# Fast Gaussian process regression by Greedy Conditional Selection

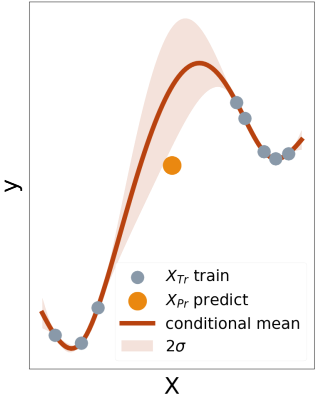
Stephen Huan    Florian Schäfer

Short & Sweet seminar

April 22, 2022

# The problem: Gaussian process regression

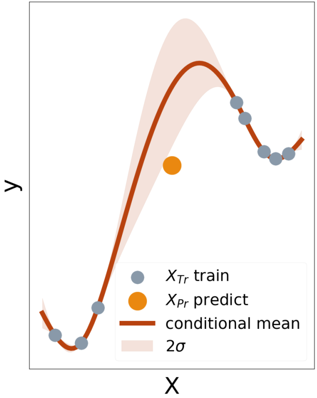
Measurements $\boldsymbol{y}_{\mathrm{Tr}}$ at $N$ points $X_{\mathrm{Tr}}$



# The problem: Gaussian process regression

Measurements $\boldsymbol{y}_{\mathrm{Tr}}$ at $N$ points $X_{\mathrm{Tr}}$

Estimate unseen data $\boldsymbol{y}_{\mathrm{Pr}}$ at $X_{\mathrm{Pr}}$

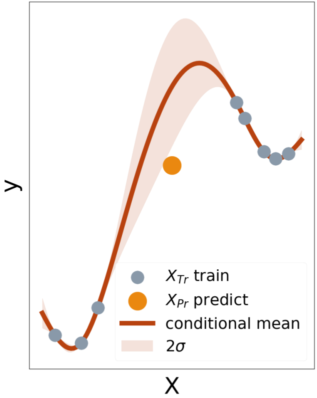


# The problem: Gaussian process regression

Measurements $\boldsymbol{y}_{\mathrm{Tr}}$ at $N$ points $X_{\mathrm{Tr}}$

Estimate unseen data $\boldsymbol{y}_{\mathrm{Pr}}$ at $X_{\mathrm{Pr}}$

Model as Gaussian process
$\rightarrow$ condition on $\boldsymbol{y}_{\mathrm{Tr}}$



# Cubic bottleneck

Closed-form conditional distribution:

$$\mathrm{E}[\boldsymbol{y}_{\mathsf{Pr}} \mid \boldsymbol{y}_{\mathsf{Tr}}] = \boldsymbol{\mu}_{\mathsf{Pr}} + \Theta_{\mathsf{Pr},\mathsf{Tr}} \Theta_{\mathsf{Tr},\mathsf{Tr}}^{-1} (\boldsymbol{y}_{\mathsf{Tr}} - \boldsymbol{\mu}_{\mathsf{Tr}})$$
$$\Theta_{\mathsf{Pr},\mathsf{Pr} \mid \mathsf{Tr}} := \mathrm{Cov}[\boldsymbol{y}_{\mathsf{Pr}} \mid \boldsymbol{y}_{\mathsf{Tr}}] = \Theta_{\mathsf{Pr},\mathsf{Pr}} - \Theta_{\mathsf{Pr},\mathsf{Tr}} \Theta_{\mathsf{Tr},\mathsf{Tr}}^{-1} \Theta_{\mathsf{Tr},\mathsf{Pr}}$$

# Cubic bottleneck

Closed-form conditional distribution:

$$\mathrm{E}[\boldsymbol{y}_{\mathsf{Pr}} \mid \boldsymbol{y}_{\mathsf{Tr}}] = \boldsymbol{\mu}_{\mathsf{Pr}} + \Theta_{\mathsf{Pr},\mathsf{Tr}} \Theta_{\mathsf{Tr},\mathsf{Tr}}^{-1} (\boldsymbol{y}_{\mathsf{Tr}} - \boldsymbol{\mu}_{\mathsf{Tr}})$$

$$\Theta_{\mathsf{Pr},\mathsf{Pr}|\mathsf{Tr}} := \mathrm{Cov}[\boldsymbol{y}_{\mathsf{Pr}} \mid \boldsymbol{y}_{\mathsf{Tr}}] = \Theta_{\mathsf{Pr},\mathsf{Pr}} - \Theta_{\mathsf{Pr},\mathsf{Tr}} \Theta_{\mathsf{Tr},\mathsf{Tr}}^{-1} \Theta_{\mathsf{Tr},\mathsf{Pr}}$$

Kernel function $K(\cdot, \cdot)$: $\Theta_{i,j} := K(\boldsymbol{x}_i, \boldsymbol{x}_j)$

# Cubic bottleneck

Closed-form conditional distribution:

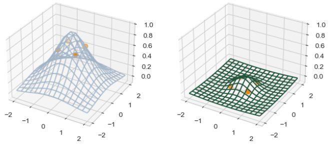
$$\mathrm{E}[\boldsymbol{y}_{\mathsf{Pr}} \mid \boldsymbol{y}_{\mathsf{Tr}}] = \boldsymbol{\mu}_{\mathsf{Pr}} + \Theta_{\mathsf{Pr},\mathsf{Tr}} \Theta_{\mathsf{Tr},\mathsf{Tr}}^{-1} (\boldsymbol{y}_{\mathsf{Tr}} - \boldsymbol{\mu}_{\mathsf{Tr}})$$
$$\Theta_{\mathsf{Pr},\mathsf{Pr} \mid \mathsf{Tr}} := \mathrm{Cov}[\boldsymbol{y}_{\mathsf{Pr}} \mid \boldsymbol{y}_{\mathsf{Tr}}] = \Theta_{\mathsf{Pr},\mathsf{Pr}} - \Theta_{\mathsf{Pr},\mathsf{Tr}} \Theta_{\mathsf{Tr},\mathsf{Tr}}^{-1} \Theta_{\mathsf{Tr},\mathsf{Pr}}$$

Kernel function $K(\cdot,\cdot)$: $\Theta_{i,j} := K(\boldsymbol{x}_i, \boldsymbol{x}_j)$

Computational cost scales as $N^3$

# Screening effect

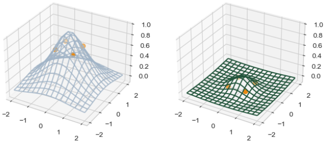
“Screening effect”



# Screening effect

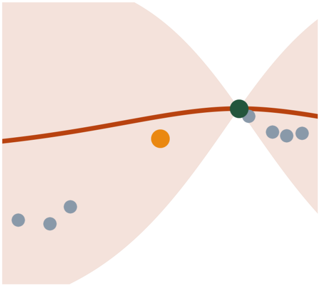
"Screening effect"

Choose $k$ most informative points

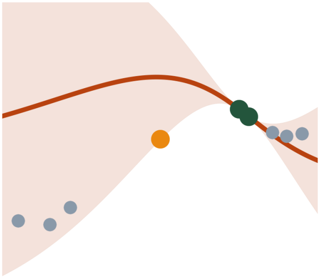


# Conditional $k$-th nearest neighbors

Naive: select $k$ closest points



# Conditional $k$-th nearest neighbors



Naive: select $k$ closest points

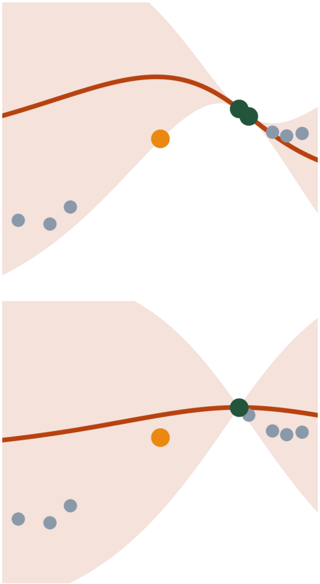
Chooses redundant information

# Conditional $k$-th nearest neighbors

Naive: select $k$ closest points

Chooses redundant information

Maximize *mutual information*!

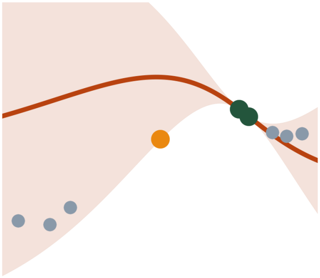


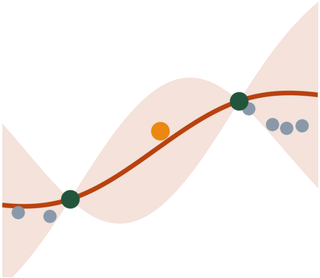
# Conditional $k$-th nearest neighbors

Naive: select $k$ closest points

Chooses redundant information

Maximize *mutual information*!





# Mutual information

Mutual information or information gain:

$$\mathrm{I}[\boldsymbol{y}_{\mathsf{Pr}}; \boldsymbol{y}_{\mathsf{Tr}}] = \mathrm{H}[\boldsymbol{y}_{\mathsf{Pr}}] - \mathrm{H}[\boldsymbol{y}_{\mathsf{Pr}} \mid \boldsymbol{y}_{\mathsf{Tr}}]$$

# Mutual information

Mutual information or information gain:

$$\mathrm{I}[\boldsymbol{y}_{\mathsf{Pr}}; \boldsymbol{y}_{\mathsf{Tr}}] = \mathrm{H}[\boldsymbol{y}_{\mathsf{Pr}}] - \mathrm{H}[\boldsymbol{y}_{\mathsf{Pr}} \mid \boldsymbol{y}_{\mathsf{Tr}}]$$

Entropy increases with log determinant of covariance

# Mutual information

Mutual information or information gain:

$$\mathrm{I}[\boldsymbol{y}_{\mathsf{Pr}}; \boldsymbol{y}_{\mathsf{Tr}}] = \mathrm{H}[\boldsymbol{y}_{\mathsf{Pr}}] - \mathrm{H}[\boldsymbol{y}_{\mathsf{Pr}} \mid \boldsymbol{y}_{\mathsf{Tr}}]$$

Entropy increases with log determinant of covariance

Information-theoretic EV-VE identity:

$$\begin{aligned} \mathrm{H}[\boldsymbol{y}_{\mathsf{Pr}}] &= \mathrm{H}[\boldsymbol{y}_{\mathsf{Pr}} \mid \boldsymbol{y}_{\mathsf{Tr}}] + \mathrm{I}[\boldsymbol{y}_{\mathsf{Pr}}; \boldsymbol{y}_{\mathsf{Tr}}] \\ \mathrm{Var}[\boldsymbol{y}_{\mathsf{Pr}}] &= \mathrm{E}[\mathrm{Var}[\boldsymbol{y}_{\mathsf{Pr}} \mid \boldsymbol{y}_{\mathsf{Tr}}]] + \mathrm{Var}[\mathrm{E}[\boldsymbol{y}_{\mathsf{Pr}} \mid \boldsymbol{y}_{\mathsf{Tr}}]] \end{aligned}$$

# Greedy mutual information maximization

Greedy selection, maintain selected indices $I$

# Greedy mutual information maximization

Greedy selection, maintain selected indices $I$

Criterion simplifies to:

$$\operatorname*{argmax}_{j \notin I} \frac{\Theta^2_{j,\mathsf{Pr}|I}}{\Theta_{j,j|I}}$$

# Greedy mutual information maximization

Greedy selection, maintain selected indices $I$

Criterion simplifies to:

$$\underset{j \notin I}{\operatorname{argmax}} \frac{\Theta^2_{j,\mathsf{Pr}|I}}{\Theta_{j,j|I}}$$

Direct computation: $\mathcal{O}(Nk^4)$

# Greedy mutual information maximization

Greedy selection, maintain selected indices $I$

Criterion simplifies to:

$$\operatorname*{argmax}_{j \notin I} \frac{\Theta^2_{j,\mathsf{Pr}|I}}{\Theta_{j,j|I}}$$

Direct computation: $\mathcal{O}(Nk^4)$

Storing a partial Cholesky factor: $\mathcal{O}(Nk^2)$

# Conditioning as rank-one update

Key idea: assume we have $\Theta_{|I}$, rank-one update to $\Theta_{|I \cup \{k\}}$

$$\Theta_{:,:|I \cup \{k\}} = \Theta_{:,:|I} - \Theta_{:,k|I} \Theta_{k,k|I}^{-1} \Theta_{k,:|I}$$

$$\boldsymbol{u} = \frac{\Theta_{:,k|I}}{\sqrt{\Theta_{k,k|I}}}$$

$$\Theta_{|I \cup \{k\}} = \Theta_{|I} - \boldsymbol{u}\boldsymbol{u}^\top$$

# Efficient computation from Cholesky factor

Statistical interpretation of Cholesky factorization:

$$
\begin{aligned}
\operatorname{chol}(\Theta) &= \begin{pmatrix} I & 0 \\ \Theta_{2,1}\Theta_{1,1}^{-1} & I \end{pmatrix} \begin{pmatrix} \operatorname{chol}(\Theta_{1,1}) & 0 \\ 0 & \operatorname{chol}(\Theta_{2,2} - \Theta_{2,1}\Theta_{1,1}^{-1}\Theta_{1,2}) \end{pmatrix} \\
&= \begin{pmatrix} \operatorname{chol}(\Theta_{1,1}) & 0 \\ \Theta_{2,1}\operatorname{chol}(\Theta_{1,1})^{-\top} & \operatorname{chol}(\Theta_{2,2} - \Theta_{2,1}\Theta_{1,1}^{-1}\Theta_{1,2}) \end{pmatrix}
\end{aligned}
$$

# Efficient computation from Cholesky factor

Statistical interpretation of Cholesky factorization:

$$\begin{aligned}
\operatorname{chol}(\Theta) &= \begin{pmatrix} I & 0 \\ \Theta_{2,1}\Theta_{1,1}^{-1} & I \end{pmatrix} \begin{pmatrix} \operatorname{chol}(\Theta_{1,1}) & 0 \\ 0 & \operatorname{chol}(\Theta_{2,2} - \Theta_{2,1}\Theta_{1,1}^{-1}\Theta_{1,2}) \end{pmatrix} \\
&= \begin{pmatrix} \operatorname{chol}(\Theta_{1,1}) & 0 \\ \Theta_{2,1} \operatorname{chol}(\Theta_{1,1})^{-\top} & \operatorname{chol}(\Theta_{2,2} - \Theta_{2,1}\Theta_{1,1}^{-1}\Theta_{1,2}) \end{pmatrix}
\end{aligned}$$

Store partial Cholesky factor $L$

$$L_i = \frac{\Theta_{:,k|:i}}{\sqrt{\Theta_{kk|:i}}}$$

# Algorithm

Indices $I$, select $k$, $i$th iteration, have:

Conditional covariances $\Theta_{:,\mathsf{Pr}|I}$

Conditional variances $\mathrm{diag}(\Theta_{:,:|I})$

First $i-1$ columns of $L$

## Algorithm

Indices $I$, select $k$, $i$th iteration, have:

Conditional covariances $\Theta_{:,\mathsf{Pr}|I}$

Conditional variances $\operatorname{diag}(\Theta_{:,:|I})$

First $i - 1$ columns of $L$

Update $L$

$$L_{:,i} \leftarrow \Theta_{:,k} - L_{:,:i-1} L_{k,:i-1}^{\top}$$

# Algorithm

Indices $I$, select $k$, $i$th iteration, have:

Conditional covariances $\Theta_{:,\mathsf{Pr}|I}$

Conditional variances $\mathrm{diag}(\Theta_{:,:|I})$

First $i-1$ columns of $L$

Update $L$

$$L_{:,i} \leftarrow \Theta_{:,k} - L_{:,:i-1} L_{k,:i-1}^{\top}$$

Update conditional values for candidate $j$

$$\Theta_{jj|I\cup\{k\}} \leftarrow \Theta_{jj|I} - L_{j,i}^2$$
$$\Theta_{j,\mathsf{Pr}|I\cup\{k\}} \leftarrow \Theta_{j,\mathsf{Pr}|I} - L_{j,i} L_{\mathsf{Pr},i}$$

# Extending to multiple prediction points

Objective conditional log determinant of prediction points

$$\operatorname{logdet}\left(\Theta_{\mathsf{Pr},\mathsf{Pr}|I\cup\{k\}}\right) = \operatorname{logdet}\left(\Theta_{\mathsf{Pr},\mathsf{Pr}|I} - \frac{\Theta_{\mathsf{Pr},k|I}\Theta_{\mathsf{Pr},k|I}^{\top}}{\Theta_{kk|I}}\right)$$

# Extending to multiple prediction points

Objective conditional log determinant of prediction points

$$\operatorname{logdet}\left(\Theta_{\mathsf{Pr},\mathsf{Pr}|I\cup\{k\}}\right) = \operatorname{logdet}\left(\Theta_{\mathsf{Pr},\mathsf{Pr}|I} - \frac{\Theta_{\mathsf{Pr},k|I}\Theta_{\mathsf{Pr},k|I}^{\top}}{\Theta_{kk|I}}\right)$$

By the matrix determinant lemma,

$$= \operatorname{logdet}\left(\Theta_{\mathsf{Pr},\mathsf{Pr}|I}\right) + \log\left(1 - \frac{\Theta_{\mathsf{Pr},k|I}^{\top}\Theta_{\mathsf{Pr},\mathsf{Pr}|I}^{-1}\Theta_{\mathsf{Pr},k|I}}{\Theta_{kk|I}}\right)$$

$$= \operatorname{logdet}\left(\Theta_{\mathsf{Pr},\mathsf{Pr}|I}\right) + \log\left(\frac{\Theta_{kk|I} - \Theta_{k,\mathsf{Pr}|I}\Theta_{\mathsf{Pr},\mathsf{Pr}|I}^{-1}\Theta_{\mathsf{Pr},k|I}}{\Theta_{kk|I}}\right)$$

# Extending to multiple prediction points

Objective conditional log determinant of prediction points

$$\operatorname{logdet}\left(\Theta_{\mathsf{Pr},\mathsf{Pr}|I\cup\{k\}}\right) = \operatorname{logdet}\left(\Theta_{\mathsf{Pr},\mathsf{Pr}|I} - \frac{\Theta_{\mathsf{Pr},k|I}\Theta_{\mathsf{Pr},k|I}^{\top}}{\Theta_{kk|I}}\right)$$

By the matrix determinant lemma,

$$= \operatorname{logdet}\left(\Theta_{\mathsf{Pr},\mathsf{Pr}|I}\right) + \log\left(1 - \frac{\Theta_{\mathsf{Pr},k|I}^{\top}\Theta_{\mathsf{Pr},\mathsf{Pr}|I}^{-1}\Theta_{\mathsf{Pr},k|I}}{\Theta_{kk|I}}\right)$$

$$= \operatorname{logdet}\left(\Theta_{\mathsf{Pr},\mathsf{Pr}|I}\right) + \log\left(\frac{\Theta_{kk|I} - \Theta_{k,\mathsf{Pr}|I}\Theta_{\mathsf{Pr},\mathsf{Pr}|I}^{-1}\Theta_{\mathsf{Pr},k|I}}{\Theta_{kk|I}}\right)$$

By the quotient rule, we combine the conditioning:

$$= \operatorname{logdet}\left(\Theta_{\mathsf{Pr},\mathsf{Pr}|I}\right) + \log\left(\frac{\Theta_{kk|I,\mathsf{Pr}}}{\Theta_{kk|I}}\right)$$

# Algorithm for multiple prediction points

Final objective simplifies to:

$$\operatorname{logdet}\left(\Theta_{\mathsf{Pr},\mathsf{Pr}|I\cup\{k\}}\right) - \operatorname{logdet}\left(\Theta_{\mathsf{Pr},\mathsf{Pr}|I}\right) = \log\left(\frac{\Theta_{kk|I,\mathsf{Pr}}}{\Theta_{kk|I}}\right)$$

# Algorithm for multiple prediction points

Final objective simplifies to:

$$\operatorname{logdet}\left(\Theta_{\mathsf{Pr},\mathsf{Pr} \mid I \cup \{k\}}\right) - \operatorname{logdet}\left(\Theta_{\mathsf{Pr},\mathsf{Pr} \mid I}\right) = \log\left(\frac{\Theta_{kk \mid I, \mathsf{Pr}}}{\Theta_{kk \mid I}}\right)$$

Store *two* factors (one for numerator, one for denominator)

# Algorithm for multiple prediction points

Final objective simplifies to:

$$\operatorname{logdet}\left(\Theta_{\mathsf{Pr},\mathsf{Pr}|I\cup\{k\}}\right) - \operatorname{logdet}\left(\Theta_{\mathsf{Pr},\mathsf{Pr}|I}\right) = \log\left(\frac{\Theta_{kk|I,\mathsf{Pr}}}{\Theta_{kk|I}}\right)$$

Store *two* factors (one for numerator, one for denominator)

"Pre-condition" numerator factor on prediction points

$$\Theta_{kk|I,\mathsf{Pr}} = \Theta_{kk|\mathsf{Pr},I}$$

# Algorithm for multiple prediction points

Final objective simplifies to:

$$\operatorname{logdet}\left(\Theta_{\mathsf{Pr},\mathsf{Pr} \mid I \cup \{k\}}\right) - \operatorname{logdet}\left(\Theta_{\mathsf{Pr},\mathsf{Pr} \mid I}\right) = \log\left(\frac{\Theta_{kk \mid I,\mathsf{Pr}}}{\Theta_{kk \mid I}}\right)$$

Store *two* factors (one for numerator, one for denominator)

"Pre-condition" numerator factor on prediction points

$$\Theta_{kk \mid I,\mathsf{Pr}} = \Theta_{kk \mid \mathsf{Pr},I}$$

Complexity of $\mathcal{O}(Nk^2 + Nm^2 + m^3)$ for $m$ prediction points

# Global approximation by KL-minimization

Approximate GP by sparse Cholesky factor of its precision

# Global approximation by KL-minimization

Approximate GP by sparse Cholesky factor of its precision

Measure resulting approximation accuracy by KL divergence:

$$L := \operatorname*{argmin}_{\hat{L} \in S} \mathbb{D}_{\mathsf{KL}} \left( \mathcal{N}(\mathbf{0}, \Theta) \middle\| \mathcal{N}(\mathbf{0}, (\hat{L}\hat{L}^{\top})^{-1}) \right)$$

# Global approximation by KL-minimization

Approximate GP by sparse Cholesky factor of its precision

Measure resulting approximation accuracy by KL divergence:

$$L := \operatorname*{argmin}_{\hat{L} \in S} \mathbb{D}_{\mathsf{KL}} \left( \mathcal{N}(\mathbf{0}, \Theta) \,\middle\|\, \mathcal{N}(\mathbf{0}, (\hat{L}\hat{L}^\top)^{-1}) \right)$$

Using the optimal unique minimizer $L$ from closed form:

$$L_{s_i,i} = \frac{\Theta_{s_i,s_i}^{-1} \boldsymbol{e}_1}{\sqrt{\boldsymbol{e}_1^\top \Theta_{s_i,s_i}^{-1} \boldsymbol{e}_1}}$$

# Global approximation by KL-minimization

Approximate GP by sparse Cholesky factor of its precision

Measure resulting approximation accuracy by KL divergence:

$$L := \operatorname*{argmin}_{\hat{L} \in S} \mathbb{D}_{\mathsf{KL}} \left( \mathcal{N}(\mathbf{0}, \Theta) \,\middle\|\, \mathcal{N}(\mathbf{0}, (\hat{L}\hat{L}^{\top})^{-1}) \right)$$

Using the optimal unique minimizer $L$ from closed form:

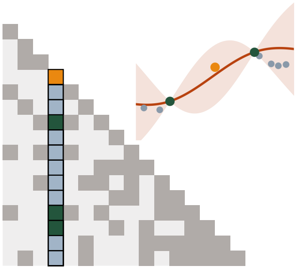
$$L_{s_i,i} = \frac{\Theta_{s_i,s_i}^{-1} \boldsymbol{e}_1}{\sqrt{\boldsymbol{e}_1^{\top} \Theta_{s_i,s_i}^{-1} \boldsymbol{e}_1}}$$

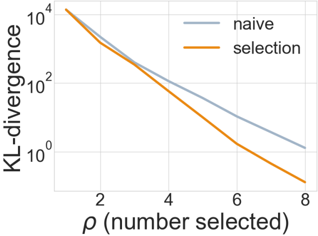
Minimize variance of $i$th point, conditional on selected!

$$2\mathbb{D}_{\mathsf{KL}} = \sum_{i=1}^{N} \left[ \log\left(\Theta_{ii|s_i - \{i\}}\right) - \log\left(\Theta_{ii|i+1:}\right) \right]$$

# Cholesky factorization by selection

Apply column-wise directly

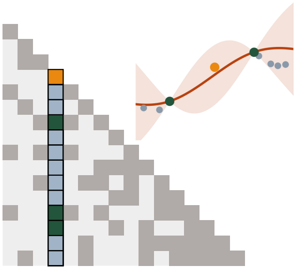


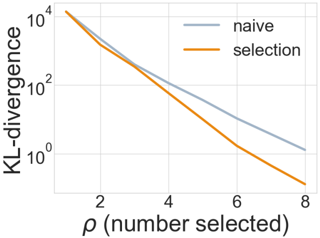


# Cholesky factorization by selection

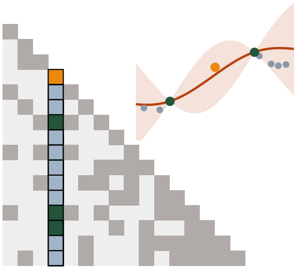
Apply column-wise directly

Improves approx. algorithm of [1]



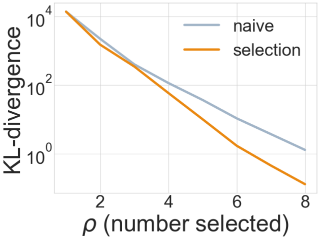


# Cholesky factorization by selection



Apply column-wise directly

Improves approx. algorithm of [1]




[1] F. Schäfer, M. Katzfuss, and H. Owhadi, "Sparse Cholesky factorization by Kullback-Leibler minimization," *arXiv preprint arXiv:2004.14455*, 2020

# GP regression by Cholesky factorization

Lower triangular factor for precision: $LL^\top = \Theta^{-1}$

# GP regression by Cholesky factorization

Lower triangular factor for precision: $LL^\top = \Theta^{-1}$

Upper triangular factor for covariance: $L^{-\top} L^{-1} = \Theta$

# GP regression by Cholesky factorization

Lower triangular factor for precision: $LL^\top = \Theta^{-1}$

Upper triangular factor for covariance: $L^{-\top}L^{-1} = \Theta$

$U = L^{-\top}$, look at submatrices:

$$\begin{aligned}
\Theta = UU^\top &= \begin{pmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{pmatrix} \begin{pmatrix} U_{11}^\top & 0 \\ U_{12}^\top & U_{22}^\top \end{pmatrix} \\
&= \begin{pmatrix} U_{11}U_{11}^\top + U_{12}U_{12}^\top & U_{12}U_{22}^\top \\ U_{22}U_{12}^\top & U_{22}U_{22}^\top \end{pmatrix}
\end{aligned}$$

$$\begin{aligned}
U_{22} &= \mathrm{chol}(\Theta_{22}) \\
U_{12} &= \Theta_{12}U_{22}^{-\top} \\
U_{11} &= \mathrm{chol}(\Theta_{11|2})
\end{aligned}$$

# GP regression by Cholesky factorization

Write conditional terms:

$$\begin{aligned}
\mathrm{E}[\boldsymbol{y}_1 \mid \boldsymbol{y}_2] &= \Theta_{12}\Theta_{22}^{-1}\boldsymbol{y}_2 \\
&= U_{12}U_{22}^{-1}\boldsymbol{y}_2 \\
\mathrm{Cov}[\boldsymbol{y}_1 \mid \boldsymbol{y}_2] &= \Theta_{11} - \Theta_{12}\Theta_{22}^{-1}\Theta_{21} \\
&= U_{11}U_{11}^{\top}
\end{aligned}$$

# GP regression by Cholesky factorization

Write conditional terms:

$$
\begin{aligned}
\mathrm{E}[\boldsymbol{y}_1 \mid \boldsymbol{y}_2] &= \Theta_{12}\Theta_{22}^{-1}\boldsymbol{y}_2 \\
&= U_{12}U_{22}^{-1}\boldsymbol{y}_2 \\
\mathrm{Cov}[\boldsymbol{y}_1 \mid \boldsymbol{y}_2] &= \Theta_{11} - \Theta_{12}\Theta_{22}^{-1}\Theta_{21} \\
&= U_{11}U_{11}^{\top}
\end{aligned}
$$

Recall: $U = L^{-\top}$ so $UL^{\top} = L^{\top}U = I$

$$
\begin{pmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{pmatrix}
\begin{pmatrix} L_{11}^{\top} & L_{21}^{\top} \\ 0 & L_{22}^{\top} \end{pmatrix}
=
\begin{pmatrix} L_{11}^{\top} & L_{21}^{\top} \\ 0 & L_{22}^{\top} \end{pmatrix}
\begin{pmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{pmatrix} = I
$$

$$
\begin{pmatrix} U_{11}L_{11}^{\top} & U_{11}L_{21}^{\top} + U_{12}L_{22}^{\top} \\ 0 & U_{22}L_{22}^{\top} \end{pmatrix}
= \begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix} = I
$$

$$
\begin{pmatrix} L_{11}^{\top}U_{11} & L_{11}^{\top}U_{12} + L_{21}^{\top}U_{22} \\ 0 & L_{22}^{\top}U_{22} \end{pmatrix}
= \begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix} = I
$$

# GP regression by Cholesky factorization

Reading from submatrices,

$$
\begin{aligned}
U_{11} &= L_{11}^{-\top} \\
U_{22} &= L_{22}^{-\top} \\
U_{12} &= -L_{11}^{-\top} L_{21}^{\top} L_{22}^{-\top}
\end{aligned}
$$

# GP regression by Cholesky factorization

Reading from submatrices,

$$
\begin{aligned}
U_{11} &= L_{11}^{-\top} \\
U_{22} &= L_{22}^{-\top} \\
U_{12} &= -L_{11}^{-\top} L_{21}^{\top} L_{22}^{-\top}
\end{aligned}
$$

Re-write conditional terms:

$$
\begin{aligned}
\mathrm{E}[\boldsymbol{y}_1 \mid \boldsymbol{y}_2] &= U_{12} U_{22}^{-1} \boldsymbol{y}_2 \\
&= (-L_{11}^{-\top} L_{21}^{\top} L_{22}^{-\top}) L_{22}^{\top} \boldsymbol{y}_2 \\
&= -L_{11}^{-\top} L_{21}^{\top} \boldsymbol{y}_2 \\
\mathrm{Cov}[\boldsymbol{y}_1 \mid \boldsymbol{y}_2] &= U_{11} U_{11}^{\top} \\
&= L_{11}^{-\top} L_{11}^{-1} \\
\boldsymbol{e}_i^{\top} \mathrm{Cov}[\boldsymbol{y}_1 \mid \boldsymbol{y}_2] \boldsymbol{e}_j &= (L_{11}^{-1} \boldsymbol{e}_i)^{\top} (L_{11}^{-1} \boldsymbol{e}_j)
\end{aligned}
$$

## Summary

Selection algorithm for Gaussian process regression

Drop-in replacement for $k$-th nearest neighbors

Leverage GP regression for sparse Cholesky factorization

Leverage Cholesky factorization for GP regression

Thank you!