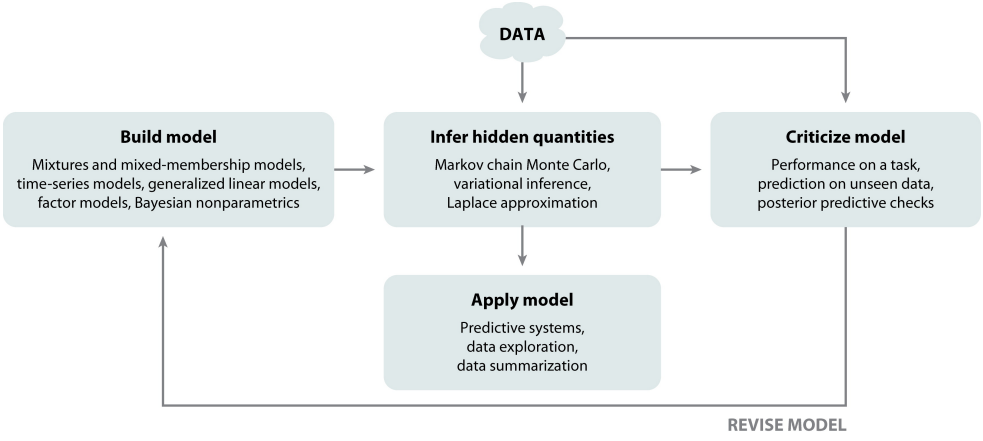
STATS271/371: Applied Bayesian Statistics

Gaussian processes, elliptical slice sampling, and Bayesian optimization

Scott Linderman

May 19, 2021

Box's Loop



Blei DM. 2014.
Annu. Rev. Stat. Appl. 1:203–32

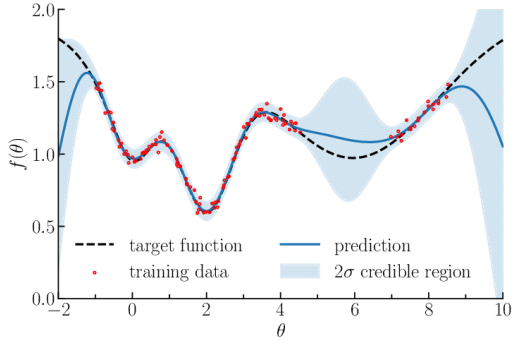
Blei, *Ann. Rev. Stat. App.* 2014.

Lap 8: Gaussian processes, elliptical slice sampling, and Bayesian optimization

- **Model:** Gaussian processes
- **Algorithm:** Elliptical slice sampling
- **Application:** Bayesian optimization

Gaussian processes

- Gaussian processes are distributions on functions $f : \mathbb{R}^D \to \mathbb{R}$. (We can generalize to other domains as well.)
- Equivalently, a GP is a continuous set of r.v.'s $\{f(\boldsymbol{x}) : \boldsymbol{x} \in \mathbb{R}^D\}$; i.e. a stochastic process.



https://www.researchgate.net/figure/Illustration-of-Gaussian-process-regressio fig1_327613136

Gaussian processes II

We say $f \sim \mathrm{GP}(\mu(\cdot), K(\cdot,\cdot))$ if

$$\begin{bmatrix} f(\boldsymbol{x}_1) \\ \vdots \\ f(\boldsymbol{x}_N) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu(\boldsymbol{x}_1) \\ \vdots \\ \mu(\boldsymbol{x}_N) \end{bmatrix}, \begin{bmatrix} K(\boldsymbol{x}_1,\boldsymbol{x}_1) & \cdots & K(\boldsymbol{x}_1,\boldsymbol{x}_N) \\ \vdots & & \vdots \\ K(\boldsymbol{x}_N,\boldsymbol{x}_1) & \cdots & K(\boldsymbol{x}_N,\boldsymbol{x}_N) \end{bmatrix} \right) \tag{1}$$

for all finite subsets of points $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\} \subset \mathbb{R}^D$.

Here, $\mu : \mathbb{R}^D \to \mathbb{R}$ is the **mean function** and $K : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ is the **covariance function**, or **kernel**.

The covariance matrix obtained by applying the covariance function to each pair of data points above is called the **Gram matrix**.

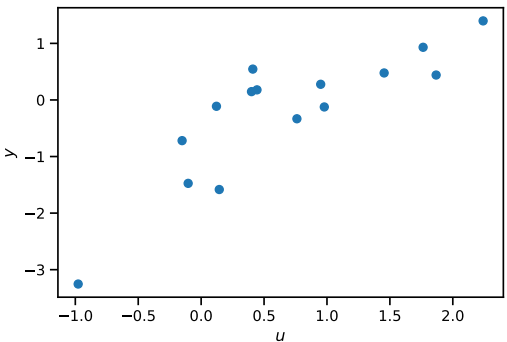
The covariance function must be positive definite; i.e. the Gram matrix must be positive definite for any subset of points.

From linear regression to GPs

- Think back to Lap 1: Bayesian Linear Regression.
- Our motivating example was approximating a $D = 1$ dimensional function $y(x) : \mathbb{R} \to \mathbb{R}$ given noisy observations $\{y_n, x_n\}_{n=1}^N$.
- We cast polynomial regression as linear regression by encoding the inputs x_n with feature vectors,

$$\boldsymbol{\phi}(x_n) = (x_n^0, x_n^1, \ldots, x_n^{P-1}) \in \mathbb{R}^P. \qquad (2)$$

- (I've changed notation slightly to match the slides above.)



From linear regression to GPs II

More generally, let $\boldsymbol{\phi}(x) = (\phi_1(x), \ldots, \phi_P(x)) \in \mathbb{R}^P$ be a function that encodes x in a P-dimensional feature space.

With these features, our linear model was,

$$\mathbb{E}[y_n \mid x_n] = \sum_{p=1}^{P} w_p \phi_p(x_n) = \boldsymbol{\phi}(x_n)^\top \boldsymbol{w} \triangleq f(x_n). \tag{3}$$

Now assume a Gaussian prior, $\boldsymbol{w} \sim \mathcal{N}(\mathbf{0}, \frac{\lambda}{P}\boldsymbol{I})$. Then for any $\{x_1, \ldots, x_N\} \subset \mathbb{R}$,

$$\begin{bmatrix} f(x_1) \\ \vdots \\ f(x_N) \end{bmatrix} = \begin{bmatrix} \phi_1(x_1) & \cdots & \phi_P(x_1) \\ \vdots & & \vdots \\ \phi_1(x_N) & \cdots & \phi_P(x_N) \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_P \end{bmatrix} = \boldsymbol{\Phi}\boldsymbol{w} \sim \mathcal{N}(\mathbf{0}, \tfrac{\lambda}{P}\boldsymbol{\Phi}\boldsymbol{\Phi}^\top) \tag{4}$$

The function $f(\cdot)$ is a GP! It has kernel $K(x_i, x_j) = \frac{\lambda}{P}\boldsymbol{\phi}(x_i)^\top \boldsymbol{\phi}(x_j)$

Example: radial basis functions

Instead of polynomial features, let's use **radial basis functions**,

$$\phi_p(x) = e^{-\frac{1}{2\ell^2}(x-c_p)^2}. \tag{5}$$

These are un-normalized Gaussian bumps of width ℓ located at $c_1, \ldots, c_P$.

With this feature encoding, the kernel is,

$$K(x_i, x_j) = \tfrac{\lambda}{P}\boldsymbol{\phi}(x_i)^\top \boldsymbol{\phi}(x_j) = \frac{\lambda}{P}\sum_{p=1}^{P} e^{-\frac{1}{2\ell^2}(x_i-c_p)^2} e^{-\frac{1}{2\ell^2}(x_j-c_p)^2} \tag{6}$$

Now take the limit of having $P \to \infty$ equally spaced centers. Then,

$$\lim_{P\to\infty} K(x_i, x_j) = \lambda \int_{-\infty}^{\infty} e^{-\frac{1}{2\ell^2}(x_i-c)^2} e^{-\frac{1}{2\ell^2}(x_j-c)^2}\, \mathrm{d}c = \sqrt{\pi}\ell\lambda e^{-\frac{1}{2(\sqrt{2}\ell)^2}(x_i-x_j)^2} \tag{7}$$

This is called a **squared exponential kernel** with length scale $\sqrt{2}\ell$ and variance $\sqrt{\pi}\ell\lambda$.

Demo: sampling GPs with squared exponential kernels

https://colab.research.google.com/drive/1PFF5EkD-6g4Ta_4oU3y5ErdzQKh0T1lX?usp=sharing

The Matérn family of kernels

The Matérn family of kernels is defined by

$$K(x_i, x_j) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}\Delta_{ij}}{\ell} \right)^{\nu} K_{\nu} \left(\frac{\sqrt{2\nu}\Delta_{ij}}{\ell} \right), \tag{8}$$

where

- $\Delta_{ij} = x_i - x_j$ is the difference of the points,
- ℓ is a positive length scale,
- the positive parameter ν controls the smoothness of the function, and
- and K_{ν} (in a potentially confusing overloading of notation) denotes the modified Bessel function.

When $\nu \to \infty$, the Matérn kernel converges to the squared exponential.

When $\nu = \frac{1}{2}$ the kernel is $K(x_i, x_j) = e^{-\frac{\Delta_{ij}}{\ell}}$, the covariance function of the **Ornstein-Uhlenbeck (OU) process**, a continuous-time AR(1) process.

When $\nu = p - \frac{1}{2}$, the kernel is the covariance of a continuous-time AR(p) process.

The Matérn family of kernels II

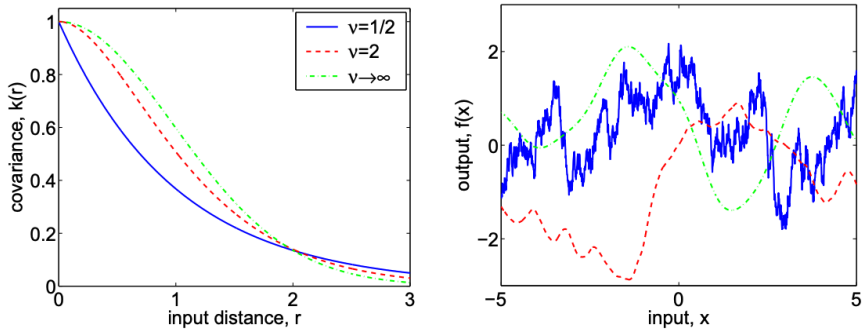


Figure 4.1: Panel (a): covariance functions, and (b): random functions drawn from Gaussian processes with Matérn covariance functions, eq. (4.14), for different values of ν, with $\ell = 1$. The sample functions on the right were obtained using a discretization of the x-axis of 2000 equally-spaced points.

Stationary kernels and Bochner's theorem

Note that both the squared exponential and the Matérn kernels are **stationary** in that $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ only depends on $\boldsymbol{\Delta}_{ij} = \boldsymbol{x}_i - \boldsymbol{x}_j$.

Stationary kernels are particularly interesting because they can be defined by their power spectrum.

Theorem (Bochner's Theorem)

A complex-valued function $K(\boldsymbol{\Delta})$ on $\mathbb{R}^D$ is the covariance function of a weakly stationary mean square continuous complex valued random process on $\mathbb{R}^D$ if and only if it can be represented as

$$K(\boldsymbol{\Delta}) = \int_{\mathbb{R}^D} e^{2\pi i \boldsymbol{s}\cdot\boldsymbol{\Delta}} \, \mathrm{d}\mu(\boldsymbol{s}) \tag{9}$$

where μ is a positive finite measure.

If μ has a density $S(\boldsymbol{s})$ then S is known as the **spectral density** or **power spectrum** corresponding to $K(\boldsymbol{\Delta})$. (This is Theorem 4.1 of Williams and Rasmussen [1996].)

Power spectrum intuition

Think of $\mu(\boldsymbol{s})$ as the power put into frequency $\boldsymbol{s}$. The constraint that it be positive is akin to requiring that the covariance be positive definite.

The **Wiener-Khintchine theorem** says that covariance function and the spectral density (assuming it exists) are Fourier duals of one another,

$$K(\boldsymbol{\Delta}) = \int S(\boldsymbol{s}) e^{2\pi i \boldsymbol{s}\cdot\boldsymbol{\Delta}} \,\mathrm{d}\boldsymbol{s}, \tag{10}$$

$$S(\boldsymbol{s}) = \int K(\boldsymbol{\Delta}) e^{-2\pi i \boldsymbol{s}\cdot\boldsymbol{\Delta}} \,\mathrm{d}\boldsymbol{\Delta} \tag{11}$$

The variance of the process is $K(\mathbf{0}) = \int S(\boldsymbol{s}) \,\mathrm{d}\boldsymbol{s}$ so the spectral density must be integrable (it must decay sufficiently fast as $|\boldsymbol{s}| \to \infty$) to define a valid process.

One nice consequence: the Gram matrix of a stationary kernel evaluated at evenly spaced points $x_1, \ldots, x_N$ in 1D is a **Toeplitz matrix**, which can be inverted in only $O(N \log N)$ time using the Fourier transform [Storkey, 1999, Cunningham et al., 2008]

Common kernels

covariance function	expression	S	ND
constant	σ_0^2	$\surd$	
linear	$\sum_{d=1}^{D} \sigma_d^2 x_d x'_d$		
polynomial	$(\mathbf{x} \cdot \mathbf{x}' + \sigma_0^2)^p$		
squared exponential	$\exp(-\frac{r^2}{2\ell^2})$	$\surd$	$\surd$
Matérn	$\frac{1}{2^{\nu-1}\Gamma(\nu)} \left(\frac{\sqrt{2\nu}}{\ell} r\right)^{\nu} K_\nu \left(\frac{\sqrt{2\nu}}{\ell} r\right)$	$\surd$	$\surd$
exponential	$\exp(-\frac{r}{\ell})$	$\surd$	$\surd$
γ-exponential	$\exp\left(-(\frac{r}{\ell})^\gamma\right)$	$\surd$	$\surd$
rational quadratic	$(1 + \frac{r^2}{2\alpha\ell^2})^{-\alpha}$	$\surd$	$\surd$
neural network	$\sin^{-1}\left(\frac{2\tilde{\mathbf{x}}^\top \Sigma \tilde{\mathbf{x}}'}{\sqrt{(1+2\tilde{\mathbf{x}}^\top \Sigma \tilde{\mathbf{x}})(1+2\tilde{\mathbf{x}}'^\top \Sigma \tilde{\mathbf{x}}')}}\right)$		$\surd$

Table 4.1: Summary of several commonly-used covariance functions. The covariances are written either as a function of $\mathbf{x}$ and $\mathbf{x}'$, or as a function of $r = |\mathbf{x} - \mathbf{x}'|$. Two columns marked 'S' and 'ND' indicate whether the covariance functions are stationary and nondegenerate respectively. Degenerate covariance functions have finite rank, see section 4.3 for more discussion of this issue.

From Williams and Rasmussen [1996]

The Kernel Cookbook:

Advice on Covariance functions

by David Duvenaud

https://www.cs.toronto.edu/~duvenaud/cookbook/

The "Automated Statistician"

Duvenaud et al. [2013] proposed a method to search over compositions of kernels to best fit the data. The idea is very cool, but unfortunately they came up with the *worst name ever*.

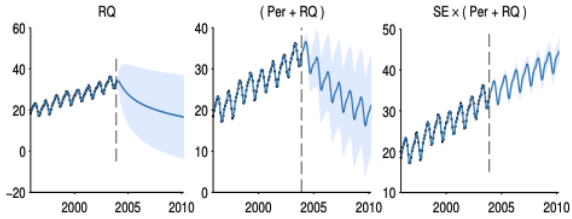


Figure 3. Posterior mean and variance for different depths of kernel search. The dashed line marks the extent of the dataset. In the first column, the function is only modeled as a locally smooth function, and the extrapolation is poor. Next, a periodic component is added, and the extrapolation improves. At depth 3, the kernel can capture most of the relevant structure, and is able to extrapolate reasonably.

Gaussian process regression

Now that we have a prior distribution on functions, suppose we observe $(y_n, \boldsymbol{x}_n)$ pairs. Assume,

$$f \sim \mathrm{GP}(\mu(\cdot), K(\cdot,\cdot)) \qquad\qquad y_n \sim \mathcal{N}(f(\boldsymbol{x}_n), \sigma^2) \tag{12}$$

independently for $n = 1, \ldots, N$.

Let $\boldsymbol{y} = [y_1, \ldots, y_N]^\top$, $\boldsymbol{\mu} = [\mu(\boldsymbol{x}_1), \ldots, \mu(\boldsymbol{x}_N)]^\top$, $\boldsymbol{f} = [f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_N)]^\top$, and $\boldsymbol{K}$ denote the Gram matrix of $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$. Under the GP prior,

$$p(\boldsymbol{f} \mid \{\boldsymbol{x}_n, y_n\}_{n=1}^N) \propto \left[\prod_{n=1}^N p(y_n \mid f(\boldsymbol{x}_n))\right] p(\boldsymbol{f})\,\mathrm{d}\boldsymbol{f} \tag{13}$$

$$= \mathcal{N}(\boldsymbol{y} \mid \boldsymbol{f}, \sigma^2 \boldsymbol{I})\, \mathcal{N}(\boldsymbol{f} \mid \boldsymbol{\mu}, \boldsymbol{K})\,\mathrm{d}\boldsymbol{f} \tag{14}$$

$$\propto \mathcal{N}(\boldsymbol{f} \mid \boldsymbol{\mu}', \boldsymbol{K}'), \tag{15}$$

where

$$\boldsymbol{K}' = (\boldsymbol{K}^{-1} + \sigma^{-2}\boldsymbol{I})^{-1} \qquad\qquad \boldsymbol{\mu}' = \boldsymbol{K}'(\boldsymbol{K}^{-1}\boldsymbol{\mu} + \sigma^{-2}\boldsymbol{y}) \tag{16}$$

Marginal distribution

Likewise, we can integrate over the random function to obtain the marginal distribution,

$$p(\boldsymbol{y}) = \int p(\boldsymbol{y} \mid \boldsymbol{f})\, p(\boldsymbol{f})\, \mathrm{d}\boldsymbol{f} \tag{17}$$

$$= \int \mathcal{N}(\boldsymbol{y} \mid \boldsymbol{f}, \sigma^2 \boldsymbol{I})\, \mathcal{N}(\boldsymbol{f} \mid \boldsymbol{\mu}, \boldsymbol{K})\, \mathrm{d}\boldsymbol{f} \tag{18}$$

$$= \mathcal{N}(\boldsymbol{y} \mid \boldsymbol{\mu}, \boldsymbol{K} + \sigma^2 \boldsymbol{I}). \tag{19}$$

This allows us to compute the marginal likelihood of the data exactly.

Posterior predictive distribution

What is the posterior predictive distribution $p(y_{N+1} \mid \boldsymbol{x}_{N+1}, \{\boldsymbol{x}_n, y_n\}_{n=1}^N)$?

It's one big Gaussian model,

$$\begin{bmatrix} y_1 \\ \vdots \\ y_N \\ y_{N+1} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu(\boldsymbol{x}_1) \\ \vdots \\ \mu(\boldsymbol{x}_N) \\ \mu(\boldsymbol{x}_{N+1}) \end{bmatrix}, \begin{bmatrix} K(\boldsymbol{x}_1, \boldsymbol{x}_1) & \cdots & K(\boldsymbol{x}_1, \boldsymbol{x}_N) & K(\boldsymbol{x}_1, \boldsymbol{x}_{N+1}) \\ \vdots & & \vdots & \\ K(\boldsymbol{x}_N, \boldsymbol{x}_1) & \cdots & K(\boldsymbol{x}_N, \boldsymbol{x}_N) & K(\boldsymbol{x}_N, \boldsymbol{x}_{N+1}) \\ K(\boldsymbol{x}_{N+1}, \boldsymbol{x}_1) & \cdots & K(\boldsymbol{x}_{N+1}, \boldsymbol{x}_N) & K(\boldsymbol{x}_{N+1}, \boldsymbol{x}_{N+1}) \end{bmatrix} + \sigma^2 \boldsymbol{I} \right) \tag{20}$$

We obtain the predictive likelihood via a **Schur complement**,

$$y_{N+1} \mid \boldsymbol{x}_{N+1}, \{\boldsymbol{x}_n, y_n\}_{n=1}^N \sim \mathcal{N}(m_{N+1}, v_{N+1}) \tag{21}$$

$$m_{N+1} = \mu(\boldsymbol{x}_{N+1}) + \boldsymbol{k}^\top (\boldsymbol{K} + \sigma^2 \boldsymbol{I})^{-1} (\boldsymbol{y} - \boldsymbol{\mu}) \tag{22}$$

$$v_{N+1} = K(\boldsymbol{x}_{N+1}, \boldsymbol{x}_{N+1}) - \boldsymbol{k}^\top (\boldsymbol{K} + \sigma^2 \boldsymbol{I})^{-1} \boldsymbol{k} + \sigma^2. \tag{23}$$

where $\boldsymbol{k} = [K(\boldsymbol{x}_1, \boldsymbol{x}_{N+1}), \ldots, K(\boldsymbol{x}_N, \boldsymbol{x}_{N+1})]^\top \in \mathbb{R}^N$.

GPs are linear predictors

Assume that $\mu(\boldsymbol{x}) = 0$ everywhere. Then we can write the predictive mean as,

$$m_{N+1} = \sum_{n=1}^{N} \alpha_n y_n, \tag{24}$$

where

$$\alpha_n = [\boldsymbol{K}^{-1}\boldsymbol{k}]_n \tag{25}$$

Question: what is the complexity of computing the predictive mean?

Recall the posterior predictive distribution in Bayesian linear regression

In Bayesian linear regression, we first compute the posterior mean of the weights.

In our notation from Slide 7, the posterior mean (under an uninformative prior) was,

$$\mathbb{E}[\boldsymbol{w} \mid \{\boldsymbol{x}_n, y_n\}_{n=1}^N] = \left(\boldsymbol{\Phi}^\top \boldsymbol{\Phi} + \boldsymbol{I}\right)^{-1} \left(\boldsymbol{\Phi}^\top \boldsymbol{y}\right) \tag{26}$$

so the prediction would be

$$\mathbb{E}[y_{N+1} \mid \boldsymbol{x}_{N+1}, \{\boldsymbol{x}_n, y_n\}_{n=1}^N] = \boldsymbol{\phi}(\boldsymbol{x}_{N+1})^\top \left(\boldsymbol{\Phi}^\top \boldsymbol{\Phi} + \boldsymbol{I}\right)^{-1} \left(\boldsymbol{\Phi}^\top \boldsymbol{y}\right) \tag{27}$$

Question: what is the complexity of computing the predictive mean in Bayesian linear regression?

Question: When is it more efficient to work with kernels rather than weights?

GP Classification

We've seen how Gaussian processes can be used for regression problems with $y_n \in \mathbb{R}$. What if we have binary observations under the following model,

$$f \sim \mathrm{GP}(\mu(\cdot), K(\cdot,\cdot)) \qquad\qquad y_n \sim \mathrm{Bern}(\sigma(f(\boldsymbol{x}_n))) \tag{28}$$

independently for $n = 1, \ldots, N$.

This is the GP analog of logistic regression.

As in logistic regression, the posterior is *not* available in closed form. All we know is,

$$p(\boldsymbol{f} \mid \{\boldsymbol{x}_n, y_n\}_{n=1}^N) \propto \left[\prod_{n=1}^N \mathrm{Bern}(y_n \mid \sigma(f(\boldsymbol{x}_n)))\right] \mathcal{N}(\boldsymbol{f} \mid \boldsymbol{\mu}, \boldsymbol{K}) \tag{29}$$

Also like logistic regression, we can use a Laplace approximation. But can we do better?

Lap 8: Gaussian processes, elliptical slice sampling, and Bayesian optimization

- **Model:** Gaussian processes
- **Algorithm: Elliptical slice sampling**
- **Application:** Bayesian optimization

References I

Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for regression*. MIT Press, 1996.

Amos J Storkey. Truncated covariance matrices and toeplitz methods in gaussian processes. In *Artificial Neural Networks, 1999. ICANN 99. Ninth International Conference on (Conf. Publ. No. 470)*, volume 1, pages 55–60 vol.1. unknown, February 1999.

John P Cunningham, Krishna V Shenoy, and Maneesh Sahani. Fast Gaussian process methods for point process intensity estimation. In *Proceedings of the 25th International Conference on Machine Learning*, pages 192–199, New York, NY, USA, July 2008. Association for Computing Machinery.

David Duvenaud, James Lloyd, Roger Grosse, Joshua Tenenbaum, and Ghahramani Zoubin. Structure discovery in nonparametric regression through compositional kernel search. In *International Conference on Machine Learning*, pages 1166–1174. PMLR, 2013.