Gaussian Processes - Part III
Advanced Topics

Philipp Hennig

MLSS 2013
30 August 2013

Max Planck Institute for Intelligent Systems
Department of Empirical Inference
Tübingen, Germany
Gaussians have been discovered before in virtually every area of science affected by uncertainty. In Thermodynamics, Brownian motion, and the Ornstein-Uhlenbeck process, stochastic calculus, stochastic differential equations, and Itō calculus are employed. Control theory utilizes stochastic control and the Kalman filter. Signal processing relies on filtering, and other communities use other names for the same concept: Kriging, Ridge-Regression, Kolmogorov-Wiener prediction, least-squares regression, Wiener process, Brownian bridge, etc.

Now: Gaussians show up in numerical methods, too. Quadrature, optimization, solving ODEs, control, . . .

Gaussian processes are central to many machine learning techniques, and all areas of quantitative science.
The big picture

we need a coherent framework for hierarchical machine learning

- uncertainty caused by finite computations should be accounted for
- uncertainty should propagate among numerical methods
- joint language required: probability

“off-the-shelf” methods are convenient, but not always efficient.
Numerical algorithms are the elements of inference inferring solutions of non-analytic problems

Numerical algorithms 
estimate (infer) an intractable property of a function given evaluations of function values.

- quadrature estimate \[ \int_a^b f(x) \, dx \] given \ \{f(x_i)\}
- optimization estimate \[ \arg \min_x f(x) \] given \ \{f(x_i), \nabla f(x_i)\}
- analysis estimate \[ x(t) \text{ under } x' = f(x, t) \] given \ \{f(x_i, t_i)\}
- control estimate \[ \min_u x(t, u) \text{ under } x' = f(x, t, u) \] \ \{f(x_i, t_i, u_i)\}

- even deterministic problems can be uncertain
- not a new idea \(^1\), but rarely studied

We need a theory of probabilistic numerics.

Gaussians, because of their connection to linear functions, are at the heart of probabilistic interpretations of numerics.

\(^1\)H. Poincaré, 1896, Diaconis 1988, O’Hagan 1992
Recall: GPs are closed under linear maps

\[ p(z) = \mathcal{N}(z; \mu, \Sigma) \implies p(Az) = \mathcal{N}(Az, A\mu, A\Sigma A^T) \]

- this is not restricted to finite linear operators (matrices) \( A \)
- \( A(x) = \mathbb{I}(a < x < b) \) gives \( Af = \int_a^b f(x) \, dx \)

\[
p\left( \int_a^b f(x) \, dx, \int_c^d f(x) \, dx \right) = \mathcal{N} \left[ \left( \int_a^b f(x) \, dx \right); \left( \int_c^d \mu(x) \, dx \right), \left( \int_a^b \int_a^b k(x, x') \, dx \, dx' \right) \right]
\]

\[
\left( \int_a^b \int_c^d k(x, x') \, dx \, dx' \quad \int_a^b \int_c^d k(x, x') \, dx \, dx' \right)
\]

\[
\left( \int_c^d \int_c^d k(x, x') \, dx \, dx' \quad \int_c^d \int_c^d k(x, x') \, dx \, dx' \right)
\]
Inferring $F = \int f$ from observations of $f$
Inferring $F = \int f$ from observations of $f$

quadrature
Quadrature with GPs
A O’Hagan, 1991; T Minka, 2000; M Osborne et al., 2012

- say what functions you expect to integrate
- find $\arg\min_X [k_{FfX} - k_{FfX} k_{fXfX}^{-1} k_{fXfX}]$ (depends on kernel!)
- Bayesian quadrature
Gaussian processes can be used to construct **quadrature** rules.
Inferring $f$ from observations of $F$

\[
\mu_f|_F = \mu_f + k_f F_X k^{-1}_{F_X F_X} (F_X - \int_X \mu) \quad k_{f f}|_F = k_{f f} - k_f F_X k^{-1}_{F_X F_X} k_{F_X f}
\]
Optimization
continuous, nonlinear, unconstrained

For $f : \mathbb{R}^N \rightarrow \mathbb{R}$, find local minimum $\arg \min f(x)$, starting at $x_0$.

An old idea: Newton’s method

$$f(x) \approx f(x_t) + (x - x_t)^\top \nabla f(x_t) + \frac{1}{2} (x - x_t)^\top \nabla^2 f(x_t)(x - x_t)$$

$\Rightarrow \quad x_{t+1} = x_t - B^{-1}(x_t) \nabla f(x_t)$

Cost: $O(N^3)$

High-dimensional optimization requires giving up knowledge in return for lower cost.
Quasi-Newton methods (think BFGS, DFP, . . . )
aka. variable metric optimization — low rank estimators for Hessians

- Instead of evaluating Hessian, build (low-rank) estimator fulfilling local difference relation . . .

\[ \nabla f(x_{t+1}) - \nabla f(x_t) = B_{t+1}(x_{t+1} - x_t) \]
\[ y_t = B_{t+1}s_t \]

- . . . otherwise close to previous estimator in \( \| B_{t+1} - B_t \|_{F,V} \)

- . . . so minimize regularised loss

\[
B_{t+1} = \arg \min_{B \in \mathbb{R}^{N \times N}} \left\{ \lim_{\beta \to 0} \frac{1}{\beta} \left\| y_t - Bs_t \right\|^2_V + \left\| B - B_t \right\|^2_{F,V} \right\} \\
= \lim_{\beta \to 0} \arg \max_B \mathcal{N}(y_t; Bs_t, \beta V)\mathcal{N}(\ddot{B}; \ddot{B}_t, V \otimes V) \\
= \arg \max_B \mathcal{N}\left[ B; B_t + \frac{(y_t - B ts_t)V s_t^T}{s_t^T V s_t}, V \otimes \left( V - V s s^T V \right) \right] \\
posterior

Quasi-Newton methods perform local maximum a-posteriori Gaussian inference on the Hessian’s elements.
Idea: replace
\[ \nabla f(x_{t+1}) - \nabla f(x_t) \approx B(x_{t+1} - x_t) \]
\[ \rightarrow = \int_{x_t}^{x_{t+1}} B(x) \, dx \]

Gaussian process prior on \( B(x^T, x) \)
\[ p(B) = \mathcal{GP}(B, B_0(x^T, x), k(x^T, x'^T) \otimes k(x, x')) \]

Gaussian likelihoods
\[ p(y_i(x^T) | B, s_i) = \lim_{\beta \to 0} \mathcal{N} \left( y_i; \sum_m s_{im} \int_0^1 B(x^T, x(t)) \, dt, k(x^T, x'^T) \otimes \beta \right) \]
\[ p(y_i(x)^T | B, s_i^T) = \lim_{\beta \to 0} \mathcal{N} \left( y_i^T; \sum_m s_{im}^T \int_0^1 B(x^T(t), x) \, dt, \beta \otimes k(x, x') \right) \]

posterior of same algebraic form as before, but with linear maps of nonlinear (integral of \( k \)) entries.

same computational complexity as L-BFGS (Nocedal, 1980): \( O(N) \)
A consistent model of the Hessian function
nonparametric inference on elements of the Hessian

P.H. & M. Kiefel, ICML 2012, JMLR 2013
A consistent model of the Hessian function

nonparametric inference on elements of the Hessian

P.H. & M. Kiefel, ICML 2012, JMLR 2013
A consistent model of the Hessian function
nonparametric inference on elements of the Hessian

P.H. & M. Kiefel, ICML 2012, JMLR 2013
A consistent model of the Hessian function

nonparametric inference on elements of the Hessian

P.H. & M. Kiefel, ICML 2012, JMLR 2013
A consistent model of the Hessian function
nonparametric inference on elements of the Hessian

P.H. & M. Kiefel, ICML 2012, JMLR 2013
Learning nonparametric models of Hessians allows:
- optimizing noisy functions
- dynamically changing functions
- parallelization
- …
Gaussian processes can be used in optimization.
GPs are closed under differentiation

\[
\mu_f / f'_X = \mu_f + k_f f'_X k^{-1}_{f'_X f'_X} (f'_X - \mu'_f)
\]

\[
k_{f f} / f'_X = k_{f f} - k_{f f} f'_X k^{-1}_{f'_X f'_X} k_{f'_X f}
\]

Rasmussen & Williams, 2006, §9.4
GPs can have multiple outputs

Reminder of Part I
Solving ODEs with GPs

observe \( c'(t) \), infer \( c(t) \)

\[
solve c'(t) = f(c(t), t) \text{ such that } c(0) = a \text{ and } c(1) = b
\]

\[
p(c(t)) = \mathcal{GP}(c; \mu_c, V \otimes k)
\]

\[
p(y_t \mid c) = \mathcal{N}(f(\hat{c}_t; t); \dot{c}_t, U)
\]

- repeatedly estimate \( \hat{c}_t \) using GP posterior mean to “observe”
  \[
c'(t) = f(\hat{c}_t) + \delta_f
\]

- estimate error in this observation by propagating Gaussian uncertainty through \( f \).

Recent work:

- Chkrebtii, Campbell, Girolami, Calderhead http://arxiv.org/abs/1306.2365
Solving ODEs with GPs

observe $c'(t)$, infer $c(t)$

solve $c'(t) = f(c(t), t)$ such that $c(0) = a$ and $c(1) = b$

$p(c(t)) = \mathcal{GP}(c; \mu_c, V \otimes k)$

$p(y_t | c) = \mathcal{N}(f(\hat{c}_t; t); \hat{c}_t, \hat{U})$

- repeatedly estimate $\hat{c}_t$ using GP posterior mean to “observe” $c'(t) = f(\hat{c}_t) + \delta_f$
- estimate error in this observation by propagating Gaussian uncertainty through $f$.

Recent work:

- Chkrebtii, Campbell, Girolami, Calderhead http://arxiv.org/abs/1306.2365
Solving ODEs with GPs

observe $c'(t)$, infer $c(t)$

\[ \text{solve } c'(t) = f(c(t), t) \text{ such that } c(0) = a \text{ and } c(1) = b \]

\[
p(c(t)) = \mathcal{GP}(c; \mu_c, V \otimes k) \\
p(y_t | c) = \mathcal{N}(f(\hat{c}_t; t); \dot{c}_t, U)
\]

\[ \triangleright \text{repeatedly estimate } \hat{c}_t \text{ using GP posterior mean to “observe”} \\
c'(t) = f(\hat{c}_t) + \delta_f \]

\[ \triangleright \text{estimate error in this observation by propagating Gaussian uncertainty through } f. \]

Recent work:

\[ \triangleright \text{Chkrebtii, Campbell, Girolami, Calderhead http://arxiv.org/abs/1306.2365} \]

\[ \triangleright \text{Hennig & Hauberg http://arxiv.org/abs/1306.0308} \]
The Advantages of a Probabilistic Formulation

joint uncertainty over solution

Hennig & Hauberg, under review
The Advantages of a Probabilistic Formulation

uncertainty over problem

Hennig & Hauberg, under review

![Diagram showing the relationship between variables and a solver]

- $x_1$ [arbitrary units]
- $x_2$ [arbitrary units]

$x \xrightarrow{}$ solver $\xrightarrow{} c$
Gaussian processes can be used to solve differential equations.
Lots of “Gaussian integrals” are known and can be used to map uncertainty through almost any function, see e.g. M. Deisenroth’s PhD, 2010.

- write \( f(x) = \sum_i \phi_i(x)^T w \) such that

\[
\int \phi_i(x) \mathcal{N}(x; \mu, \Sigma) \, dx \quad \int \phi_i(x) \phi_j(x) \mathcal{N}(x; \mu, \Sigma) \, dx
\]

is analytic.
Lots of “Gaussian integrals” are known
and can be used to map uncertainty through almost any function
see e.g. M. Deisenroth’s PhD, 2010

\[
\int f(x) \mathcal{N}(x; \mu, \Sigma) \, dx = \sum_i w_i \int \phi_i(x) \mathcal{N}(x; \mu, \Sigma) \, dx
\]

\[
\int f^2(x) \mathcal{N}(x; \mu, \Sigma) \, dx = \sum_i \sum_j w_i w_j \int \phi_i(x) \phi_j(x) \mathcal{N}(x; \mu, \Sigma) \, dx
\]

- also works if \( f \in \mathbb{R}^N \), and if \( p(w) = \mathcal{N}(w; m, V) \)
Some useful Gaussian integrals
an expressive basis set for function approximation

\[
\int x^p \mathcal{N}(x; 0, \sigma^2) \, dx = \begin{cases} 
0 & \text{if } p \text{ odd} \\
\sigma^p \prod_{i=1:2:p-1} i & \text{if } p \text{ even}
\end{cases}
\]

\[
\int |x|^p \mathcal{N}(x; 0, \sigma^2) \, dx = \frac{\sigma^p}{\sqrt{\pi}} 2^{p/2} \Gamma \left( \frac{p + 1}{2} \right)
\]

\[
\int (x - m)^\top V (x - m) \mathcal{N}(x; \mu, \Sigma) \, dx = (\mu - m)^\top V (\mu - m) + \text{tr}[V \Sigma]
\]

\[
\int \mathcal{N}(x; a, A) \mathcal{N}(x; b, B) \, dx = \mathcal{N}(a, b, A + B)
\]

\[
\int \int_{-\infty}^{(x-m)/s} \mathcal{N}(\tilde{x}, 0, 1) \, d\tilde{x} \, \mathcal{N}(x; \mu, \sigma^2) \, dx = \int_{-\infty}^{(\mu-m)/\sqrt{s^2+\sigma^2}} \mathcal{N}(\tilde{x}, 0, 1) \]

\[
= \frac{1}{2} \left[ 1 + \text{erf} \left( \frac{\mu - m}{\sqrt{2(s^2 + \sigma^2)}} \right) \right]
\]

Expected values of monomials for moment computations

$$\int x^p \mathcal{N}(x; \mu, \sigma) = \sigma^p (-i \sqrt{2} \text{sgn } \mu)^p U \left( -\frac{p}{2}, 1, -\frac{1}{2} \frac{\mu^2}{\sigma^2} \right) \quad p \in \mathbb{N}_0$$

where $U$ is Tricomi’s confluent hypergeometric function (cheap)
Expected values of monomials for moment computations

\[ \int x^p \mathcal{N}(x; \mu, \sigma) = \sigma^p (-i \sqrt{2} \text{sgn } \mu)^p U \left( -\frac{p}{2}, \frac{1}{2}, -\frac{1}{2} \frac{\mu^2}{\sigma^2} \right) \quad p \in \mathbb{N}_0 \]

where \( U \) is Tricomi’s confluent hypergeometric function (cheap)
Expected values of monomials for moment computations

\[
\int x^p \mathcal{N}(x; \mu, \sigma) = \sigma^p (-i \sqrt{2} \text{sgn } \mu)^p U \left( -\frac{p}{2}, \frac{1}{2}, -\frac{1}{2} \frac{\mu^2}{\sigma^2} \right) \quad p \in \mathbb{N}_0
\]

where $U$ is Tricomi’s confluent hypergeometric function (cheap)
Expected values of monomials
for moment computations

\[
\int x^p \mathcal{N}(x; \mu, \sigma) = \sigma^p (-i\sqrt{2} \text{sgn} \mu)^p U \left( -\frac{p}{2}, \frac{1}{2}, -\frac{1}{2} \frac{\mu^2}{\sigma^2} \right) \quad p \in \mathbb{N}_0
\]

where \( U \) is Tricomi’s confluent hypergeometric function (cheap)
Expected values of error functions
for moment computations

\[ \Phi(x) = \int_{-\infty}^{x} \mathcal{N}(x; 0, 1) \, dx \]
\[ z = \frac{\mu - m}{\sqrt{v^2 + \sigma^2}} \]

\[ \int \Phi \left( \frac{x - m}{v} \right) \mathcal{N}(x; \mu, \sigma) = \Phi(z) \]
Expected values of error functions
for moment computations
e.g. Rasmussen & Williams, §3.9

\[ \Phi(x) = \int_{-\infty}^{x} \mathcal{N}(x; 0, 1) \, dx \]

\[ z = \frac{\mu - m}{\sqrt{v^2 + \sigma^2}} \]

\[ \int x \Phi \left( \frac{x - m}{v} \right) \mathcal{N}(x; \mu, \sigma) = \mu \Phi(z) + \frac{\sigma^2}{\sqrt{v^2 + \sigma^2}} \mathcal{N}(z; 0, 1) \]
Expected values of error functions for moment computations

\[ \Phi(x) = \int_{-\infty}^{x} \mathcal{N}(x; 0, 1) \, dx \]

\[ z = \frac{\mu - m}{\sqrt{v^2 + \sigma^2}} \]

\[ \int x^2 \Phi \left( \frac{x - m}{v} \right) \mathcal{N}(x; \mu, \sigma) = (\mu^2 + \sigma^2) \Phi(z) + \left( 2\mu \frac{\sigma^2}{\sqrt{v^2 + \sigma^2}} - \frac{z\sigma^4}{\sqrt{v^2 + \sigma^2}} \right) \mathcal{N}(z; 0, 1) \]
Treating Cancer with GPs

Analytical Probabilistic Modelling in Radiation Therapy

Bangert, Hennig, Oelfke, 2013

image source: wikipedia
the data

CT images

source: wikipedia
the parameter space
multi-beam plans

source: M. Bangert, DKFZ
setup errors can be disastrous
human bodies are complicated

- setup errors of 5mm and less can drastically change the clinical outcome
- accounting for these errors is currently not clinical practice
- some prior work\textsuperscript{2,3}, but problems of computational cost

\textsuperscript{2}Unkelbach et al.: Reducing the sensitivity of IMPT treatment plans to setup errors and range uncertainties via probabilistic treatment planning. 2009 Med. Phys. 36: 149
Propagating Gaussian uncertainty through nonlinearities using integrals against Gaussian measures

- works on virtually any continuous function
- guaranteed numerical precision, fixed at design time
- low computational cost: just matrix-matrix multiplications
Error Bars on Radiation Dose
setup error 1mm ±2mm, range error 3%

Bangert, Hennig, Oelfke, 2013

FIG. 5. Nominal dose $d$ (first row), expectation value $E[d]$ (second row), and standard deviation for delivery in 1 (third row), 10 (forth row), and 30 fractions (fifth row) for 9 photon beams (left column) and 3 proton beams (right column). We assume a range error of 3%, a systematic setup error of 1mm, and a random setup error of 2mm. The color scale in the first row applies to the nominal dose and the expectation value and the color scale in the third row to the standard deviation.
Gaussian algebra can be used to build numerical methods for probabilistic computations.
Gaussians provide the linear algebra of inference

- products of Gaussians are Gaussians

\[ \mathcal{N}(x; a, A)\mathcal{N}(x; b, B) = \mathcal{N}(x; c, C)\mathcal{N}(a; b, A + B) \]

\[ C := (A^{-1} + B^{-1})^{-1} \quad c := C(A^{-1}a + B^{-1}b) \]

- marginals of Gaussians are Gaussians

\[
\int \mathcal{N}\left[ \begin{pmatrix} x \\ y \end{pmatrix}; \begin{pmatrix} \mu_x \\ \mu_y \end{pmatrix}, \begin{pmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{pmatrix} \right] \, dy = \mathcal{N}(x; \mu_x, \Sigma_{xx})
\]

- (linear) conditionals of Gaussians are Gaussians

\[
p(x \mid y) = \frac{p(x, y)}{p(y)} = \mathcal{N} \left( x; \mu_x + \Sigma_{xy} \Sigma_{yy}^{-1} (y - \mu_y), \Sigma_{xx} - \Sigma_{xy} \Sigma_{yy}^{-1} \Sigma_{yx} \right)
\]

- linear projections of Gaussians are Gaussians

\[
p(z) = \mathcal{N}(z; \mu, \Sigma) \quad \Rightarrow \quad p(Az) = \mathcal{N}(Az, A\mu, A\Sigma A^\top)
\]

- analytical integrals allow moment matching “projection to Gaussians”

\[
\int f(x)\mathcal{N}(x; \mu, \Sigma) = \text{known} \quad \text{e.g. for } f(x) = x^p, \text{erf}(x), \mathcal{N}(x), x^\top V x
\]
Generalised linear models learn nonlinear functions

\[ f(x) = \phi(x)^\top w \quad p(w) = \mathcal{N}(w; \mu, \Sigma) \]
Generalised linear models learn nonlinear functions

\[ f(x) = \phi(x)^\top w \quad p(w) = \mathcal{N}(w; \mu, \Sigma) \]
infinite feature sets give **nonparametric** models

\[ p(f) = \mathcal{GP}(f; \mu, k) \]
Gaussian processes are powerful, but not magic

**powerful models**
- kernels use **infinitely many features**
- kernels can be **combined** to form expressive models
- hyperparameters can be learned by **hierarchical inference**
- individual **nonlinear effects** can be separated from **superpositions**
- some kernels are **universal**

**but no magic**
- every model has parameters chosen **a priori**
- universal kernels can have **logarithmic convergence rate**
Gaussian processes are at heart of probabilistic numerics

Gaussians have great algebraic properties

- GPs are closed under linear projections, including
  - differentiation
  - integration
- GPs can be integrated against an expressive set of functions

They are the elementary tool of probabilistic numerics

- quadrature rules can be derived from GPs
- quasi-Newton optimization can be generalised using GPs
- GPs allow ODE solvers capable of probabilistic input
- moment matching allows numerical probabilistic computations

Numerics is about turning nonlinear problems into linear ones. That’s what Gaussian regression does.
Questions?
Bibliography

- T. O'Hagan
  Bayes-Hermite Quadrature
  J. Statistical Planning and Inference 29, pp. 245–260

- C.E. Rasmussen & C.K.I. Williams
  Gaussian Processes for Machine Learning
  MIT Press, 2006

- T. Minka
  Deriving quadrature rules from Gaussian processes
  Tech. Report 2000

  Active Learning of Model Evidence Using Bayesian Quadrature
  Advances in NIPS, 2012

- P. Hennig & M. Kiefel
  Quasi-Newton Methods: a new direction
  ICML 2012 (short form), and JMLR 14 (2013), pp. 807–829

- P. Hennig
  Fast Probabilistic Optimization from Noisy Gradients
  ICML 2013

- J. Skilling
  Bayesian solution of ordinary differential equations
  Maximum Entropy and Bayesian Methods, 1991

- O. Chkrebtii, D.A. Campbell, M.A. Girolami, B. Calderhead
  Bayesian Uncertainty Quantification for Differential Equations
  http://arxiv.org/abs/1306.2365

- M. Bangert, P. Hennig, U. Oelfke
  Analytical probabilistic modeling for radiation therapy treatment planning
  Physics in Medicine and Biology, 2013, in press