) = \det(\Theta) = \exp(\text{trace}(\text{log}(\Theta)))$ or that $\text{logdet}(\Theta) = \text{trace}(\text{log}(\Theta))$ as desired. To prove Jacobi's formula we can use the Jordan normal form and the Taylor series definition of the matrix exponential; this closely mirrors the simpler proof when A is diagonalizable.

5 Nyström method

Recall that the Cholesky factorization $\Theta = LL^\top$ looks something like the below.



$$(H) = L L^\top$$

Figure 1: Cholesky factorization $\Theta = LL^\top$.

The Nyström approximation takes the first $|X| = m$ columns of L . For slightly technical reasons we will prefer to work with factors of the *precision* Θ^{-1} , which effectively reverses the order of the variables (and transposes the matrices). This is depicted below.

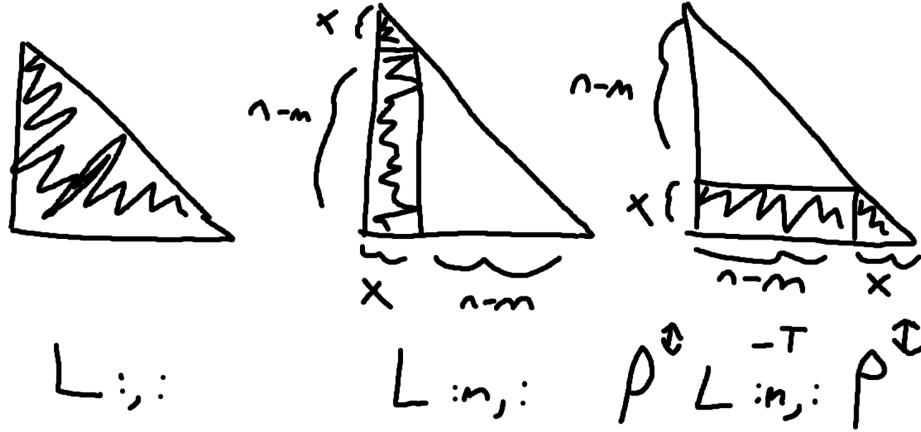


Figure 2: Nyström approximation $\Theta \approx L_{:, m, :} L_{:, m, :}^\top$.

6 Kullback-Leibler divergence

We visualize the formula $L_{s_i, i} = \frac{\Theta_{s_i, s_i}^{-1} e_1}{\sqrt{e_1^\top \Theta_{s_i, s_i}^{-1} e_1}}$, recalling that the first entry of the (sorted list) s_i is i since sparsity patterns must include the diagonal to have finite KL divergence.

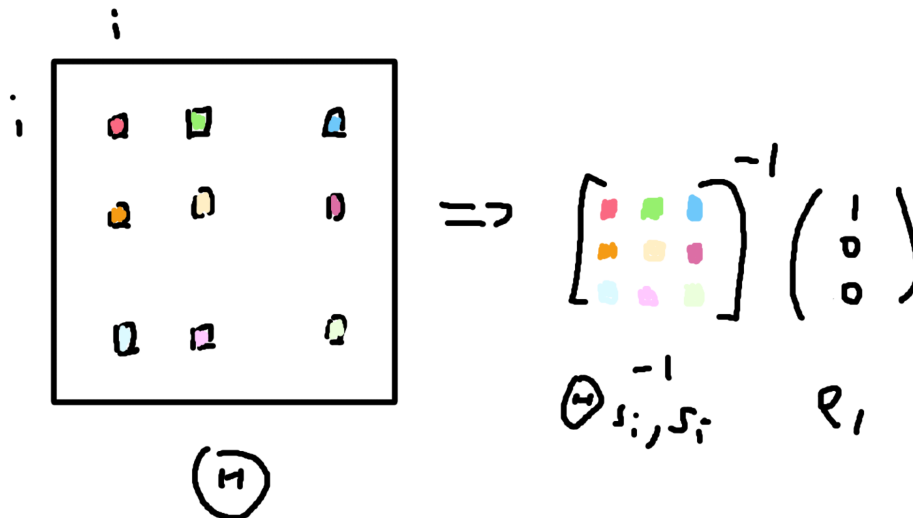


Figure 3: Column formed from the submatrix $\Theta_{s_i, s_i}^{-1} e_1$.

7 Gaussian process regression

Recall the conditional formulas

$$\begin{aligned}\mathbb{E}[\mathbf{y}_{\text{Pr}} \mid \mathbf{y}_{\text{Tr}}] &= \boldsymbol{\mu}_{\text{Pr}} + \Theta_{\text{Pr},\text{Tr}} \Theta_{\text{Tr},\text{Tr}}^{-1} (\mathbf{y}_{\text{Tr}} - \boldsymbol{\mu}_{\text{Tr}}) \\ \mathbb{Cov}[\mathbf{y}_{\text{Pr}} \mid \mathbf{y}_{\text{Tr}}] &= \Theta_{\text{Pr},\text{Pr}} - \Theta_{\text{Pr},\text{Tr}} \Theta_{\text{Tr},\text{Tr}}^{-1} \Theta_{\text{Tr},\text{Pr}}.\end{aligned}$$

But we only have an efficient approximation of the Cholesky factor of psd matrices Θ . How do we efficiently compute quantities that involve $\Theta_{\text{Pr},\text{Tr}}$ (which is not even square)? One slightly inefficient thing we can do would be to form a block *joint* covariance matrix

$$\Theta := \begin{pmatrix} \Theta_{\text{Tr},\text{Tr}} & \Theta_{\text{Tr},\text{Pr}} \\ \Theta_{\text{Pr},\text{Tr}} & \Theta_{\text{Pr},\text{Pr}} \end{pmatrix}, \quad \Theta_{\text{Pr},\text{Tr}} \mathbf{x} = \begin{pmatrix} \mathbf{0} & \text{Id} \end{pmatrix} \Theta \begin{pmatrix} \mathbf{x} \\ \mathbf{0} \end{pmatrix}$$

or in other words, extracts a product $\Theta_{\text{Pr},\text{Tr}} \mathbf{x}$ by computing a product with Θ against \mathbf{x} padded with zeros and reads the desired values from the output. Although this is efficient in the sense that (approximate) products with Θ are efficient, clearly this is wasteful as many computations are discarded. Instead, we observe that Θ already naturally encodes the cross interactions. The natural order is to put the training points *before* the testing points as a Cholesky factor conditions on the previous columns. In our inverse factors, this corresponds to putting “prediction points first”. Indeed, under this ordering we can efficiently read off the quantities from the Cholesky factor with no waste.

$$\begin{aligned}\mathbb{E}[\mathbf{y}_{\text{Pr}} \mid \mathbf{y}_{\text{Tr}}] &= -L_{\text{Pr},\text{Pr}}^{-\top} L_{\text{Tr},\text{Pr}}^{\top} \mathbf{y}_{\text{Tr}} \\ \mathbb{Cov}[\mathbf{y}_{\text{Pr}} \mid \mathbf{y}_{\text{Tr}}] &= L_{\text{Pr},\text{Pr}}^{-\top} L_{\text{Pr},\text{Pr}}^{-1} \\ \mathbf{e}_i^{\top} \mathbb{Cov}[\mathbf{y}_{\text{Pr}} \mid \mathbf{y}_{\text{Tr}}] \mathbf{e}_j &= (L_{\text{Pr},\text{Pr}}^{-1} \mathbf{e}_i)^{\top} (L_{\text{Pr},\text{Pr}}^{-1} \mathbf{e}_j)\end{aligned}$$