Fast Kronecker Inference in Gaussian Processes with non-Gaussian Likelihoods

Seth Flaxman, Andrew Gordon Wilson, Daniel B. Neill, Hannes Nickisch and Alexander J. Smola

Discussed by: Yizhe Zhang

March 11, 2016
Outline

1. Introduction to Gaussian processes (GPs)

2. Methods

3. Experiments
Gaussian processes

- Given a dataset \( D = (y, X) \), we assume the relationship between the predictors and targets is determined by a latent Gaussian process:

\[
f(x) \sim \mathcal{GP}(m, k_\theta), y \sim p(y(x)|f(x)) \tag{1}
\]

- We are interested in:
  - **Predictive distribution** For any test input \( x_* \), predict \( f_* \)

\[
p(f_*|D, x_*, \theta) = \int p(f_*|X, x_*, f, \theta)p(f|D, \theta)df. \tag{2}
\]

  - **Marginal likelihood** Optimize this likelihood to learn \( \theta \)

\[
p(y|\theta, X) = \int p(y|f)p(f|X, \theta)df. \tag{3}
\]
Gaussian processes with non-Gaussian likelihoods

- Unfortunately, for all but the Gaussian likelihood, (2) and (3) are intractable.
- Laplace approximation uses a second order Taylor expansion to approximate the unnormalized log posterior $\log p(f|D)$:

$$
\Psi(f) \triangleq \log p(f|D) = \log p(y|f) + \log p(f|X) + \text{const},
$$

$$
\nabla \Psi(f) = \nabla \log p(y|f) - K^{-1}(f - \mu),
$$

$$
\nabla\nabla \Psi(f) = \nabla\nabla \log p(y|f) - K^{-1},
$$

where $\mu = m(x_i)$ is the mean function.
Gaussian processes with non-Gaussian likelihoods (Cont’d)

• The $\hat{f}$ that maximize the $\Psi(f)$ can be found by applying Newton’s method:

$$f^{new} \leftarrow f^{old} - (\nabla\nabla\Psi)^{-1}\nabla\Psi. \quad (7)$$

• The approximated distribution is given by

$$q(f|D) \sim \mathcal{N}(\hat{f}, (K^{-1} + W)^{-1}). \quad (8)$$

$W \triangleq \nabla\nabla \log p(y|f)$ is a diagonal matrix since the likelihood $p(y|f)$ factorizes as $\prod_i p(y_i|f_i)$. 
Gaussian processes with non-Gaussian likelihoods (Cont’d)

- **Predictive distribution** Defining $A = W^{-1} + K$, one can obtain the predictive distribution as

$$p(f_*|D, x_*, \theta) \simeq \mathcal{N}(k_*^T \nabla \log p(y|\hat{f}), k_{**} - k_*^T A^{-1} k_*), \quad (9)$$

where $k_* = [k(x_*, x_1), \cdots, k(x_*, x_n)]^T$, and $k_{**} = k(x_*, x_*)$.

- **Marginal likelihood**

$$\log p(y|X, \theta) = \log \int \exp[\Psi(f)] df \quad (10)$$

$$\simeq \log p(y|\hat{f}) - \frac{1}{2} \alpha^T (\hat{f} - \mu) - \frac{1}{2} \log |I + KW|, \quad (11)$$

where $\alpha \triangleq K^{-1}(\hat{f} - \mu)$

---

$^1$they originally write as $\alpha^T K^{-1} \alpha$
Numerical conditioning

• For numerical stability, they use the transformation in Rasmussen & Williams (2006)

\[
B = I + W^{1/2}KW^{1/2}, \quad Q = W^{1/2}B^{-1}W^{1/2},
\]
\[
b = W(f - \mu) + \nabla \log p(y|f), \quad a = b - QKb. \tag{13}
\]

• \( B \) has eigenvalues bounded below by 1 and bounded above by \( 1 + n\max_{i,j}(K_{ij})/4 \), so for many covariance functions, \( B \) is guaranteed to be well-conditioned [Rasmussen & Williams (2006)].

• The Newton update in (7) becomes \(^2\):

\[
f^{\text{new}} \leftarrow Ka. \tag{14}
\]

• The predictive distribution in (9) becomes:

\[
p(f_*|\mathcal{D}, x_*, \theta) \approx \mathcal{N}(k_*^T\nabla \log p(y|\hat{f}), k_{**} - k_*^TQk_*) \tag{15}
\]

\(^2\)following Woodbury matrix identity
Outline

1. Introduction to Gaussian processes (GPs)

2. Methods

3. Experiments
Kronecker Methods

- Inference and learning require solving linear systems $M^{-1}v$ and computing log-determinants $\log |M|$, where $M$ is a $n \times n$ matrix. This typically requires $O(n^3)$ time and $O(n^2)$ space.
- Using Kronecker methods, these operations only require $O(Dn^{\frac{D+1}{D}})$ time and $O(Dn^{\frac{2}{D}})$ storage, for $D$ input dimensions.
- The Newton update step in (14) requires costly matrix-vector multiplications and inversions of $B$. They replace (14) with the following two steps \(^3\):
  \[
  Bz = W^{-1/2}b, \tag{16}
  \]
  \[
  \alpha^{\text{new}} = W^{1/2}z. \tag{17}
  \]
  \(^3\)The Newton updates can be applied to $\alpha$ rather than $f$ (Rasmussen & Williams [2006]).
Fiedler’s bound

- Evaluating the marginal likelihood in (11) requires computing $\log |I + KW|$.
- Fiedler (1971) showed that for Hermitian positive semidefinite matrices $U$ and $V$:
  \[ |U + V| \leq \prod_{i} (u_i + v_{n-i+1}), \tag{18} \]
  where $u_1 \leq \cdots \leq u_n$ and $v_1 \leq \cdots \leq v_n$ are eigenvalues of $U$ and $V$.
- Applying this bound, they obtain,
  \[
  \log |I + KW| = \log(|K + W^{-1}||W|) \\
  \leq \log \prod_{i} (e_i + w_i^{-1}) \prod_{i} w_i = \sum_{i} \log(1 + e_i w_i) \tag{20}
  \]
- Using non-linear conjugate gradients to find the best $\hat{\theta}$ to maximize the approximate marginal likelihood
Log-marginal likelihood approximation

- To compare the accuracy and speed of this approximation, they generated synthetic data on an $\sqrt{n} \times \sqrt{n}$ grid.

![Graphs showing approximation ratios and runtime comparisons](image)

**Figure:** a), the approximation ratio of their approximation to the marginal likelihood. b) and d), the approximation ratios for the log-determinant divided by the exact log-determinant. In c) the runtime comparison.
Algorithm 1 Kronecker GP Inference and Learning

1: **Input:** $\theta, \mu, K, p(y|f), y$
2: Construct $K_1, \ldots, K_D$
3: $\alpha \leftarrow 0$
4: **repeat**
5: \hphantom{5:} $f \leftarrow K\alpha + \mu$ \hfill # Eq. (A22)
6: \hphantom{5:} $W \leftarrow -\nabla^2 \log p(y|f)$ \hfill # Diagonal
7: \hphantom{5:} $b \leftarrow W(f - \mu) + \nabla p(y|f)$
8: \hphantom{5:} Solve $Bz = W^{-\frac{1}{2}}b$ with CG \hfill # Eq. (16)
9: \hphantom{5:} $\Delta\alpha \leftarrow W^{\frac{1}{2}}z - \alpha$ \hfill # Eq. (17)
10: $\hat{\xi} \leftarrow \arg\min_{\xi} \Psi(\alpha + \xi \Delta\alpha)$ \hfill # Line Search
11: $\alpha \leftarrow \alpha + \hat{\xi} \Delta\alpha$ \hfill # Update
12: **until** convergence of $\Psi$
13: $e = \text{eig}(K)$ \hfill # exploit Kronecker structure
14: $Z \leftarrow \alpha^\top(f - \mu)/2 + \sum_i \log(1 + e_i W_i)/2 - \log p(y|f)$
15: **Output:** $f, \alpha, Z$
Analysis for complexity

- **line 2** Construct kernel: $O(Dn^{2/D})$
- **line 5** Matrix-vector multiplication: $O(Dn^{D+1/D})$
- **line 8** Linear conjugate descent: $O(Dn^{D+1/D})$
- **line 13** Eigendecomposition: $O(mDn^{3/D})$
- **Overall time complexity**: $O(Dn^{D+1/D})$ vs standard GP with Cholesky approach $O(n^3)$
- **Storage complexity**: $O(Dn^{2/D})$ vs standard GP $O(n^2)$
Outline

1 Introduction to Gaussian processes (GPs)
2 Methods
3 Experiments
Log-Gaussian Cox Process (LGCP)

- An LGCP is a Cox process (inhomogeneous Poisson process with stochastic intensity) driven by a latent log intensity function with a GP prior:

\[
\begin{align*}
  f(s) &\sim \mathcal{GP}(\mu(s), k_\theta(\cdot, \cdot)), \\
  \lambda(s) &\equiv \exp[f(s)], \\
  y_S|\lambda(s) &\sim \text{Poisson}(\int_{s \in S} \lambda(s) ds).
\end{align*}
\]  

\[ (21) \]

\[ (22) \]

\[ (23) \]

- After binning the observation with a fixed width window, this becomes

\[
y_i|f(s_i) \sim \text{Poisson}(\exp(f(s_i)))
\]

\[ (24) \]

- They also considered negative binomial likelihood to deal with overdispersion.
Model specification

- They employ two types of covariance functions: Matern kernels and Spectral Mixture (SM) kernels.
- Spectral Mixture kernels is a scale-location mixture of Gaussians, which can approximate any stationary covariance function.

\[
k(z, z') = \sum_{q=1}^{Q} w_q \exp[-2\pi^2 \tau^2 (z - z')^2 v_q] \cos(2\pi \tau \mu_q). \tag{25}
\]

- For space-time data with coordinates \((x, y, t)\), they specify the covariance function \(k_\theta\) as:

\[
k_\theta((x, y, t), (x', y', t')) = k_x(x, x') k_y(y, y') k_t(t, t'), \tag{26}
\]

where \(k_x\) and \(k_y\) are Matern-5/2 kernels and \(k_z\) is a SM kernel.
Synthetic data

- **Data:** A grid of $n \times n \times n$ with 3 SM kernels. 
  \[ y_i \sim \text{NegativeBinomial}(\exp(f(s_i)) + 1) \]

- **Comparing metric:** Predictive log-likelihood. (the hyperparameters are learned)

- Compared with FITC approximation (an inducing point method) and low-rank approximation [Groot et al. (2014)].

**Figure:** Run-time and accuracy (mean squared error) of optimizing the hyperparameters of a GP with the Laplace approximation.
Real-world dataset

- **Crime rate forecasting in chicago** The dataset consists of 233,088 reported incidents of assault on a $288 \times 446$ grid over 10 years.
- For learning the hyperparameters, they ran 200 iterations with Gaussian likelihood, then another 200 iterations for negative binomial likelihood.

*Figure: Local area posterior forecasts.*
Real-world dataset (Cont’d)

- **Kron NB SM-20** (their method) uses Kronecker inference with a negative binomial observation model and an SM kernel with 20 components ($Q = 20$).
- **KronNB SM-20 Low Rank** uses a rank 5 approximation.
- **KronGauss SM-20** uses a Gaussian observation model.
- **FITC 100** uses the same observation model and kernel as KronNB SM-20 with 100 inducing points and FITC inference.
- **SSGPR-200** uses a Gaussian observation model and 200 spectral points.

<table>
<thead>
<tr>
<th></th>
<th>KronNB SM-20</th>
<th>KronNB SM-20 Low Rank</th>
<th>KronGauss SM-20</th>
<th>FITC-100 NB SM-20</th>
<th>SSGPR-200</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Training RMSE</strong></td>
<td>0.79</td>
<td>1.13</td>
<td>$10^{-11}$</td>
<td>2.14</td>
<td>1.45</td>
</tr>
<tr>
<td><strong>Forecast RMSE</strong></td>
<td>1.26</td>
<td>1.24</td>
<td>1.28</td>
<td>1.77</td>
<td>1.26</td>
</tr>
<tr>
<td><strong>Forecast log-likelihood</strong></td>
<td>-33,916</td>
<td>-172,879</td>
<td>-352,320</td>
<td>-42,897</td>
<td>-82,781</td>
</tr>
<tr>
<td><strong>Runtime</strong></td>
<td>2.8 hours</td>
<td>9 hours</td>
<td>22 min.</td>
<td>4.5 hours</td>
<td>2.8 hours</td>
</tr>
</tbody>
</table>
Highlights

- They use Kronecker methods for non-Gaussian likelihoods. Specifically, they use a Laplace approximation, which naturally harmonizes with Kronecker methods. EP and VB are not scalable with large $n$.
- They use a bound to estimate the log determinant term.
- They develop a spatiotemporal log Gaussian Cox process (LGCP), with highly expressive spectral mixture covariance kernels.
- Experiments on public policy problem shows good results.