Modern Gaussian Processes: Scalable Inference and Novel Applications

(Part III) Applications, Challenges & Opportunities

Edwin V. Bonilla and Maurizio Filippone

CSIRO's Data61, Sydney, Australia and EURECOM, Sophia Antipolis, France

July 14th, 2019

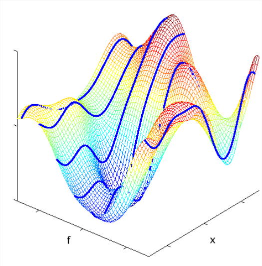
Outline

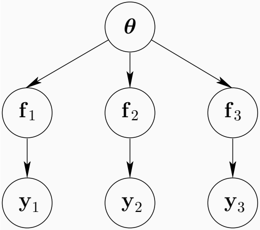
1. Multi-task Learning
2. The Gaussian Process Latent Variable Model (GPLVM)
3. Bayesian Optimisation
4. Deep Gaussian Processes
5. Other Interesting GP/DGP-based Models

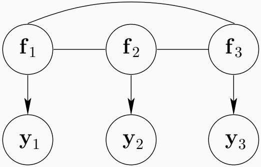
Multi-task Learning

Data Fusion and Multi-task Learning (1)

- Sharing information across tasks/problems/modalities
- Very little data on test task
- Can model dependencies *a priori*
- Correlated GP prior over latent functions



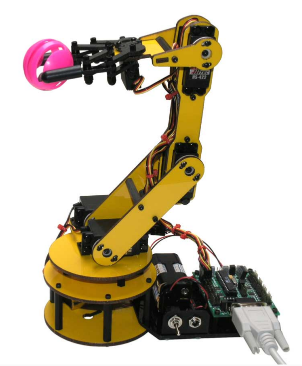




Data Fusion and Multi-task Learning (2)

Multi-task GP (Bonilla et al, NeurIPS, 2008)

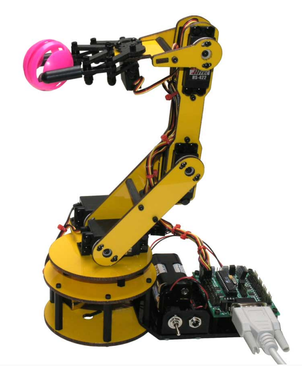
- $\mathrm{Cov}(f_\ell(\mathbf{x}), f_m(\mathbf{x}')) = \mathbf{K}^f_{\ell m}\kappa(\mathbf{x}, \mathbf{x}')$
- $\mathbf{K}$ can be estimated from data
- Kronecker-product covariances
  - 'Efficient' computation
- Robot inverse dynamics (Chai et al, NeurIPS, 2009)



Data Fusion and Multi-task Learning (2)

Multi-task GP (Bonilla et al, NeurIPS, 2008)

- $\mathrm{Cov}(f_\ell(\mathbf{x}), f_m(\mathbf{x}')) = \mathbf{K}^f_{\ell m}\kappa(\mathbf{x}, \mathbf{x}')$
- $\mathbf{K}$ can be estimated from data
- Kronecker-product covariances
  - ‘Efficient’ computation
- Robot inverse dynamics (Chai et al, NeurIPS, 2009)



Generalisations and other settings:

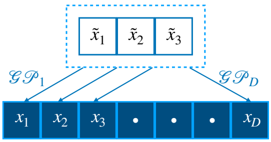
- Convolution formalism (Alvarez and Lawrence, JMLR, 2011)
- GP regression networks (Wilson et al, ICML, 2012)
- Many more ...

The Gaussian Process Latent Variable Model (GPLVM)

Non-linear Dimensionality Reduction with GPs

The **Gaussian Process Latent Variable Model** (GPLVM; Lawrence, NeurIPS, 2004):

- Probabilistic non-linear dimensionality reduction
- Use independent GPs for each observed dimension
- Estimate latent projections of the data via maximum likelihood



Modelling of Human Poses with GPLVMs (Grochow et al, SIGGRAPH 2004)

Style-Based Inverse Kinematics: Given a set of constraints, produce the most likely pose

- High dimensional data derived from pose information
  - joint angles, vertical orientation, velocity and accelerations
- GPLVM used to learn low-dimensional trajectories
- GPLVM predictive distribution used in cost function for finding new poses with constraints

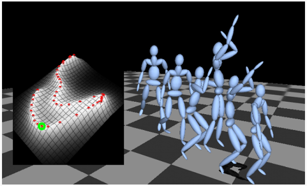


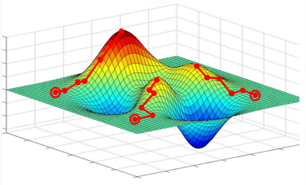
Fig. and cool videos at
`http://grail.cs.washington.edu/projects/styleik/`

Bayesian Optimisation

Probabilistic Numerics: Bayesian Optimisation (1)

Optimisation of black-box functions:

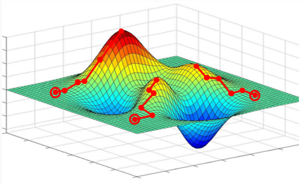
- Do not know their implementation
- Costly to evaluate
- Use GPs as surrogate models



Probabilistic Numerics: Bayesian Optimisation (1)

Optimisation of black-box functions:

- Do not know their implementation
- Costly to evaluate
- Use GPs as surrogate models



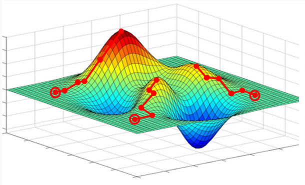
Vanilla BO iterates:

1. Get a few samples from true function

Probabilistic Numerics: Bayesian Optimisation (1)

Optimisation of black-box functions:

- Do not know their implementation
- Costly to evaluate
- Use GPs as surrogate models



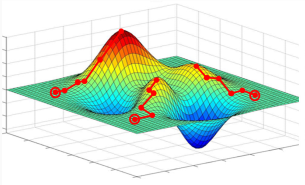
Vanilla BO iterates:

1. Get a few samples from true function
2. Fit a GP to the samples

Probabilistic Numerics: Bayesian Optimisation (1)

Optimisation of black-box functions:

- Do not know their implementation
- Costly to evaluate
- Use GPs as surrogate models



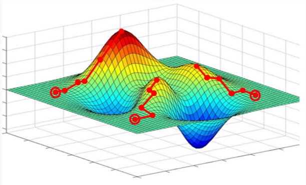
Vanilla BO iterates:

1. Get a few samples from true function
2. Fit a GP to the samples
3. Use GP predictive distribution along with acquisition function to suggest new sample locations

Probabilistic Numerics: Bayesian Optimisation (1)

Optimisation of black-box functions:

- Do not know their implementation
- Costly to evaluate
- Use GPs as surrogate models



Vanilla BO iterates:

1. Get a few samples from true function
2. Fit a GP to the samples
3. Use GP predictive distribution along with acquisition function to suggest new sample locations

What are sensible acquisition functions?

Bayesian Optimisation (2)

A taxonomy of algorithms proposed by D. R. Jones (2001)

- $\mu(\mathbf{x}_\star), \sigma^2(\mathbf{x}_\star)$: pred. mean, variance
- $\mathcal{I} \stackrel{\text{def}}{=} f(\mathbf{x}_\star) - f_{\text{best}}$: pred. improvement

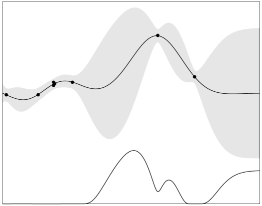


Fig. from Boyle (2007)

Bayesian Optimisation (2)

A taxonomy of algorithms proposed by D. R. Jones (2001)

- $\mu(\mathbf{x}_\star), \sigma^2(\mathbf{x}_\star)$: pred. mean, variance
- $\mathcal{I} \stackrel{\text{def}}{=} f(\mathbf{x}_\star) - f_{\text{best}}$: pred. improvement
- **Expected improvement**:

$$\mathrm{EI}(\mathbf{x}_\star) = \int_0^\infty \mathcal{I} p(\mathcal{I}) d\mathcal{I}$$

  - Simple 'analytical form'
  - Exploration-exploitation

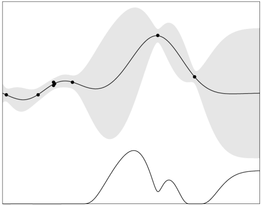


Fig. from Boyle (2007)

Bayesian Optimisation (2)

A taxonomy of algorithms proposed by D. R. Jones (2001)

- $\mu(\mathbf{x}_\star), \sigma^2(\mathbf{x}_\star)$: pred. mean, variance
- $\mathcal{I} \stackrel{\text{def}}{=} f(\mathbf{x}_\star) - f_{\text{best}}$: pred. improvement
- **Expected improvement**:

$$\text{EI}(\mathbf{x}_\star) = \int_0^\infty \mathcal{I} p(\mathcal{I}) d\mathcal{I}$$

  - Simple 'analytical form'
  - Exploration-exploitation

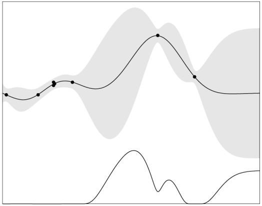


Fig. from Boyle (2007)

Main idea: Sample $\mathbf{x}_\star$ so as to maximize the EI

Bayesian Optimisation (3)

Many cool applications of BO and probabilistic numerics:

- Optimisation of ML algorithms (Snoek et al, NeurIPS, 2012)
- Preference learning (Chu and Gahramani, ICML 2005; Brochu et al, NeurIPS, 2007; Bonilla et al, NeurIPS, 2010)
- Multi-task BO (Swersky et al, NeurIPS, 2013)
- Bayesian Quadrature

See `http://probabilistic-numerics.org/` and references therein

Deep Gaussian Processes

The Deep Learning Revolution

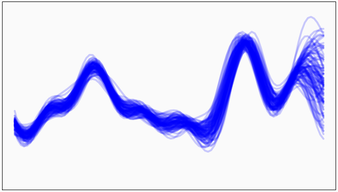
- Large representational power
- *Big data* learning through stochastic optimisation
- Exploit GPU and distributed computing
- Automatic differentiation
- Mature development of regularization (e.g., dropout)
- Application-specific representations (e.g., convolutional)

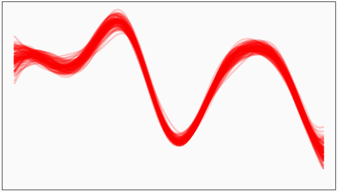
Is There Any Hope for Gaussian Process Models?

Can we exploit what made Deep Learning successful for practical and scalable learning of Gaussian processes?

Deep Gaussian Processes

- Composition of Processes

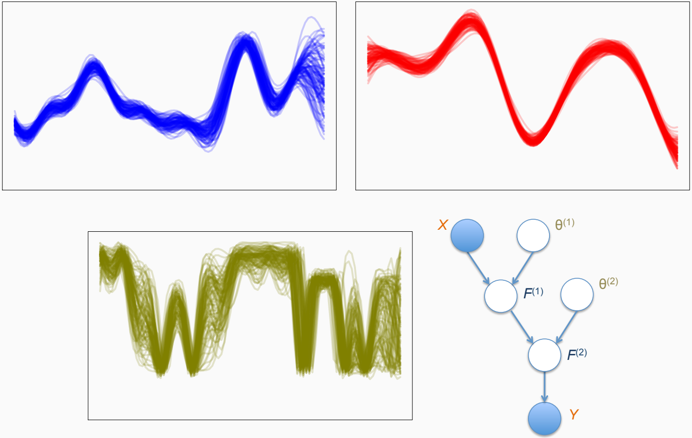




$$(f \circ g)(x)??$$

Teaser — Modern GPs: Flexibility and Scalability

- Composition of processes: Deep Gaussian Processes



Damianou and Lawrence, *AISTATS*, 2013 – Cutajar, Bonilla, Michiardi, Filippone, *ICML*, 2017

Learning Deep Gaussian Processes

- Inference requires calculating integrals of this kind:

$$
\begin{aligned}
p(\mathbf{Y}|\mathbf{X},\theta) \;\; = \;\; \int p\left(\mathbf{Y}|\mathbf{F}^{(N_\mathrm{h})},\theta^{(N_\mathrm{h})}\right) \times \\
p\left(\mathbf{F}^{(N_\mathrm{h})}|\mathbf{F}^{(N_\mathrm{h}-1)},\theta^{(N_\mathrm{h}-1)}\right) \times \ldots \times \\
p\left(\mathbf{F}^{(1)}|\mathbf{X},\theta^{(0)}\right) d\mathbf{F}^{(N_\mathrm{h})} \ldots d\mathbf{F}^{(1)}
\end{aligned}
$$

- Extremely challenging!

Inference for DGPs

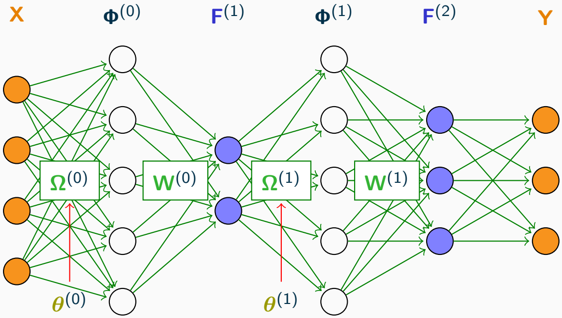
- Inducing-variable approximations
  - VI+Titsias
    - Damianou and Lawrence (AISTATS, 2013)
    - Hensman and Lawrence, (arXiv, 2014)
    - Salimbeni and Deisenroth, (NeurIPS, 2017)
  - EP+FITC: Bui et al. (ICML, 2016)
  - MCMC+Titsias
    - Havasi et al (arXiv, 2018)
- VI+Random feature-based approximations
  - Gal and Ghahramani (ICML 2016)
  - Cutajar et al. (ICML 2017)

Inference for DGPs

- Inducing-variable approximations
  - VI+Titsias
    - Damianou and Lawrence (AISTATS, 2013)
    - Hensman and Lawrence, (arXiv, 2014)
    - Salimbeni and Deisenroth, (NeurIPS, 2017)
  - EP+FITC: Bui et al. (ICML, 2016)
  - MCMC+Titsias
    - Havasi et al (arXiv, 2018)
- VI+Random feature-based approximations
  - Gal and Ghahramani (ICML 2016)
  - Cutajar et al. (ICML 2017)

Example: DGPs with Random Features are Bayesian DNNs

Recall RF approximations to GPs (part II-a). Then we have:



Stochastic Variational Inference

- Define $\mathbf{\Psi} = (\mathbf{\Omega}^{(0)}, \ldots, \mathbf{W}^{(0)}, \ldots)$
- Lower bound for $\log\left[p(\mathbf{Y}|\mathbf{X}, \theta)\right]$

$$\mathbb{E}_{q(\mathbf{\Psi})}\left(\log\left[p\left(\mathbf{Y}|\mathbf{X}, \mathbf{\Psi}, \theta\right)\right]\right) - \mathrm{DKL}\left[q(\mathbf{\Psi}) \| p\left(\mathbf{\Psi}|\theta\right)\right],$$

  where $q(\mathbf{\Psi})$ approximates $p(\mathbf{\Psi}|\mathbf{Y}, \theta)$.

- DKL computable analytically if q and p are Gaussian!

Optimize the lower bound wrt the parameters of $q(\mathbf{\Psi})$

Stochastic Variational Inference

- Assume that the likelihood factorizes

$$p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\Psi}, \theta) = \prod_k p(\mathbf{y}_k|\mathbf{x}_k, \boldsymbol{\Psi}, \theta)$$

- Doubly stochastic **unbiased** estimate of the expectation term

Stochastic Variational Inference

- Assume that the likelihood factorizes

$$p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\Psi}, \theta) = \prod_k p(\mathbf{y}_k|\mathbf{x}_k, \boldsymbol{\Psi}, \theta)$$

- Doubly stochastic **unbiased** estimate of the expectation term
  - Mini-batch

$$\mathbb{E}_{q(\boldsymbol{\Psi})}\left(\log\left[p\left(\mathbf{Y}|\mathbf{X}, \boldsymbol{\Psi}, \theta\right)\right]\right) \approx \frac{n}{m} \sum_{k \in \mathcal{I}_m} \mathbb{E}_{q(\boldsymbol{\Psi})}\left(\log\left[p(\mathbf{y}_k|\mathbf{x}_k, \boldsymbol{\Psi}, \theta)\right]\right)$$

Stochastic Variational Inference

- Assume that the likelihood factorizes

$$p(\mathbf{Y}|\mathbf{X}, \mathbf{\Psi}, \theta) = \prod_k p(\mathbf{y}_k|\mathbf{x}_k, \mathbf{\Psi}, \theta)$$

- Doubly stochastic **unbiased** estimate of the expectation term
  - Mini-batch

$$\mathbb{E}_{q(\mathbf{\Psi})}\left(\log\left[p\left(\mathbf{Y}|\mathbf{X}, \mathbf{\Psi}, \theta\right)\right]\right) \approx \frac{n}{m} \sum_{k \in \mathcal{I}_m} \mathbb{E}_{q(\mathbf{\Psi})}\left(\log\left[p(\mathbf{y}_k|\mathbf{x}_k, \mathbf{\Psi}, \theta)\right]\right)$$

  - Monte Carlo

$$\mathbb{E}_{q(\mathbf{\Psi})}\left(\log\left[p(\mathbf{y}_k|\mathbf{x}_k, \mathbf{\Psi}, \theta)\right]\right) \approx \frac{1}{N_{\mathrm{MC}}} \sum_{r=1}^{N_{\mathrm{MC}}} \log[p(\mathbf{y}_k|\mathbf{x}_k, \tilde{\mathbf{\Psi}}_r, \theta)]$$

with $\tilde{\mathbf{\Psi}}_r \sim q(\mathbf{\Psi})$.

Stochastic Variational Inference

- Reparameterization trick

$$(\tilde{\mathbf{W}}_r^{(l)})_{ij} = \sigma_{ij}^{(l)} \varepsilon_{rij}^{(l)} + \mu_{ij}^{(l)},$$

  with $\varepsilon_{rij}^{(l)} \sim \mathcal{N}(0, 1)$
- ... same for $\boldsymbol{\Omega}$
- Variational parameters

$$\mu_{ij}^{(l)}, (\sigma^2)_{ij}^{(l)} \ldots$$

  ... and the ones for $\boldsymbol{\Omega}$
- Optimization with automatic differentiation in TensorFlow

Kingma and Welling, *ICLR*, 2014

Other Interesting GP/DGP-based Models

Other Interesting GP/DGP-Based Models (1)

Convolutional GPs and DGPs

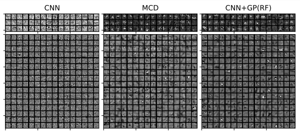
- Wilson et al (NeuriPS, 2016)
- van der Wilk et al (NeurIPS, 2017)
- Bradshaw et al (Arxiv, 2017)
- Tran et al (AISTATS, 2019)

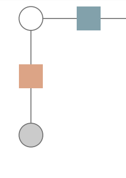
Structured Prediction

- Galliani et al (AISTATS, 2017)

Network-structure discovery

- Linderman and Adams (ICML, 2014)
- Dezfouli, Bonilla and Nock (ICML, 2018)





Other Interesting GP/DGP-Based Models (2)

Autoencoders

- Dai et al (ICLR, 2015); Domingues et al (Mach. Learn., 2018)

Constrained dynamics

- Lorenzi and Filippone, (ICML), 2018

Reinforcement Learning

- Rasmussen & Kauss (NIPS, 2004); Engel et al (ICML, 2005)
- Deisenroth and Rasmussen (ICML, 2011)
- Martin and Englot (Arxiv, 2018)

Doubly stochastic Poisson processes

- Adams et al (ICML, 2009); Lloyd et al (ICML, 2015)
- John and Hensman (ICML, 2018)
- Aglietti, Damoulas and Bonilla (AISTATS, 2019)

Conclusions

Applications and extensions of GP models by using more complex priors (e.g. coupled, compositions) and likelihoods

- Multi-task GPs by using correlated priors
- Dimensionality reduction via the GPLVM
- Probabilistic numerics, e.g. Bayesian optimisation
- Deep GPs
- Convolutional GPs
- Other settings such as RL, structured prediction, Poisson point processes