Extended and Unscented Gaussian Processes

Daniel M. Steinberg
daniel.steinberg@nicta.com.au
NICTA

Edwin V. Bonilla
e.bonilla@unsw.edu.au
The University of New South Wales

NICTA

UNSW AUSTRALIA

Inverse Problems

In many problems in science and engineering we have access to a **forward** or system model, $g(\cdot)$:

$$\mathbf{y} = g(\mathbf{f}) + \boldsymbol{\epsilon}$$

- We can measure the outputs of the system, $\mathbf{y}$, but the inputs, $\mathbf{f}$, are **latent**.
- We wish to infer these inputs *without* access to the inverse system model, $g^{-1}(\cdot)$.
- $\mathbf{y}$ may be a continuous process or path (robot arm motion), so $\mathbf{f}$ can be a **Gaussian process** (GP).

Aims

1. Compute a posterior distribution over $\mathbf{f}$.
2. Avoid 'hand-coding' methods for every new $g(\cdot)$, i.e. generic inference for non-linear likelihoods.
3. The gradients, $\partial g(\mathbf{f})/\partial \mathbf{f}$, may not be known.
4. Avoid expensive simulations (cf. MCMC).

GPs with nonlinear likelihoods

Prior on latent functions $\mathbf{f}$ at locations $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^{N}$,

$$p(\mathbf{f}) = \mathcal{N}(\mathbf{f}|\boldsymbol{\mu}, \mathbf{K}), \qquad (1)$$

where $\boldsymbol{\mu}, \mathbf{K}$ evaluate the mean function $\mu(\mathbf{x})$ and the covariance function $k(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta})$ at all observed locations.

Likelihood encodes $\mathbf{y} = g(\mathbf{f}) + \text{noise}$,

$$p(\mathbf{y}|\mathbf{f}) = \mathcal{N}\big(\mathbf{y}|g(\mathbf{f}), \boldsymbol{\Sigma}\big) = \prod_{n=1}^{N} \mathcal{N}\big(y_n|g(f_n), \sigma^2\big). \qquad (2)$$

This factorisation simplifies computation but is not required.

The **posterior** is the solution to our inverse problem:

$$p(\mathbf{f}|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{f})\, p(\mathbf{f}), \qquad (3)$$

which is generally *intractable* due to *non-linear* $g(\mathbf{f})$.

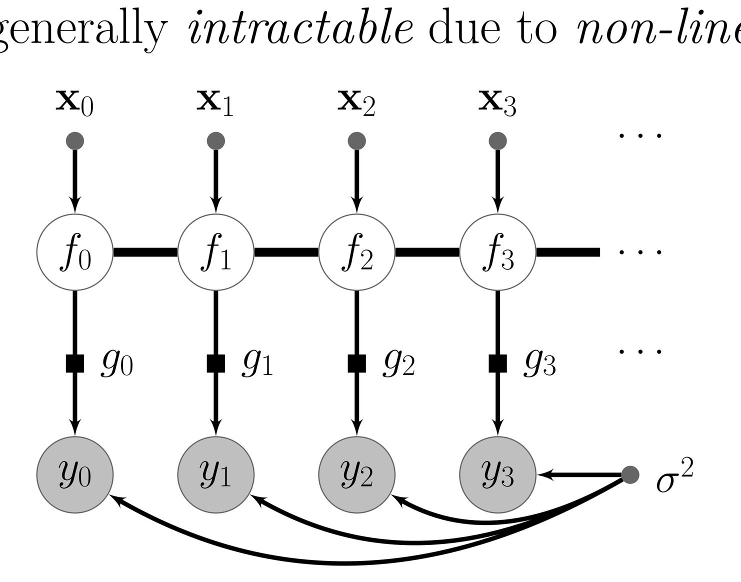

Figure 1: A Gaussian process for inversion problems – the mapping from f_n to y_n is given by the nonlinear forward model $g(f_n)$.

Variational Inference

We approximate $p(\mathbf{f}|\mathbf{y}) \approx q(\mathbf{f}) = \mathcal{N}(\mathbf{f}|\mathbf{m}, \mathbf{C})$, via the maximisation of the evidence *lower bound*:

$$\mathcal{F} = \langle \log p(\mathbf{y}|\mathbf{f}) \rangle_{q\mathbf{f}} - \mathrm{KL}[q(\mathbf{f}) \,\|\, p(\mathbf{f})]. \qquad (4)$$

Main difficulty: 'Intractable' expected log likelihood:

$$\langle \log p(\mathbf{y}|\mathbf{f}) \rangle_{q\mathbf{f}} = -\frac{1}{2\sigma^2} \Big\langle \big(\mathbf{y} - g(\mathbf{f})\big)^{\top} \big(\mathbf{y} - g(\mathbf{f})\big) \Big\rangle_{q\mathbf{f}} + \ldots$$

Our Solution:

1. **Linearise** the forward model:

$$g(\mathbf{f}) \approx \tilde{g}(\mathbf{f}) = \mathbf{A}\mathbf{f} + \mathbf{b}, \qquad (5)$$

and obtain linearised objective $\tilde{\mathcal{F}}$.

2. **Newton method** on $\tilde{\mathcal{F}}$ to find $\mathbf{m}$, and 'closed-form' updates for $\mathbf{C}$:

$$\mathbf{m}_{k+1} = (1 - \alpha)\,\mathbf{m}_k + \alpha\boldsymbol{\mu} + \alpha\mathbf{H}_k(\mathbf{y} - \mathbf{b}_k - \mathbf{A}_k\boldsymbol{\mu}),$$
$$\mathbf{C} = (\mathbf{I}_N - \mathbf{H}_k\mathbf{A}_k)\mathbf{K},$$

where $\mathbf{H}_k = \mathbf{K}\mathbf{A}_k^{\top}\left(\boldsymbol{\Sigma} + \mathbf{A}_k\mathbf{K}\mathbf{A}_k^{\top}\right)^{-1}$.

3. **Methods:** *How to linearise → EGP vs. UGP.*

Extended Gaussian Process (EGP)

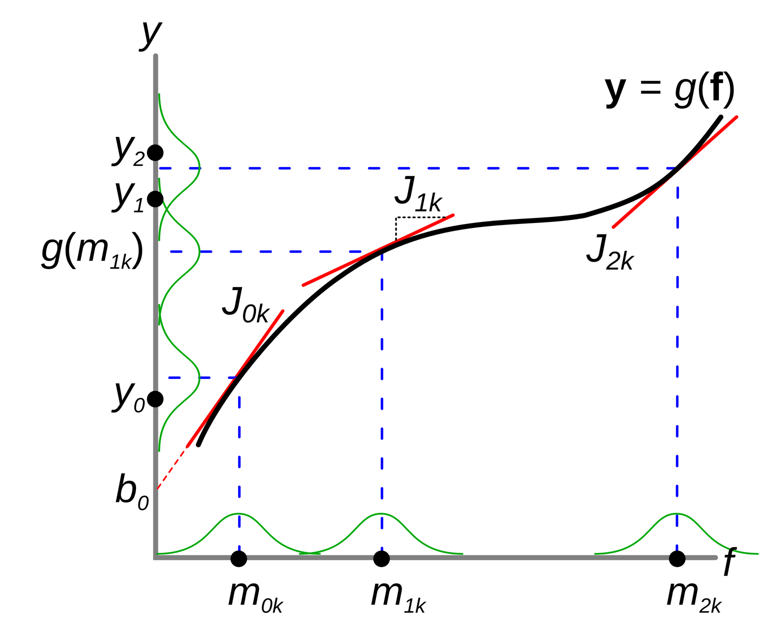

Figure 2: EGP linearises $g(\cdot)$ using a 1st-order Taylor expansion.

First order Taylor series expansion about $\mathbf{m}_k$,

$$g(f_n) \approx g(m_{nk}) + J_{m_{nk}}(f_n - m_{nk}),$$

and $J_{m_{nk}} = \partial g(m_{nk})/\partial m_{nk}$. Then equating coefficients,

$$\mathbf{A}_k = \mathrm{diag}([J_{m_{0k}}, \ldots, J_{m_{Nk}}]), \qquad (6)$$
$$\mathbf{b}_k = [g(m_{0k}) - J_{m_{0k}}m_{0k}, \ldots, g(m_{Nk}) - J_{m_{Nk}}m_{Nk}]^{\top}.$$

This results in the *Gauss Newton* method.

Unscented Gaussian Process (UGP)

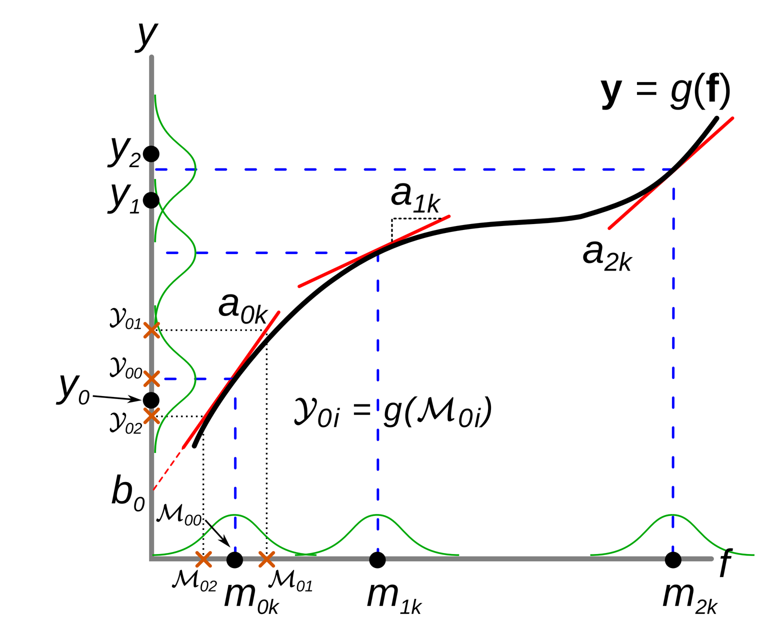

Figure 4: UGP linearises $g(\cdot)$ using the unscented transform.

Make sigma points [2] $\mathcal{M}_{ni}$ and $\mathcal{Y}_{ni}$,

$$\mathcal{M}_{n0} = m_{nk},$$
$$\mathcal{M}_{n1} = m_{nk} + \sqrt{(1+\kappa)\, C_{nnk}},$$
$$\mathcal{M}_{n2} = m_{nk} - \sqrt{(1+\kappa)\, C_{nnk}},$$
$$\mathcal{Y}_{ni} = g(\mathcal{M}_{ni}),$$

then solving N scalar linear regression problems,

$$\underset{a_{nk}, b_{nk}}{\mathrm{argmin}} \sum_{i=0}^{2} \|\mathcal{Y}_{ni} - (a_{nk}\mathcal{M}_{ni} + b_{nk})\|_2^2$$

gives,

$$\mathbf{A}_k = \mathrm{diag}([a_{0k}, \ldots, a_{Nk}]), \qquad (7)$$
$$\mathbf{b}_k = [\bar{y}_0 - a_{0k}m_{0k}, \ldots, \bar{y}_N - a_{Nk}m_{Nk}]^{\top}.$$

Here $a_{nk} = \Gamma_{ym,nk} C_{nnk}^{-1}$, $\Gamma_{ym,nk}$ is the cross-covariance between $\mathcal{M}_{ni}$ and $\mathcal{Y}_{ni}$ $\forall i$, and $\bar{y}_n = \sum_{i=0}^{2} w_i \mathcal{Y}_{ni}$.

Key Results

- UGP treats the likelihood as a 'black box' by not requiring knowledge of its form or its derivatives.
- $\mathbf{A}$ is a *diagonal* matrix because of the factorising likelihood in (2) – so similar complexity as Laplace approx.
- The iterative extended and sigma-point Kalman filters are specific instances of our variational framework.

Experiments

Synthetic inversion problems

Table 1: Performance on synthetic inversion problems.

$g(\mathbf{f})$	Algorithm	NLPD f^*		SMSE f^*		SMSE y^*	
		mean	std.	mean	std.	mean	std.
$\mathbf{f}$	UGP	-0.90046	0.06743	0.01219	0.00171	–	–
	EGP	-0.89908	0.06608	0.01224	0.00178	–	–
	[1]	-0.27590	0.06884	0.01249	0.00159	–	–
	GP	**-0.90278**	0.06988	**0.01211**	0.00160	–	–
$\mathbf{f}^3 + \mathbf{f}^2 + \mathbf{f}$	UGP	**-0.23622**	1.72609	0.01534	0.00202	**0.02184**	0.00525
	EGP	-0.22325	1.76231	**0.01518**	0.00203	**0.02184**	0.00528
	[1]	-0.14559	0.04026	0.06733	0.01421	0.02686	0.00266
$\exp(\mathbf{f})$	UGP	-0.75475	0.32376	**0.13860**	0.04833	**0.03865**	0.00403
	EGP	**-0.75706**	0.32051	0.13971	0.04842	0.03872	0.00411
	[1]	-0.08176	0.10986	0.17614	0.04845	0.05956	0.01070
$\sin(\mathbf{f})$	UGP	**-0.59710**	0.22861	**0.03305**	0.00840	0.11513	0.00521
	EGP	-0.59705	0.21611	0.03480	0.00791	**0.11478**	0.00532
	[1]	-0.04363	0.03883	0.05913	0.01079	0.11890	0.00652
$\tanh(2\mathbf{f})$	UGP	**0.01101**	0.60256	**0.15703**	0.06077	**0.08767**	0.00292
	EGP	0.57403	1.25248	0.18739	0.07869	0.08874	0.00394
	[1]	0.15743	0.14663	0.16049	0.04563	0.09434	0.00425

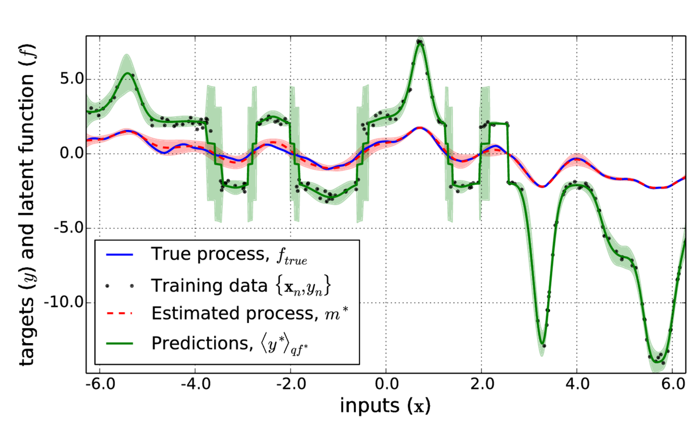

Figure 3: Learning the UGP with the forward model $g(\mathbf{f}) = 2 \times \mathrm{sign}(\mathbf{f}) + \mathbf{f}^3$.

Binary classification

Table 2: Perf. on USPS for classes '3' and '5'.

Algorithm	NLP y^*	Error rate (%)
GP – Laplace	0.11528	2.9754
GP – EP	0.07522	2.4580
GP – VB	0.10891	3.3635
SVM (RBF)	0.08055	2.3286
Logistic Reg.	0.11995	3.6223
UGP	**0.07290**	**1.9405**
EGP	0.08051	2.1992

Learning the EGP and the UGP

Variational-EM updates:

- Optimize posterior parameters $\mathbf{m}, \mathbf{C}$
- Optimize hyperparameters and forward model's parameters for fixed posterior parameters

References

[1] M. Opper and C. Archambeau. The variational Gaussian approximation revisited. *Neural computation*, 21(3):786–792, 2009.

[2] S.J. Julier and J.K. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, Mar 2004.

Code

Github – https://github.com/NICTA/linearizedGP