Theory and Computation for Gaussian Processes

Michael Stein

University of Chicago

IPAM, February 2015

Funders & Collaborators

- US Department of Energy, US National Science Foundation (STATMOS)
- Mihai Anitescu, Jie Chen, Ying Sun

Gaussian processes

A process Z on a set S is called Gaussian if all finite-dimensional distributions are multivariate normal.

Gaussian process determined by its mean and covariance functions:

- $EZ(\boldsymbol{x}) = \mu(\boldsymbol{x})$
- $\operatorname{cov}\{Z(\boldsymbol{x}), Z(\boldsymbol{y})\} = K(\boldsymbol{x}, \boldsymbol{y})$

Estimating covariance function generally causes most computational problems, so assume mean is known (and 0) here.

If, as is often the case in computer experiments, only have one realization of process, need to assume some kind of at least local stationarity.

K is a valid autocovariance function ⇔ K is positive definite:

$$\sum_{\ell,k=1}^{n} \lambda_\ell \lambda_k K(\boldsymbol{x}_\ell - \boldsymbol{x}_k) \geq 0$$

for all finite n, all real $\lambda_1, \ldots, \lambda_n$ and all $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^d$.

Bochner's Theorem: K is a valid (complex-valued) continuous autocovariance function ⇔ for some positive finite measure F,

$$K(\boldsymbol{x}) = \int_{\mathbb{R}^d} e^{i\boldsymbol{\omega}'\boldsymbol{x}} F(\boldsymbol{d\omega}).$$

Assume here $F(\boldsymbol{d\omega}) = f(\omega)\boldsymbol{d\omega}$.

$$\text{derivatives of } Z \Leftrightarrow \text{derivatives of } K \Leftrightarrow \text{moments of } f$$

In one dimension:

$$Z \text{ has } m \text{ mean square derivatives} \Leftrightarrow K^{(2m)}(0) \text{ exists} \Leftrightarrow f \text{ has } 2m \text{ moments}$$

Examples:

- $K(x) = e^{-|x|}$ has no derivatives at 0

  $\Leftrightarrow f(\omega) \propto (1+\omega^2)^{-1}$ has no moments
- $K(x) = e^{-|x|}(1+|x|) = 1 - \frac{1}{2}x^2 + \frac{1}{3}|x|^3 + O(x^4)$ as $x \to 0$, so has exactly two derivatives

  $\Leftrightarrow f(\omega) \propto (1+\omega^2)^{-2}$ has two moments

These two examples are special cases of the *Matérn* model:

- $K(x) = \phi\beta|x|^{\nu}\mathcal{K}_{\nu}(\beta|x|)$ has $\lceil 2\nu - 1 \rceil$ derivatives
  $\Leftrightarrow f(\omega) \propto (\beta^2 + \omega^2)^{-\nu-\frac{1}{2}}$ has $\lceil 2\nu - 1 \rceil$ moments
  - $K(x) = e^{-|x|}$ $\Leftrightarrow \beta = 1$ and $\nu = 0.5$
  - $K(x) = e^{-|x|}(1 + |x|)$ $\Leftrightarrow \beta = 1$ and $\nu = 1.5$
- The Matérn model is a valid isotropic autocovariance function in any number of dimensions d with $f(\boldsymbol{\omega}) \propto (\beta^2 + |\boldsymbol{\omega}|^2)^{-\nu-\frac{d}{2}}$.

Sensible starting place for modeling spatial data.

- No "surprises" in the correlation structure.

Following 3 examples are of models with "surprises."

Example 1: $K(x) = (1 - |x|)^+$, “triangular” autocovariance function.

- $f(\omega) = (1 - \cos\omega)/(\pi\omega^2)$

Then

$$\lim_{\epsilon\downarrow 0} \operatorname{corr}\{Z(\epsilon) - Z(0), Z(t+\epsilon) - Z(t)\} = \begin{cases} 1 & t = 0 \\ -\frac{1}{2} & t = \pm 1 \\ 0 & \text{otherwise} \end{cases}$$

Example 2: $\boldsymbol{x} = (x_1, x_2)' \in \mathbb{R}^2$ and $\boldsymbol{\omega} = (\omega_1, \omega_2)'$,

$$K(\boldsymbol{x}) = e^{-|x_1|-|x_2|} \quad \text{and} \quad f(\boldsymbol{\omega}) = \frac{1}{\pi^2(1+\omega_1^2)(1+\omega_2^2)}.$$

Then

$$\lim_{\epsilon \downarrow 0} \operatorname{corr}\{Z(0,\epsilon) - Z(0,0), Z(s, t+\epsilon) - Z(s,t)\} = \begin{cases} e^{-|s|} & t = 0 \\ 0 & t \neq 0 \end{cases}$$

Example 3: Squared exponential (sometimes called "Gaussian") autocovariance function $K(x) = e^{-(x/\theta)^2}$.

- limit of Matérn as $\nu \to \infty$
- $f(\omega) = (\theta/\sqrt{4\pi})e^{-\theta^2\omega^2/4}$

Write e_n for error of best linear predictor of $Z(0)$ based on $Z(1/n), Z(2/n), \ldots, Z(1)$.
Using results in Lam and Loh (2000),

$$\text{corr}^2\{e_n, Z(1+1/n)\} = e^{-2/(n\theta^2)} = 1 - \frac{2}{n\theta^2} + O\left(n^{-2}\right)$$

as $n \to \infty$.

If added $Z(1+1/n)$ to other observations, mse of best predictor of $Z(0)$ would go down by factor $\approx 2/(n\theta^2)$.

Characterization of spectral densities

Recall f in the three examples:

1. $\dfrac{1-\cos\omega}{\pi\omega^2}$
2. $\dfrac{1}{\pi^2(1+\omega_1^2)(1+\omega_2^2)}$
3. $f(\omega) = \dfrac{\theta}{\sqrt{4\pi}} e^{-\theta^2\omega^2/4}$

All violate the condition:

For all finite R,

$$\lim_{|\boldsymbol{\omega}|\to\infty} \sup_{|\boldsymbol{\nu}|<R} \left| \frac{f(\boldsymbol{\omega}+\boldsymbol{\nu})}{f(\boldsymbol{\omega})} - 1 \right| = 0.$$

I claim this restriction is quite natural for spatial or space-time data. What about for computer experiments?

Fitting GP models to data

Suppose μ and K known up to finite number of parameters (and let's assume $\mu = 0$ here).

- If you are willing to take your GP model seriously, then the likelihood function is the "correct" summary of the information in the observations about these parameters.

Let $\boldsymbol{Z} \in \mathbb{R}^n$ be vector of observations. Write $K(\boldsymbol{\theta})$ for the covariance matrix of $\boldsymbol{Z}$, with $\boldsymbol{\theta} \in \mathbb{R}^p$ unknown.

Then the loglik is (ignoring an additive constant)

$$\ell(\boldsymbol{\theta}) = -\frac{1}{2}\log|K(\boldsymbol{\theta})| - \frac{1}{2}\boldsymbol{Z}'K(\boldsymbol{\theta})^{-1}\boldsymbol{Z}.$$

Whether one takes a frequentist or Bayesian perspective, need to compute this for many $\boldsymbol{\theta}$.

Computational challenges

Exact computations (kriging, Gaussian likelihoods) for large, irregularly sited datasets generally requires $O(n^3)$ computation and $O(n^2)$ memory.

- Computation is becoming cheap much faster than memory.
- Increasing emphasis on "matrix-free" methods in which never have to store an $n \times n$ matrix, even if requires more computation.

Options for large n:

- Parallel computation (Paciorek, *et al.*, R package 'bigGP').
- Use model that reduces computation and/or storage.
- Use approximate methods.
- Change what you compute.
- Some combination of these.

Some approaches to reducing computations/memory

Models

Markov random fields

Markov (in time)

Separable

Stationary (gridded data)

Low rank

Covariance tapering

Approximate/Alternative Computations

Spectral methods

Composite likelihoods

Score functions

Estimating equations

Stochastic trace approximants

Fast multipole methods

Empirical variograms

Not so easy to categorize covariance tapering.

- Important distinction between methods that give (exact/approximate) unbiased estimating equations under model of interest versus those that don't: for $g : \mathbb{R}^n \to \mathbb{R}^p$,

  $$g_{\boldsymbol{\theta}}(\boldsymbol{Z}) = \mathbf{0}$$

  is a set of unbiased estimating equations if $E_{\boldsymbol{\theta}} g_{\boldsymbol{\theta}}(\boldsymbol{Z}) = \mathbf{0}$ for all $\boldsymbol{\theta}$.

Change what you compute

In loglikelihood, need both $\log|K(\boldsymbol{\theta})|$ and $\boldsymbol{Z}'K(\boldsymbol{\theta})^{-1}\boldsymbol{Z}$.

Second term can often be done using iterative methods (e.g., conjugate gradient).

- Matrix-free (assuming can compute elements of $K(\boldsymbol{\theta})$ as needed).
- Iterative methods based on multiplying vectors by $K(\boldsymbol{\theta})$ quickly.
  - Will consider later.

Computing $\log|K(\boldsymbol{\theta})|$ more problematic, although people are trying (e.g., Zhang and Leithead, 2007).

- If flops are "free," then there is a matrix-free way to do it.

Instead, let's consider ways of avoiding it.

Score function

Writing K_i for $\frac{\partial}{\partial\theta_i}K(\boldsymbol{\theta})$ and suppressing dependence on θ, the score equations are

$$\frac{1}{2}\boldsymbol{Z}'K^{-1}K_iK^{-1}\boldsymbol{Z} - \frac{1}{2}\text{tr}(K^{-1}K_i) = 0$$

for $i = 1, \ldots, p$. No more determinant!

- First term requires just a single "solve" $K^{-1}\boldsymbol{Z}$.
- Second term requires n solves, which is no bargain, but is at least matrix-free (if use iterative methods).

For $\boldsymbol{U}_1, \ldots, \boldsymbol{U}_N$ random vectors in $\mathbb{R}^n$ with iid symmetric Bernoulli components,

$$\frac{1}{2}\boldsymbol{Z}'K^{-1}K_iK^{-1}\boldsymbol{Z} - \frac{1}{2N}\sum_{j=1}^{N}\boldsymbol{U}_j'K^{-1}K_i\boldsymbol{U}_j$$

is an unbiased estimate (conditional on $\boldsymbol{Z}$) of i'th component of score function. Hutchinson trace approximation.

If we can choose N much smaller than n, than this approximation reduces computations.

Stein, Chen and Anitescu (AoAS, 2013) shows that statistically efficiency of this approximation depends on the condition number of K.

- Number of iterations needed for iterative solver also related to this condition number! Preconditioning is key.
  - One way to precondition is to find C such that $C\boldsymbol{Z}$ has covariance matrix close to identity matrix.
- We showed that even when neighboring observations strongly dependent, with appropriate preconditioning, $N \approx 100$ can yield estimates with nearly same statistical efficiency as exact ML.

But $N = 100$ is still quite a few solves.

Unbiased estimating equations

Sun and Stein (in press, *JCGS*) consider modifying the score equations to reduce computations: for some matrix V,

$$\boldsymbol{Z}' V K_i K^{-1} \boldsymbol{Z} - \mathrm{tr}(V K_i) = 0 \tag{1}$$

for $i = 1, \ldots, p$ and

$$\boldsymbol{Z}' V K_i V \boldsymbol{Z} - \mathrm{tr}(V K_i V K) = 0 \tag{2}$$

for $i = 1, \ldots, p$ are both sets of unbiased estimating equations for $\boldsymbol{\theta}$. If $V = K^{-1}$, then both reduce to exact score equations.

Assuming V is available explicitly

- (1) requires one solve and (2) none
- $\mathrm{tr}(V K_i)$ may be much easier to compute than $\mathrm{tr}(V K_i V K)$.

If V is a sufficiently good approximation to K^{-1}, then a better approximation to score equations is $2 \times (1) - (2)$:

$$\boldsymbol{Z}'(2VK_iK^{-1} - VK_iV)\boldsymbol{Z} - \mathrm{tr}(2VK_i - VK_iVK) = 0. \tag{3}$$

How to pick V? Essentially the same as preconditioning problem.

- "Close" to K^{-1}.
- Easy to compute and store.
- Multiplying V times a vector is fast.

Choosing V based on a sparse inverse Cholesky approximation (Kolotolina and Yeremin (1993, *SIMAX*), although this is effectively what Vecchia (1988, *JRSSB*) was doing) provides a general approach to problem.

We give examples showing can get estimates with essentially same efficiency as MLE with a small fraction of the computational effort.

Matrix-vector multiplication

Iterative linear solvers require repeated matrix-vector multiplications.

For a dense, unstructured $n \times n$ matrix, this requires $O(n^2)$ flops.

- Sparse matrices reduce this computation.
- If data are on a (partial) grid and process is stationary, multiplication of vector by covariance matrix can be done with fast Fourier transform.

If data not on a grid, then fast multipole method (FMM) might help:

- Many covariance functions $K(x, y)$ are smooth away from the plane $x = y$, so properly chosen *off-diagonal* blocks of the covariance matrix of observations may be well-approximated by a low rank matrix.

My collaborator, Jie Chen, has found this approach works well in some circumstances, although it is not easy to implement.

Solving estimating equations

To find point estimates, have to compute estimating functions multiple times.

Several interesting computational issues arise:

- Can we use the value of, say, $K(\boldsymbol{\theta})^{-1}\boldsymbol{Z}$ to help us find $K(\boldsymbol{\theta}')^{-1}\boldsymbol{Z}$ for $\boldsymbol{\theta}'$ near $\boldsymbol{\theta}$?
- When using Hutchinson trace estimator, should one always use different $\boldsymbol{U}_j$'s for different $\boldsymbol{\theta}$?
- Should approximations to full score equations start out crude and then get sharper as estimate gets closer to final value?
  - Sun and Stein first found solution to (1) and used that as starting value for solving (3).
  - In Hutchinson trace estimator, one can use fairly small N when first updating $\boldsymbol{\theta}$ values and then increase N in last few iterations.
  - Preconditioners can get more elaborate in later iterations (e.g., less sparsity in sparse inverse Cholesky).

Composite loglikelihood

Approaches that consider all of the observations simultaneously aren't suitable for really large n and unstructured covariance matrices.

- Even computing all elements of the covariance matrix once when $n = 10^6$ (one day's worth of fairly modest resolution satellite data) is problematic.

For really large data sets, moderate loss of statistical efficiency may be a small concern. More important issues:

- Good models.
- (Approximately) unbiased estimation.
- Defensible uncertainty estimates.

Composite likelihood methods provide a way forward.

Let $\boldsymbol{p}_1, \ldots \boldsymbol{p}_b$ (prediction sets) and $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_b$ (conditioning sets) be subvectors of the observation vector $\boldsymbol{Z}$ and $w_1, \ldots, w_b$ nonnegative weights. Then

$$\sum_{j=1}^{b} w_j \log f_\theta(\boldsymbol{p}_j | \boldsymbol{c}_j)$$

is a weighted composite loglikelihood for $\boldsymbol{\theta}$.

To get standard errors, can use Godambe information matrix (generalization of Fisher information matrix). Calculating this exactly requires working with $n \times n$ matrices.

- But only needs to be done once.
- Can approximate using sampling if even that is not possible.

From both statistical and computational perspectives, I consider composite likelihoods a good solution to the large n problem.

- The problem with them is how to choose the exact form in any specific application.
- Unfortunately, I think coming up with an automated solution to this problem is difficult for space-time data.

What about for computer experiments?

- I think biggest issues are conceptual, not computational.

Random questions on GPs for computer experiments

GPs in computer experiments: “machine learning” or “statistics”?

- As far as I know, this approach was first developed by Sacks, Welch and others in the statistics community in the late 1980s.
- If “machine learning” is to mean anything, it should concern highly automated procedures requiring minimal human input.

If a computer model is very expensive to run, then it should be worth investing a fair amount of human effort to develop an effective emulator.

What might this human effort entail?

- Developing a good mean function, perhaps via simplified computer model.
- Figuring out which inputs matter most.
- Figuring out which interactions between inputs matter most.

With input vector $\mathbf{x} = (x_1, \ldots, x_p)$, suggests model of form

$$Z(\mathbf{x}) = \mu(\mathbf{x}) + \sum_{i=1}^{p} Z_i(x_i) + \sum_{1 \le i < j \le p} Z_{ij}(x_i, x_j) + \cdots$$

with Z_i's, Z_{ij}'s, etc., (perhaps) independent mean 0 Gaussian processes.

Random questions on GPs for computer experiments

Is the squared exponential a good starting point for form of covariance functions?

- For natural processes, no.
- For computer experiments, maybe. But if Z has *any* kind of singularity in domain of interest, will eventually have a problem.

My covariance matrices often turn out too close to singular to decompose. What should I do?

- Standard solution of adding a small multiple of identity (a "nugget") to covariance matrix is a bad idea.
- The right solution is to use higher precision arithmetic.

But doesn't the finite precision of my computer model output create justification for a nugget?

- Yes, but the nugget is a variance (the mean of the square of something), so for a nugget to represent truncation error, its size should be around the square of this error.

Random questions on GPs for computer experiments

Suppose, as in many particle systems, the dimension of input vector is large. Are dimension-reduction methods appropriate?

- They could be, but then you *really do* need to include a nugget effect in your GP.

At least for my starting designs, I use Latin hypercube samples because that appears to be the norm. Is that a good idea?

- Theoretical justification for LHS is that it works well for integrating a deterministic function over some box and the output is close to additive in the inputs (Stein, 1987).
- Other designs may be much better for interpolation, depending on covariance structure.

Questions on GPs for many particle systems

Inputs for many particle systems (atomic labels, interatomic distances) cannot take on arbitrary values. Does this matter?

- Not sure. But worth thinking about.

How do I use GPs on $\mathbb{R}^p$ given that:

- The properties of atoms are not well-described by a few real numbers.
- The size of input vector depends on the size of molecule.
- There are invariances such as to reordering the labels on atoms.

Don't think of as process on $\mathbb{R}^p$! Don't convert element names of atoms into (one or more) real numbers!

Questions on GPs for many particle systems

Develop special-purpose models for chemical properties of molecules rather than trying to use existing statistical/machine learning methods.

Possible way forward:

- Approximate response as sum over functions of contiguous subsets of atoms in molecule M.

  For a subset of atoms S, write $C(S) =$ for the labels and configuration of the atoms in S.

  Try a model of the form

  $$Z(M) = \sum_i Z_1(C(S_i)) \; + \; \text{interactions}$$

  Use all of the information in $C(S_i)$ in developing form for Z_1. Possible for S_i relatively small (functional group)?