Scalable Gaussian-Process Approximations for Big Data

Matthias Katzfuss

Department of Statistics
Texas A&M University
Outline

1 Introduction: Gaussian processes

2 Vecchia approximation

3 Extensions and applications
   - Gaussian noise
   - Generalized GPs
   - Scaled Vecchia for computer-model emulation

4 Conclusions
Outline

1 Introduction: Gaussian processes

2 Vecchia approximation

3 Extensions and applications
   - Gaussian noise
   - Generalized GPs
   - Scaled Vecchia for computer-model emulation

4 Conclusions
Consider a function, observed incompletely, and with noise/error
Consider a function, observed incompletely, and with noise/error.
Consider a function, observed incompletely, and with noise/error.
Gaussian processes (GPs): Probabilistic function estimators

GPs provide an optimal function estimate under the assumption of an infinite-dimensional normal distribution and quantify uncertainty in the form of a joint probability distribution.
GPs provide an optimal function estimate under the assumption of an infinite-dimensional normal distribution and quantify uncertainty in the form of a joint probability distribution.
GPs provide an optimal function estimate under the assumption of an infinite-dimensional normal distribution and quantify uncertainty in the form of a joint probability distribution.
Application areas

Examples:

- Time series
- Geospatial fields (e.g., kriging)
- Emulation of computer experiments
- (Nonlinear) regression and classification
- Machine learning
- Bayesian black-box optimization
GPs: Well suited for big data

- Gap-fill noisy data with UQ
- More data $\rightarrow$ learn more fine-scale features
- Highly flexible

Matthias Katzfuss (Texas A&M)
GPs: Well suited for big data

- Gap-fill noisy data with UQ
- More data $\rightarrow$ learn more fine-scale features
- Highly flexible
GPs: Well suited for big data

- Gap-fill noisy data with UQ
- More data → learn more fine-scale features
- Highly flexible
BUT: GPs are not scalable

For \( n \) data points, need to work with \( n \times n \) covariance matrix:

\[
\Sigma = \begin{pmatrix} K(x_i, x_j) \end{pmatrix}_{i,j=1,\ldots,n}
\]

- \( K \) is a positive-definite kernel or covariance function
- \( x_1, \ldots, x_n \) are input values (e.g., covariate values or spatial locations)

Direct inference has \( \mathcal{O}(n^3) \) time and \( \mathcal{O}(n^2) \) memory complexity

Want methods/approximations that scale linearly in \( n \)
BUT: GPs are not scalable

For \( n \) data points, need to work with \( n \times n \) covariance matrix:

\[ \Sigma = \left( K(x_i, x_j) \right)_{i,j=1,...,n} \]

- \( K \) is a positive-definite kernel or covariance function
- \( x_1, \ldots, x_n \) are input values (e.g., covariate values or spatial locations)

Direct inference has \( \mathcal{O}(n^3) \) time and \( \mathcal{O}(n^2) \) memory complexity

Want methods/approximations that scale linearly in \( n \)
BUT: GPs are not scalable

For $n$ data points, need to work with $n \times n$ covariance matrix:

$$
\Sigma = \left( K(x_i, x_j) \right)_{i,j=1,...,n}
$$

- $K$ is a positive-definite kernel or covariance function
- $x_1, \ldots, x_n$ are input values (e.g., covariate values or spatial locations)

Direct inference has $O(n^3)$ time and $O(n^2)$ memory complexity

Want methods/approximations that scale linearly in $n$
Existing approaches for computational feasibility

Existing approaches include:

- Low-rank $\Sigma$ (e.g., Higdon, 1998; Wikle and Cressie, 1999; Quiñonero-Candela and Rasmussen, 2005; Banerjee et al., 2008; Cressie and Johannesson, 2008)

- Sparse $\Sigma$ (e.g., Furrer et al., 2006; Kaufman et al., 2008)

- Sparse $\Sigma^{-1}$ (e.g., Rue and Held, 2005; Lindgren et al., 2011; Nychka et al., 2015)

- Sparse Cholesky factor of $\Sigma^{-1}$ (Vecchia, 1988; Stein et al., 2004)
Outline

1 Introduction: Gaussian processes

2 Vecchia approximation

3 Extensions and applications
   - Gaussian noise
   - Generalized GPs
   - Scaled Vecchia for computer-model emulation

4 Conclusions
Vecchia approximation

Assume \( y = (y_1, \ldots, y_n) \sim \mathcal{N}(0, \Sigma) \). Density function can be factorized as

\[
p(y) = \prod_{i=1}^{n} p(y_i | y_{h(i)}),
\]

where \( h(i) = \{1, \ldots, i - 1\} \) are the previously ordered indices.

This factorization motivates the Vecchia (1988) approximation:

\[
\hat{p}(y) = \prod_{i=1}^{n} p(y_i | y_{q(i)}),
\]

where \( q(i) \subset h(i) \) is the conditioning set of size \(|q(i)| \leq m\).

Tuning parameter \( m \): Accuracy and computation time both increase with \( m \), but high accuracy with small \( m \) often possible (screening effect).
Vecchia approximation

Assume $\mathbf{y} = (y_1, \ldots, y_n) \sim \mathcal{N}(\mathbf{0}, \Sigma)$. Density function can be factorized as

$$p(\mathbf{y}) = \prod_{i=1}^{n} p(y_i | \mathbf{y}_{h(i)}),$$

where $h(i) = \{1, \ldots, i - 1\}$ are the previously ordered indices.

This factorization motivates the Vecchia (1988) approximation:

$$\hat{p}(\mathbf{y}) = \prod_{i=1}^{n} p(y_i | \mathbf{y}_{q(i)}),$$

where $q(i) \subset h(i)$ is the conditioning set of size $|q(i)| \leq m$.

Tuning parameter $m$: Accuracy and computation time both increase with $m$, but high accuracy with small $m$ often possible (screening effect).
Vecchia approximation

Assume $\mathbf{y} = (y_1, \ldots, y_n) \sim \mathcal{N}(\mathbf{0}, \Sigma)$. Density function can be factorized as

$$p(\mathbf{y}) = \prod_{i=1}^{n} p(y_i | \mathbf{y}_{h(i)}),$$

where $h(i) = \{1, \ldots, i - 1\}$ are the previously ordered indices.

This factorization motivates the Vecchia (1988) approximation:

$$\hat{p}(\mathbf{y}) = \prod_{i=1}^{n} p(y_i | \mathbf{y}_{q(i)}),$$

where $q(i) \subset h(i)$ is the conditioning set of size $|q(i)| \leq m$.

Tuning parameter $m$: Accuracy and computation time both increase with $m$, but high accuracy with small $m$ often possible (screening effect).
• Maximum-minimum-distance (maximin) ordering can be much more accurate than coordinate ordering (Guinness, 2018)

• Conditioning usually on $m$ nearest (previously ordered) neighbors (NN), but more complicated schemes possible
**Sparse inverse Cholesky**

Vecchia approximation: \( \hat{p}(y) = \mathcal{N}_n(y|0, \hat{\Sigma}) \) with \( \hat{\Sigma}^{-1} = UU^\top \), where nonzero entries of \( U \) can be computed easily based on the kernel \( K \)

\( U \) is the optimal sparse triangular matrix under KL divergence (Schäfer, Katzfuss & Owhadi, 2021):

\[
U = \arg \min_{\hat{U} \in S} KL\left( \mathcal{N}(0, \Sigma) \parallel \mathcal{N}(0, (\hat{U}\hat{U}^\top)^{-1}) \right)
\]

for fixed sparsity \( S = \{ A \in \mathbb{R}^{n \times n} : A_{ji} \neq 0 \Rightarrow i = j \text{ or } j \in q(i) \} \)

- \( U \) is sparse with at most \( m \) off-diagonal nonzeros per column
- Closed-form solution can be computed in \( O(nm^3) \) time
- Computations for the \( n \) columns are embarrassingly parallel
Vecchia approximation: \( \hat{p}(y) = \mathcal{N}_n(y | 0, \hat{\Sigma}) \) with \( \hat{\Sigma}^{-1} = UU^\top \), where nonzero entries of \( U \) can be computed easily based on the kernel \( K \)

\( U \) is the optimal sparse triangular matrix under KL divergence (Schäfer, Katzfuss & Owhadi, 2021):

\[
U = \underset{\hat{U} \in S}{\arg \min} KL \left( \mathcal{N}(0, \Sigma) \bigg\| \mathcal{N}(0, (\hat{U}\hat{U}^\top)^{-1}) \right)
\]

for fixed sparsity \( S = \{ A \in \mathbb{R}^{n \times n} : A_{ji} \neq 0 \Rightarrow i = j \text{ or } j \in q(i) \} \)

- \( U \) is sparse with at most \( m \) off-diagonal nonzeros per column
- Closed-form solution can be computed in \( O(nm^3) \) time
- Computations for the \( n \) columns are embarrassingly parallel
Sparse inverse Cholesky

Vecchia approximation: \( \hat{p}(y) = \mathcal{N}_n(y|0, \hat{\Sigma}) \) with \( \hat{\Sigma}^{-1} = \mathbf{U}\mathbf{U}^\top \), where nonzero entries of \( \mathbf{U} \) can be computed easily based on the kernel \( K \).

\( \mathbf{U} \) is the optimal sparse triangular matrix under KL divergence (Schäfer, Katzfuss & Owhadi, 2021):

\[
\mathbf{U} = \arg \min_{\hat{\mathbf{U}} \in S} \text{KL} \left( \mathcal{N}(\mathbf{0}, \Sigma) \parallel \mathcal{N}(\mathbf{0}, (\hat{\mathbf{U}}\hat{\mathbf{U}}^\top)^{-1}) \right)
\]

for fixed sparsity \( S = \{ \mathbf{A} \in \mathbb{R}^{n \times n} : A_{ji} \neq 0 \Rightarrow i = j \text{ or } j \in q(i) \} \)

- \( \mathbf{U} \) is sparse with at most \( m \) off-diagonal nonzeros per column
- Closed-form solution can be computed in \( O(nm^3) \) time
- Computations for the \( n \) columns are embarrassingly parallel
Vecchia illustration in noiseless case

Exact GP  vs.  Vecchia with $m = 4$
Vecchia illustration in noiseless case

Exact GP vs. Vecchia with $m = 4$
Theory

For $n \times n$ Matérn-type covariance matrix under in-fill asymptotics (under maximin ordering and regularity conditions in $d$ dimensions):

- $\epsilon$-accurate approximation can be computed in $O(n \log^{2d}(\frac{n}{\epsilon}))$ time, which is best known complexity (Schäfer, Katzfuss & Owhadi, 2021)
- This implies consistent estimation and prediction for $m = O(\log^d n)$
Many popular existing GP approximations can be viewed as Vecchia approximations:

- Low-rank approaches (e.g., Quiñonero-Candela and Rasmussen, 2005; Banerjee et al., 2008; Finley et al., 2009)
- Full-scale approximation or PIC (e.g., Snelson and Ghahramani, 2007; Sang et al., 2011)
- Multi-resolution approximation (e.g., Katzfuss, 2017; Katzfuss and Gong, 2020)
- Nearest-neighbor GP (e.g., Datta et al., 2016; Finley et al., 2019)
- ...
Vecchia prediction (Katzfuss et al., 2020a)

For prediction of $\mathbf{y}^P = (y_1^P, \ldots, y_{nP}^P)^\top$ at unobserved locations, apply Vecchia to
\[
(y_1, \ldots, y_n, y_1^P, \ldots, y_{nP}^P)^\top
\]

Important: allow $y_i^P$ to condition on previously ordered prediction variables, $\{y_j^P : j < i\}$
Outline

1 Introduction: Gaussian processes

2 Vecchia approximation

3 Extensions and applications
   - Gaussian noise
   - Generalized GPs
   - Scaled Vecchia for computer-model emulation

4 Conclusions
Outline

1 Introduction: Gaussian processes

2 Vecchia approximation

3 Extensions and applications
   - Gaussian noise
   - Generalized GPs
   - Scaled Vecchia for computer-model emulation

4 Conclusions
GP with Gaussian noise: $\Sigma = K + \text{diag}$

Standard Vecchia approximation: applied to data directly (i.e., to $\Sigma$)

Exact GP vs. (standard) Vecchia with $m = 4$

Works well for data without noise
Works very poorly if data are noisy
GP with Gaussian noise: $\Sigma = K + \text{diag}$

Standard Vecchia approximation: applied to data directly (i.e., to $\Sigma$)

Exact GP vs. (standard) Vecchia with $m = 4$

Works well for data without noise

Works very poorly if data are noisy
GP with Gaussian noise: $\Sigma = K + \text{diag}$

Standard Vecchia approximation: applied to data directly (i.e., to $\Sigma$)

Exact GP vs. (standard) Vecchia with $m = 4$

Works well for data without noise

Works very poorly if data are noisy
GP with Gaussian noise: $\Sigma = K + \text{diag}$

Latent Vecchia: applied to the latent GP (i.e., to $K$)

Works well even if data are noisy ($m = 4$)
Computational challenges

- Latent inference requires Cholesky of posterior precision, which can be very dense and expensive (Katzfuss and Guinness, 2021)
- We use incomplete Cholesky (Schäfer, Katzfuss & Owhadi, 2021)
- Comparison for Matérn1.5 at $10^4$ random locations on $[0, 1]^2$
  - Naive: standard Vecchia (cheap)
  - Exact: exact latent Vecchia (expensive)
  - IC, nonzeros($LL^\top$): latent + incomplete Cholesky (cheap)
Outline

1 Introduction: Gaussian processes

2 Vecchia approximation

3 Extensions and applications
   - Gaussian noise
   - Generalized GPs
   - Scaled Vecchia for computer-model emulation

4 Conclusions
Non-Gaussian spatial data: Generalized GP

Conditional on GP, data are non-Gaussian (from exponential family): binary, categorical, counts, right-skewed, . . .

Example: Binary classification using logistic GGP:

- Take GP function
- Transform into probability using logistic link, then draw from Bernoulli distribution
Non-Gaussian spatial data: Generalized GP

Conditional on GP, data are non-Gaussian (from exponential family): binary, categorical, counts, right-skewed, . . .

Example: Binary classification using logistic GGP:

- Take GP function
- Transform into probability using logistic link, then draw from Bernoulli distribution
Non-Gaussian spatial data: Generalized GP

Conditional on GP, data are non-Gaussian (from exponential family): binary, categorical, counts, right-skewed, …

Example: Binary classification using logistic GGP:
- Take GP function
- Transform into probability using logistic link, then draw from Bernoulli distribution
Laplace for non-Gaussian data

For generalized GP, posterior is intractable $\rightarrow$ 2nd-order Taylor expansion of log-posterior at the mode (Laplace approximation).

Newton-Raphson: Iterative GP prediction using Gaussian pseudo-data

But still $O(n^3) \rightarrow$ infeasible for large $n$
Laplace for non-Gaussian data

For generalized GP, posterior is intractable → 2nd-order Taylor expansion of log-posterior at the mode (Laplace approximation).

Newton-Raphson: Iterative GP prediction using Gaussian pseudo-data

But still $O(n^3)$ → infeasible for large $n$
Laplace for non-Gaussian data

For generalized GP, posterior is intractable → 2nd-order Taylor expansion of log-posterior at the mode (Laplace approximation).

Newton-Raphson: Iterative GP prediction using Gaussian pseudo-data

But still $O(n^3) \rightarrow$ infeasible for large $n$
Vecchia-Laplace (Zilber & Katzfuss, 2021)

Given pseudo-data with Gaussian noise, can approximate using Vecchia

Comparison of MSE relative to Laplace:

Can also be used for analysis of point patterns (log-Gaussian Cox process)
Vecchia-Laplace (Zilber & Katzfuss, 2021)

Given pseudo-data with Gaussian noise, can approximate using Vecchia

Comparison of MSE relative to Laplace:

\( \nu = 0.5 \) (Logistic)

\( \nu = 0.5 \) (Poisson)

\( \nu = 0.5 \) (Gamma)

Can also be used for analysis of point patterns (log-Gaussian Cox process)
Vecchia-Laplace (Zilber & Katzfuss, 2021)

Given pseudo-data with Gaussian noise, can approximate using Vecchia

Comparison of MSE relative to Laplace:

\( \nu = 0.5 \) (Logistic)  
\( \nu = 0.5 \) (Poisson)  
\( \nu = 0.5 \) (Gamma)

Can also be used for analysis of point patterns (log-Gaussian Cox process)
Outline

1 Introduction: Gaussian processes

2 Vecchia approximation

3 Extensions and applications
   - Gaussian noise
   - Generalized GPs
   - Scaled Vecchia for computer-model emulation

4 Conclusions
Scaled Vecchia (Katzfuss et al., 2020b)

ARD kernel: different relevance for each input dimension
⇒ Carry out maximin ordering and NN conditioning in scaled space
Comparison for Matérn GP in 10 input dimensions

Known parameters

Estimated parameters
Comparison for satellite-drag computer model

Data and (H-)laGP results from Sun et al. (2019)

![Graph showing comparison between SVecchia, Vecchia, laGP, and H-laGP]

8 input dimensions, \( n = 2 \) million runs, 6 chemical species
SVecchia took 13–14min (2 orders of magnitude faster than H-laGP)
Outline

1. Introduction: Gaussian processes
2. Vecchia approximation
3. Extensions and applications
   - Gaussian noise
   - Generalized GPs
   - Scaled Vecchia for computer-model emulation
4. Conclusions
Conclusions

• Vecchia framework for GP approximations:
  • Highly accurate
  • Can lead to almost universal GP toolbox
  • Can guarantee linear scalability, plus parallel computations and mini-batching

• R packages GPvecchia (K et al) and GpGp (Guinness & K) on CRAN

• Supported by NSF DMS–1654083, DMS–1953005, CCF–1934904, TAMUS NLO, and TAMIDS.
Main references


