Probabilistic Modelling and Reasoning: A Machine Learning Approach

Approximate Inference

Edwin V. Bonilla

Principal Research Scientist, CSIRO's Data61
Associate Professor (Hon.), Australian National University

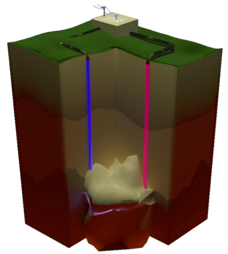
December 16th, 2021

This Lecture: Outline

1. Latent Gaussian Process Models (LGPMs)
2. Variational Inference
3. Scalability through Inducing Variables and Stochastic Variational Inference (SVI)
4. Challenges & Opportunities
5. Theory
6. Code

Challenges in Bayesian Reasoning with Gaussian Process Priors

- $p(\mathbf{f})$: prior over geology and rock properties
- $p(\mathbf{y} \mid \mathbf{f})$: observation model's likelihood



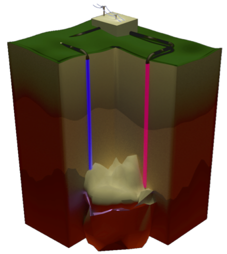
$20 Million geothermal well

Geol. surveys and explorations

Challenges in Bayesian Reasoning with Gaussian Process Priors

- $p(\mathbf{f})$: prior over geology and rock properties
- $p(\mathbf{y} \mid \mathbf{f})$: observation model's likelihood
- $p(\mathbf{f}|\mathbf{y})$: posterior geological model:

$$p(\mathbf{f} \mid \mathbf{y}, \boldsymbol{\theta}) = \frac{p(\mathbf{f} \mid \boldsymbol{\theta}) p(\mathbf{y} \mid \mathbf{f})}{\underbrace{\int p(\mathbf{f} \mid \boldsymbol{\theta}) p(\mathbf{y} \mid \mathbf{f}) d\mathbf{f}}_{\text{hard bit}}}$$



$20 Million geothermal well

Geol. surveys and explorations

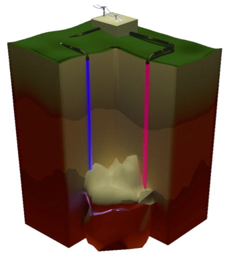
Challenges in Bayesian Reasoning with Gaussian Process Priors

- $p(\mathbf{f})$: prior over geology and rock properties
- $p(\mathbf{y} \mid \mathbf{f})$: observation model's likelihood
- $p(\mathbf{f}|\mathbf{y})$: posterior geological model:

$$p(\mathbf{f} \mid \mathbf{y}, \theta) = \frac{p(\mathbf{f} \mid \theta) p(\mathbf{y} \mid \mathbf{f})}{\underbrace{\int p(\mathbf{f} \mid \theta) p(\mathbf{y} \mid \mathbf{f}) d\mathbf{f}}_{\text{hard bit}}}$$

Challenges:

- Non-linear likelihood models
- Large datasets

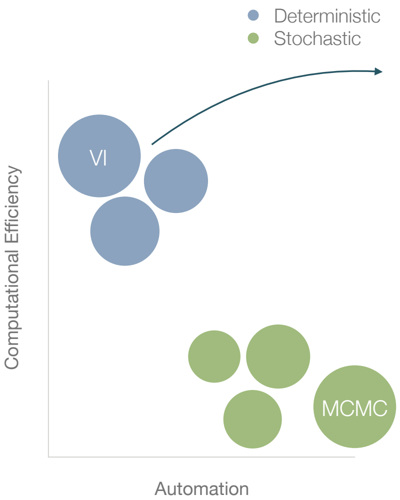


$20 Million geothermal well

Geol. surveys and explorations

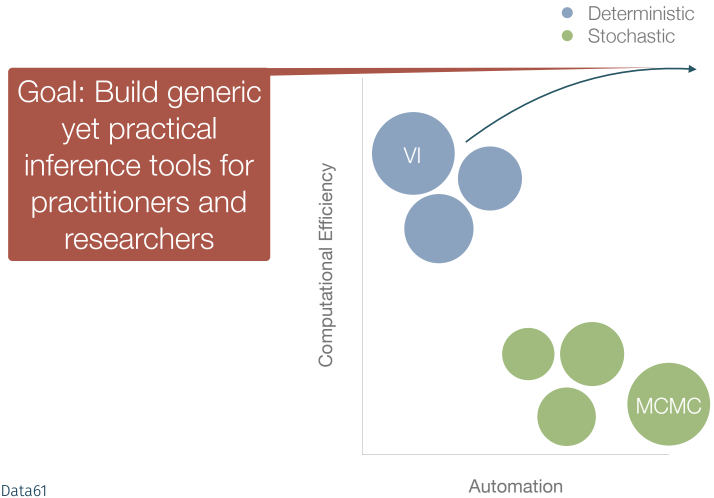
Automated Probabilistic Reasoning

- Approximate inference



Automated Probabilistic Reasoning

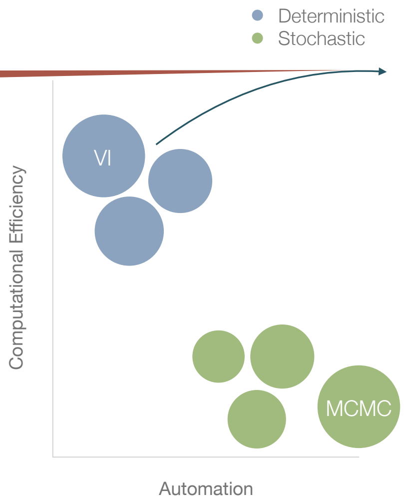
- Approximate inference



Automated Probabilistic Reasoning

- Approximate inference

Goal: Build generic yet practical inference tools for practitioners and researchers



- Other dimensions:
  - Accuracy
  - Convergence

Outline

1. Latent Gaussian Process Models (LGPMs)
2. Variational Inference
3. Scalability through Inducing Variables and Stochastic Variational Inference (SVI)
4. Challenges & Opportunities
5. Theory
6. Code

Latent Gaussian Process Models (LGPMs)

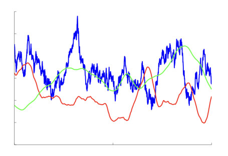
Latent Gaussian Process Models (LGPMs)

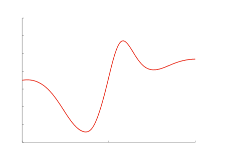
Supervised learning $\mathcal{D} = \{\mathbf{x}_n, \mathbf{y}_n\}_{n=1}^{N}$

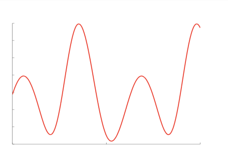
- Factorised GP priors over Q latent functions:

$$f_j(\mathbf{x}) \sim \mathcal{GP}(0, \kappa_j(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}))$$

$$p(\mathbf{F} \mid \mathbf{X}, \boldsymbol{\theta}) = \prod_{j=1}^{Q} \mathcal{N}(\mathbf{F}_{\cdot j}; \mathbf{0}, \mathbf{K}_j)$$



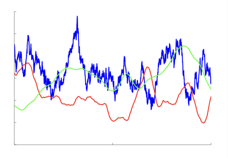




Latent Gaussian Process Models (LGPMs)

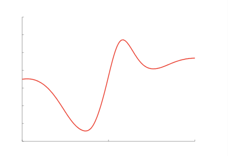
Supervised learning $\mathcal{D} = \{\mathbf{x}_n, \mathbf{y}_n\}_{n=1}^{N}$

- Factorised GP priors over Q latent functions:

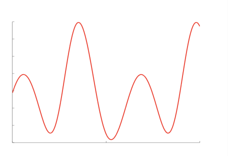


$$f_j(\mathbf{x}) \sim \mathcal{GP}(0, \kappa_j(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}))$$

$$p(\mathbf{F} \mid \mathbf{X}, \boldsymbol{\theta}) = \prod_{j=1}^{Q} \mathcal{N}(\mathbf{F}_{\cdot j}; \mathbf{0}, \mathbf{K}_j)$$



- Factorised likelihood over observations



$$p(\mathbf{Y} \mid \mathbf{X}, \mathbf{F}, \boldsymbol{\phi}) = \prod_{n=1}^{N} p(\mathbf{Y}_{n\cdot} \mid \mathbf{F}_{n\cdot}, \boldsymbol{\phi})$$

Latent Gaussian Process Models (LGPMs)

Supervised learning $\mathcal{D} = \{\mathbf{x}_n, \mathbf{y}_n\}_{n=1}^{N}$

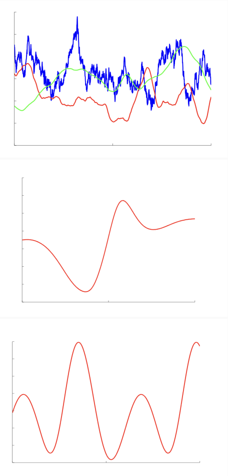
- Factorised GP priors over Q latent functions:

$$f_j(\mathbf{x}) \sim \mathcal{GP}(0, \kappa_j(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}))$$

$$p(\mathbf{F} \,|\, \mathbf{X}, \boldsymbol{\theta}) = \prod_{j=1}^{Q} \mathcal{N}(\mathbf{F}_{\cdot j}; \mathbf{0}, \mathbf{K}_j)$$

- Factorised likelihood over observations

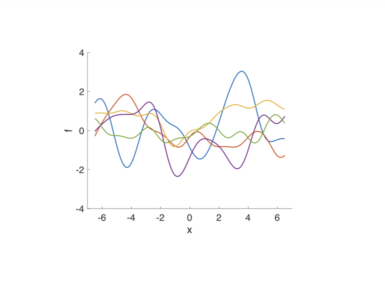
$$p(\mathbf{Y} \,|\, \mathbf{X}, \mathbf{F}, \phi) = \prod_{n=1}^{N} p(\mathbf{Y}_{n\cdot} \,|\, \mathbf{F}_{n\cdot}, \phi)$$

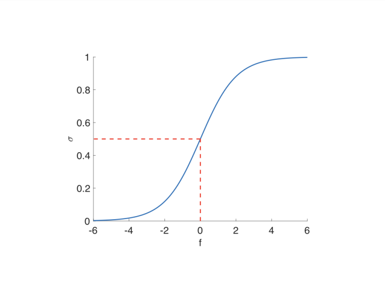


What can we model within this framework?

Examples of LGPMs (1)

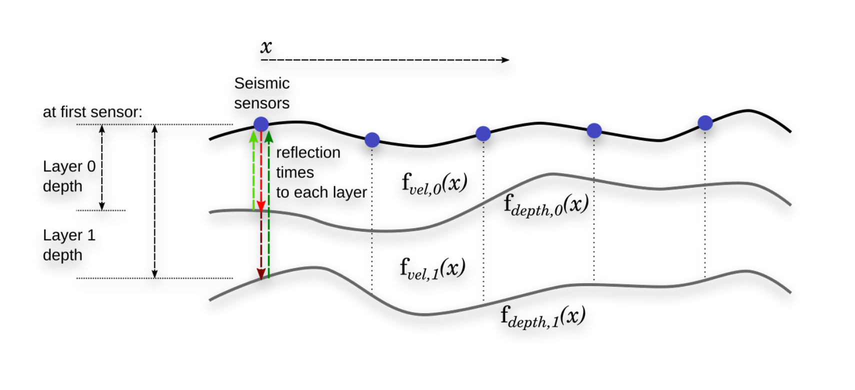
- Multi-output regression
- Multi-class classification
  - $P = Q$ classes
  - softmax likelihood





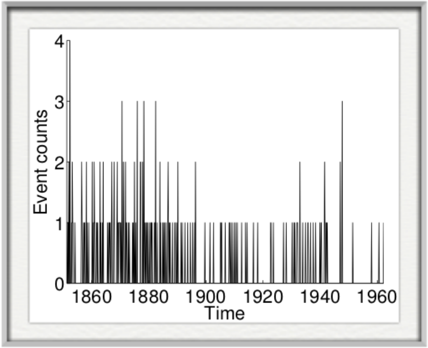
Examples of LGPMs (2)

- Inversion problems



Examples of LGPMs (3)

- Log Gaussian Cox processes (LGCPs)



Inference in LGPMs

We only require access to ‘black-box’ likelihoods. *How can we carry out inference in these general models?*

Variational Inference

Variational Inference (VI): Optimise Rather than Integrate

Recall our posterior estimation problem:

$$\underbrace{p(\mathsf{F} \mid \mathsf{Y})}_{\text{posterior}} = \frac{1}{\underbrace{p(\mathsf{Y})}_{\substack{\text{marginal} \\ \text{likelihood}}}} \underbrace{p(\mathsf{F})}_{\text{prior}} \underbrace{p(\mathsf{Y} \mid \mathsf{F})}_{\substack{\text{conditional} \\ \text{likelihood}}}$$

Variational Inference (VI): Optimise Rather than Integrate

Recall our posterior estimation problem:

$$\underbrace{p(\mathsf{F} \mid \mathsf{Y})}_{\text{posterior}} = \frac{1}{\underbrace{p(\mathsf{Y})}_{\substack{\text{marginal}\\ \text{likelihood}}}} \underbrace{p(\mathsf{F})}_{\text{prior}} \underbrace{p(\mathsf{Y} \mid \mathsf{F})}_{\substack{\text{conditional}\\ \text{likelihood}}}$$

- Estimating $p(\mathsf{Y}) = \int p(\mathsf{F}) p(\mathsf{Y} \mid \mathsf{F}) d\mathsf{F}$ is hard

Variational Inference (VI): Optimise Rather than Integrate

Recall our posterior estimation problem:

$$\underbrace{p(\mathsf{F} \mid \mathsf{Y})}_{\text{posterior}} = \frac{1}{\underbrace{p(\mathsf{Y})}_{\substack{\text{marginal} \\ \text{likelihood}}}} \underbrace{p(\mathsf{F})}_{\text{prior}} \underbrace{p(\mathsf{Y} \mid \mathsf{F})}_{\substack{\text{conditional} \\ \text{likelihood}}}$$

- Estimating $p(\mathsf{Y}) = \int p(\mathsf{F}) p(\mathsf{Y} \mid \mathsf{F}) d\mathsf{F}$ is hard
- Instead, approximate $q(\mathsf{F} \mid \boldsymbol{\lambda}) \approx p(\mathsf{F} \mid \mathsf{Y})$ to minimize:

$$\mathrm{KL}\left[q(\mathsf{F} \mid \boldsymbol{\lambda}) \,\|\, p(\mathsf{F} \mid \mathsf{Y})\right] \stackrel{\text{def}}{=} \mathbb{E}_{q(\mathsf{F} \mid \boldsymbol{\lambda})} \log \frac{q(\mathsf{F} \mid \boldsymbol{\lambda})}{p(\mathsf{F} \mid \mathsf{Y})}$$

Variational Inference (VI): Optimise Rather than Integrate

Recall our posterior estimation problem:

$$\underbrace{p(\mathsf{F} \mid \mathsf{Y})}_{\text{posterior}} = \frac{1}{\underbrace{p(\mathsf{Y})}_{\substack{\text{marginal}\\ \text{likelihood}}}} \underbrace{p(\mathsf{F})}_{\text{prior}} \underbrace{p(\mathsf{Y} \mid \mathsf{F})}_{\substack{\text{conditional}\\ \text{likelihood}}}$$

- Estimating $p(\mathsf{Y}) = \int p(\mathsf{F}) p(\mathsf{Y} \mid \mathsf{F}) d\mathsf{F}$ is hard
- Instead, approximate $q(\mathsf{F} \mid \boldsymbol{\lambda}) \approx p(\mathsf{F} \mid \mathsf{Y})$ to minimize:

$$\text{KL}\left[q(\mathsf{F} \mid \boldsymbol{\lambda}) \parallel p(\mathsf{F} \mid \mathsf{Y})\right] \stackrel{\text{def}}{=} \mathbb{E}_{q(\mathsf{F} \mid \boldsymbol{\lambda})} \log \frac{q(\mathsf{F} \mid \boldsymbol{\lambda})}{p(\mathsf{F} \mid \mathsf{Y})}$$

Properties:

$$\text{KL}\left[q \parallel p\right] \geq 0,$$

$$\text{KL}\left[q \parallel p\right] = 0 \text{ iff } q = p.$$

Decomposition of the Marginal Likelihood

$$\log p(\mathbf{Y}) = \text{KL}\left[q(\mathbf{F} \,|\, \boldsymbol{\lambda}) \,\|\, p(\mathbf{F} \,|\, \mathbf{Y})\right] + \mathcal{L}_{\text{ELBO}}(\boldsymbol{\lambda})$$

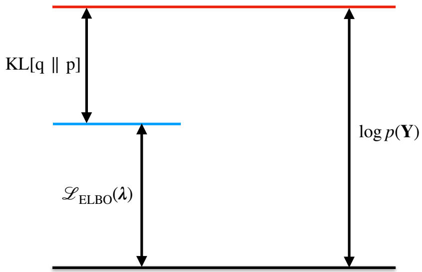


Fig reproduced from Bishop (2006)

- $\mathcal{L}_{\text{ELBO}}(\boldsymbol{\lambda})$ is a lower bound on the log marginal likelihood
- The optimum is achieved when $q = p$
- Maximizing $\mathcal{L}_{\text{ELBO}}(\boldsymbol{\lambda}) \equiv$ minimizing $\text{KL}\left[q(\mathbf{F} \,|\, \boldsymbol{\lambda}) \,\|\, p(\mathbf{F} \,|\, \mathbf{Y})\right]$

Variational Inference Strategy

- The evidence lower bound $\mathcal{L}_{\text{ELBO}}(\boldsymbol{\lambda})$ can be written as:

$$\mathcal{L}_{\text{ELBO}}(\boldsymbol{\lambda}) \stackrel{\text{def}}{=} \underbrace{\mathbb{E}_{q(\mathsf{F} \mid \boldsymbol{\lambda})} \log p(\mathsf{Y} \mid \mathsf{F})}_{\text{expected log likelihood (ELL)}} - \underbrace{\text{KL}\left[q(\mathsf{F} \mid \boldsymbol{\lambda}) \parallel p(\mathsf{F})\right]}_{\text{KL(approx. posterior } \parallel \text{ prior)}}$$

- ELL is a model-fit term and KL is a penalty term

Variational Inference Strategy

- The evidence lower bound $\mathcal{L}_{\text{ELBO}}(\boldsymbol{\lambda})$ can be written as:

$$\mathcal{L}_{\text{ELBO}}(\boldsymbol{\lambda}) \stackrel{\text{def}}{=} \underbrace{\mathbb{E}_{q(\mathbf{F} \mid \boldsymbol{\lambda})} \log p(\mathbf{Y} \mid \mathbf{F})}_{\text{expected log likelihood (ELL)}} - \underbrace{\text{KL}\left[q(\mathbf{F} \mid \boldsymbol{\lambda}) \,\|\, p(\mathbf{F})\right]}_{\text{KL(approx. posterior } \| \text{ prior)}}$$

- ELL is a model-fit term and KL is a penalty term

- What family of distributions?
  - As flexible as possible
  - Tractability is the main constraint
  - No risk of over-fitting

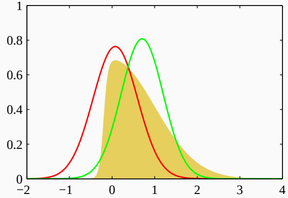


Fig from Bishop (2006)

Variational Inference Strategy

- The evidence lower bound $\mathcal{L}_{\text{ELBO}}(\boldsymbol{\lambda})$ can be written as:

$$\mathcal{L}_{\text{ELBO}}(\boldsymbol{\lambda}) \stackrel{\text{def}}{=} \underbrace{\mathbb{E}_{q(\mathbf{F} \mid \boldsymbol{\lambda})} \log p(\mathbf{Y} \mid \mathbf{F})}_{\text{expected log likelihood (ELL)}} - \underbrace{\text{KL}\left[q(\mathbf{F} \mid \boldsymbol{\lambda}) \parallel p(\mathbf{F})\right]}_{\text{KL(approx. posterior } \parallel \text{ prior)}}$$

- ELL is a model-fit term and KL is a penalty term

- What family of distributions?
  - As flexible as possible
  - Tractability is the main constraint
  - No risk of over-fitting

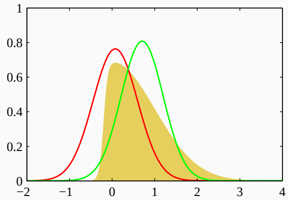


Fig from Bishop (2006)

We want to maximise $\mathcal{L}_{ELBO}(\boldsymbol{\lambda})$ wrt variational parameters $\boldsymbol{\lambda}$

Automated VI for LGPMs

(Nguyen and Bonilla, NeurIPS, 2014)

Goal: Approximate posterior $p(\mathsf{F} \mid \mathsf{Y})$ with variational distribution

$$q(\mathsf{F} \mid \boldsymbol{\lambda}) = \sum_{k=1}^{K} \pi_k q_k(\mathsf{F} \mid \boldsymbol{\lambda}) = \sum_{k=1}^{K} \pi_k \prod_{j=1}^{Q} \mathcal{N}(\mathsf{F}_k; \mathsf{m}_{kj}, \mathsf{S}_{kj})$$

with variational parameters $\boldsymbol{\lambda} = \{\mathsf{m}_{kj}, \mathsf{S}_{kj}\}$,

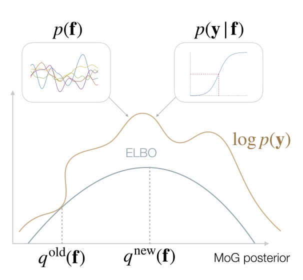
Goal: Approximate posterior $p(\mathsf{F} \mid \mathsf{Y})$ with variational distribution

$$q(\mathsf{F} \mid \boldsymbol{\lambda}) = \sum_{k=1}^{K} \pi_k q_k(\mathsf{F} \mid \boldsymbol{\lambda}) = \sum_{k=1}^{K} \pi_k \prod_{j=1}^{Q} \mathcal{N}(\mathsf{F}_k; \mathbf{m}_{kj}, \mathsf{S}_{kj})$$

with variational parameters $\boldsymbol{\lambda} = \{\mathbf{m}_{kj}, \mathsf{S}_{kj}\}$,

Recall $\mathcal{L}_{\text{ELBO}}(\boldsymbol{\lambda})$ = - KL + ELL:

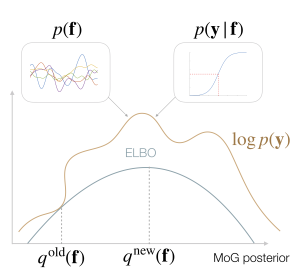
- KL term can be bounded using Jensen's inequality
  - Exact gradients of parameters



Goal: Approximate posterior $p(\mathsf{F} \mid \mathsf{Y})$ with variational distribution

$$q(\mathsf{F} \mid \boldsymbol{\lambda}) = \sum_{k=1}^{K} \pi_k q_k(\mathsf{F} \mid \boldsymbol{\lambda}) = \sum_{k=1}^{K} \pi_k \prod_{j=1}^{Q} \mathcal{N}(\mathsf{F}_k; \mathbf{m}_{kj}, \mathsf{S}_{kj})$$

with variational parameters $\boldsymbol{\lambda} = \{\mathbf{m}_{kj}, \mathsf{S}_{kj}\}$,



Recall $\mathcal{L}_{\text{ELBO}}(\boldsymbol{\lambda})$ = - KL + ELL:

- KL term can be bounded using Jensen's inequality
  - Exact gradients of parameters

ELL and its gradients can be estimated *efficiently*

Expected Log Likelihood Term

Th.1: Efficient estimation

The ELL and its gradients can be estimated using expectations over univariate Gaussian distributions.

$$q_{k(n)} \stackrel{\text{def}}{=} q_{k(n)}(\mathbf{F}_{\cdot n} \mid \boldsymbol{\lambda}_{k(n)})$$

$$\mathbb{E}_{q_k} \log p(\mathbf{Y} \mid \mathbf{F}) = \sum_{n=1}^{N} \mathbb{E}_{q_{k(n)}} \log p(\mathbf{Y}_{n\cdot} \mid \mathbf{F}_{n\cdot})$$

$$\nabla_{\boldsymbol{\lambda}_{k(n)}} \mathbb{E}_{q_{k(n)}} \log p(\mathbf{Y}_{n\cdot} \mid \mathbf{F}_{n\cdot}) = \mathbb{E}_{q_{k(n)}} \nabla_{\boldsymbol{\lambda}_{k(n)}} \log q_{k(n)}(\mathbf{F}_{\cdot n} \mid \boldsymbol{\lambda}_{k(n)}) \log p(\mathbf{Y}_{n\cdot} \mid \mathbf{F}_{n\cdot})$$

Expected Log Likelihood Term

Th.1: Efficient estimation
The ELL and its gradients can be estimated using expectations over univariate Gaussian distributions.

$$q_{k(n)} \overset{\text{def}}{=} q_{k(n)}(\mathsf{F}_{\cdot n} \mid \boldsymbol{\lambda}_{k(n)})$$

$$\mathbb{E}_{q_k} \log p(\mathsf{Y} \mid \mathsf{F}) = \sum_{n=1}^{N} \mathbb{E}_{q_{k(n)}} \log p(\mathsf{Y}_{n\cdot} \mid \mathsf{F}_{n\cdot})$$

$$\nabla_{\boldsymbol{\lambda}_{k(n)}} \mathbb{E}_{q_{k(n)}} \log p(\mathsf{Y}_{n\cdot} \mid \mathsf{F}_{n\cdot}) = \mathbb{E}_{q_{k(n)}} \nabla_{\boldsymbol{\lambda}_{k(n)}} \log q_{k(n)}(\mathsf{F}_{\cdot n} \mid \boldsymbol{\lambda}_{k(n)}) \log p(\mathsf{Y}_{n\cdot} \mid \mathsf{F}_{n\cdot})$$

Practical consequences

- Can use unbiased Monte Carlo estimates
- Gradients of the likelihood are not required
- Holds $\forall Q \geq 1$

Scalability through Inducing Variables and Stochastic Variational Inference (SVI)

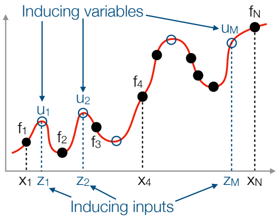
Inducing Variables in GP Models

Inducing variables $\mathbf{u}$

- Latent values of the GP, as $\mathbf{f}$ and $\mathbf{f}_*$
- Usually marginalized (integrated out)

Inducing inputs $\mathbf{Z}$

- Corresponding input location, as $\mathbf{x}$
- Imprint on final solution

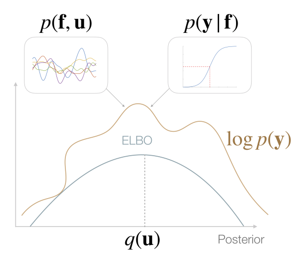


Generalization of "support points", "active set", "pseudo-inputs"

Variational Learning of Inducing Variables

(Titisias, AISTATS, 2009)

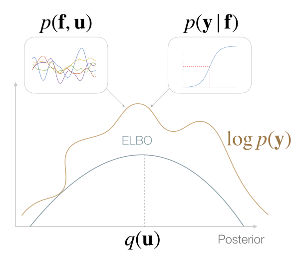
- Augmented prior $p(\mathbf{f}, \mathbf{u}) = p(\mathbf{f} \mid \mathbf{u}) p(\mathbf{u})$, exact marginal $p(\mathbf{f})$
- Approximate posterior $q(\mathbf{f}, \mathbf{u}) = p(\mathbf{f} \mid \mathbf{u}) q(\mathbf{u})$



Variational Learning of Inducing Variables

(Titisias, AISTATS, 2009)

- Augmented prior $p(\mathbf{f}, \mathbf{u}) = p(\mathbf{f} \mid \mathbf{u})p(\mathbf{u})$, exact marginal $p(\mathbf{f})$
- Approximate posterior $q(\mathbf{f}, \mathbf{u}) = p(\mathbf{f} \mid \mathbf{u})q(\mathbf{u})$
- Cubic operations on N 'vanish'

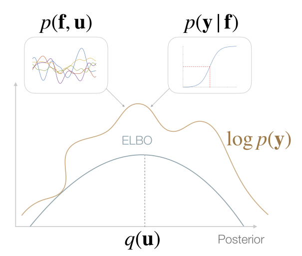


Variational Learning of Inducing Variables

(Titisias, AISTATS, 2009)

- Augmented prior $p(\mathbf{f}, \mathbf{u}) = p(\mathbf{f} \mid \mathbf{u})p(\mathbf{u})$, exact marginal $p(\mathbf{f})$
- Approximate posterior $q(\mathbf{f}, \mathbf{u}) = p(\mathbf{f} \mid \mathbf{u})q(\mathbf{u})$

- Cubic operations on N 'vanish'
- Exact optimal solution for Gaussian likelihood

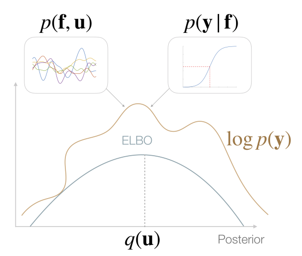


Variational Learning of Inducing Variables

(Titisias, AISTATS, 2009)

- Augmented prior $p(\mathbf{f}, \mathbf{u}) = p(\mathbf{f} \mid \mathbf{u})p(\mathbf{u})$, exact marginal $p(\mathbf{f})$
- Approximate posterior $q(\mathbf{f}, \mathbf{u}) = p(\mathbf{f} \mid \mathbf{u})q(\mathbf{u})$

- Cubic operations on N 'vanish'
- Exact optimal solution for Gaussian likelihood
- Hyper-parameters and inducing inputs optimized *jointly*

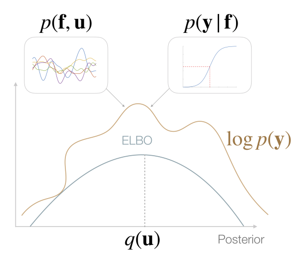


Variational Learning of Inducing Variables

(Titisias, AISTATS, 2009)

- Augmented prior $p(\mathbf{f}, \mathbf{u}) = p(\mathbf{f} \mid \mathbf{u})p(\mathbf{u})$, exact marginal $p(\mathbf{f})$
- Approximate posterior $q(\mathbf{f}, \mathbf{u}) = p(\mathbf{f} \mid \mathbf{u})q(\mathbf{u})$

- Cubic operations on N 'vanish'
- Exact optimal solution for Gaussian likelihood
- Hyper-parameters and inducing inputs optimized *jointly*



Computation dominated by:

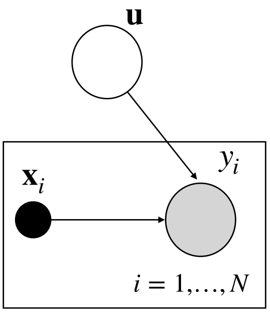
$$\mathbf{K}_{XZ}\mathbf{K}_{ZZ}^{-1}\mathbf{K}_{ZX}$$

Time cost $\mathcal{O}(NM^2)$, *can we do better?*

Stochastic Variational Inference for GP Models

Maintain an explicit representation of $q(\mathbf{u}) = \mathcal{N}(\mathbf{m}, \mathbf{S})$

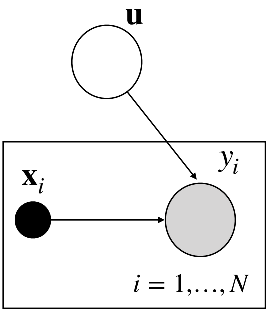
- Inducing variables act as global variables
- ELBO decomposes across observations
- Use stochastic optimization
- $\mathbf{K}_{\mathbf{x}_i \mathbf{Z}} \mathbf{K}_{\mathbf{ZZ}}^{-1} \mathbf{K}_{\mathbf{Z}\mathbf{x}_i}$: Time cost $\mathcal{O}(M^3) \rightarrow$ big data!



Stochastic Variational Inference for GP Models

Maintain an explicit representation of $q(\mathbf{u}) = \mathcal{N}(\mathbf{m}, \mathbf{S})$

- Inducing variables act as global variables
- ELBO decomposes across observations
- Use stochastic optimization
- $\mathbf{K}_{\mathbf{x}_i \mathbf{Z}} \mathbf{K}_{\mathbf{Z}\mathbf{Z}}^{-1} \mathbf{K}_{\mathbf{Z}\mathbf{x}_i}$: Time cost $\mathcal{O}(M^3) \rightarrow$ big data!

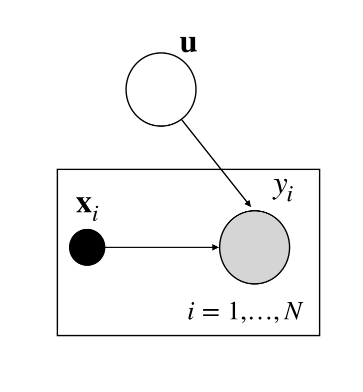


- Converge to optimal solution for Gaussian likelihoods (Hensman et al, UAI, 2013)

Stochastic Variational Inference for GP Models

Maintain an explicit representation of $q(\mathbf{u}) = \mathcal{N}(\mathbf{m}, \mathbf{S})$

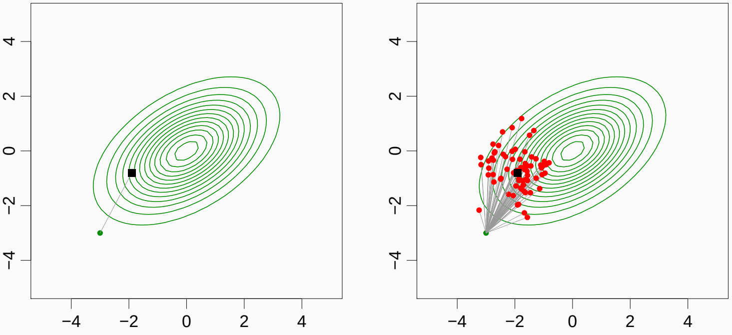
- Inducing variables act as global variables
- ELBO decomposes across observations
- Use stochastic optimization
- $\mathbf{K}_{\mathbf{x}_i \mathbf{Z}} \mathbf{K}_{\mathbf{Z}\mathbf{Z}}^{-1} \mathbf{K}_{\mathbf{Z}\mathbf{x}_i}$: Time cost $\mathcal{O}(M^3) \rightarrow$ big data!



- Converge to optimal solution for Gaussian likelihoods (Hensman et al, UAI, 2013)
- Generalization to LGPMs (Dezfouli & Bonilla, NeurIPS, 2015)

Stochastic Gradient Optimization

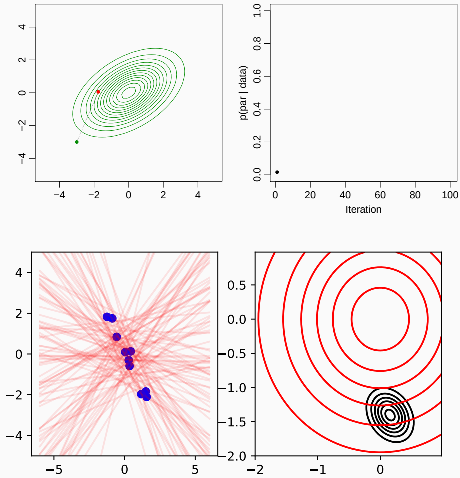
$$\mathbb{E}\left\{\widetilde{\nabla_{\text{vpar}}\text{LowerBound}}\right\} = \nabla_{\text{vpar}}\text{LowerBound}$$



Robbins and Monro, *AoMS*, 1951

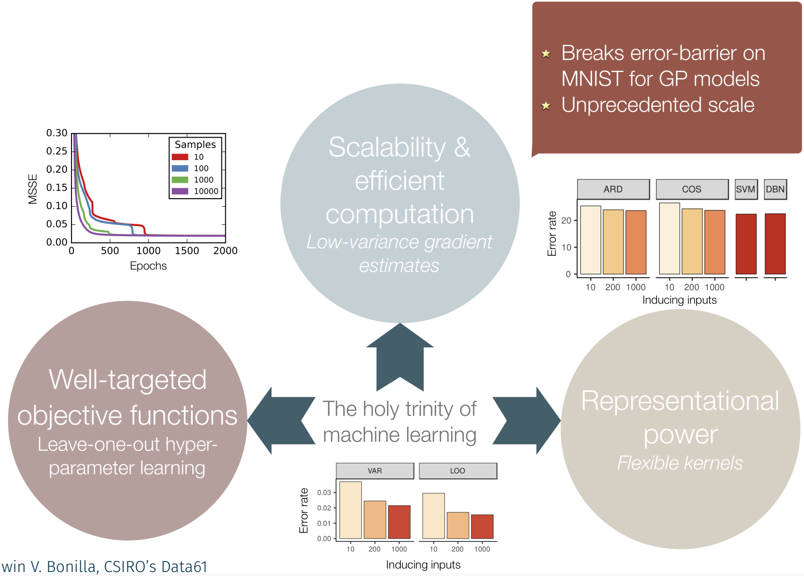
Stochastic Variational Inference

$$\text{vpar}' = \text{vpar} + \frac{\alpha_t}{2}\widetilde{\nabla_{\text{vpar}}}(\text{LowerBound}) \qquad \alpha_t \to 0$$



Further Developments: AutoGP

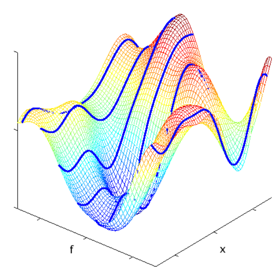
(Krauth et al UAI, 2017)

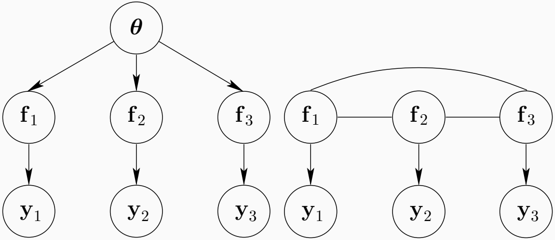


Challenges & Opportunities

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

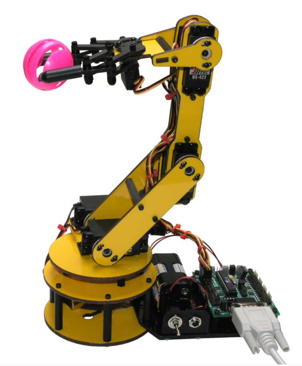




Data Fusion and Multi-task Learning (2)

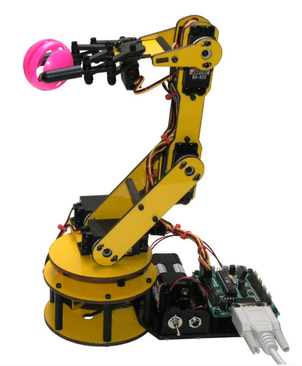
Multi-task GP (Bonilla et al, NeurIPS, 2008)

- $\mathrm{Cov}(f_\ell(\mathbf{x}), f_m(\mathbf{x}')) = \mathbf{K}^f_{\ell m}\kappa(\mathbf{x}, \mathbf{x}')$
- $\mathbf{K}$ 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)



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

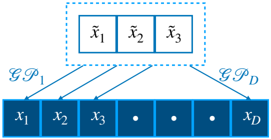
Generalisations and other settings:

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

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

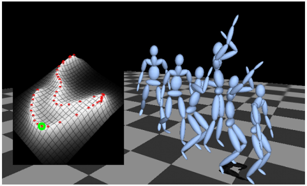
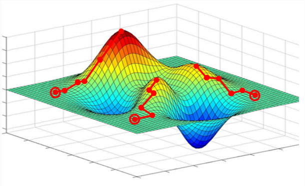


Fig. and cool videos at `http://grail.cs.washington.edu/projects/styleik/`

Probabilistic Numerics: Bayesian Optimisation (1)

Optimisation of black-box functions:

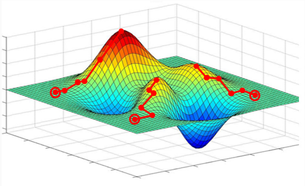
- Do not know their implementation
- Costly to evaluate
- Use GPs as surrogate models



Probabilistic Numerics: Bayesian Optimisation (1)

Optimisation of black-box functions:

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



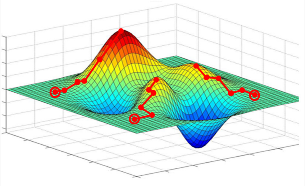
Vanilla BO iterates:

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



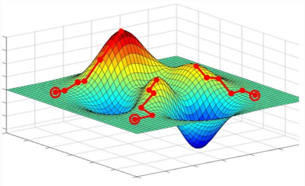
Vanilla BO iterates:

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



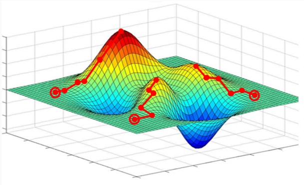
Vanilla BO iterates:

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

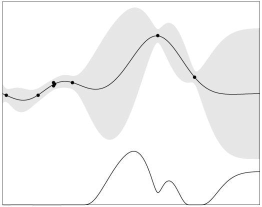


Fig. from Boyle (2007)

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**:

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

  - Simple ‘analytical form’
  - Exploration-exploitation

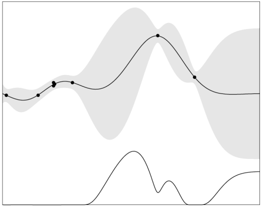


Fig. from Boyle (2007)

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

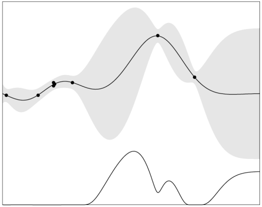


Fig. from Boyle (2007)

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

The Deep Learning Revolution

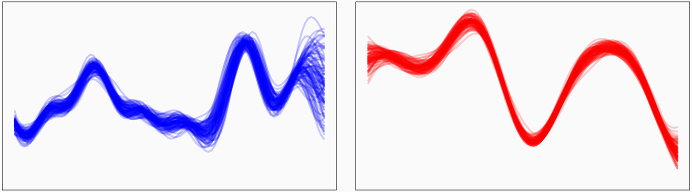
- Large representational power
- *Big data* learning through stochastic optimisation
- Exploit GPU and distributed computing
- Automatic differentiation
- Mature development of regularization (e.g., dropout)
- Application-specific representations (e.g., convolutional)

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

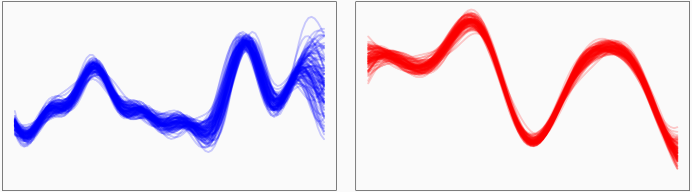


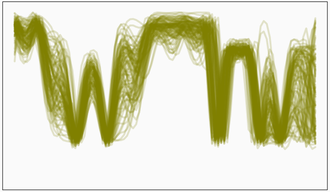
$$(f \circ g)(x)??$$

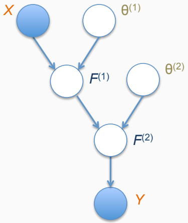
Damianou and Lawrence, *AISTATS*, 2013 – Cutajar, Bonilla, Michiardi, Filippone, *ICML*, 2017

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:

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

- Extremely challenging!

Inference for DGPs

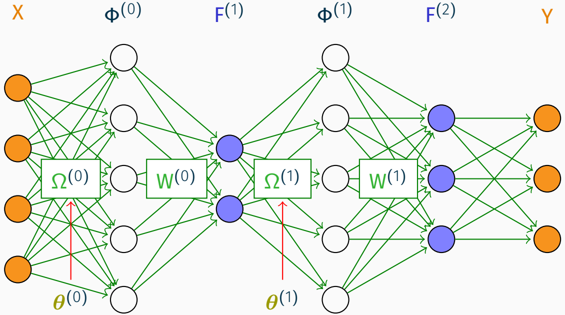
- Inducing-variable approximations
  - VI+Titsias
    - Damianou and Lawrence (AISTATS, 2013)
    - Hensman and Lawrence, (arXiv, 2014)
    - Salimbeni and Deisenroth, (NeurIPS, 2017)
  - EP+FITC: Bui et al. (ICML, 2016)
  - MCMC+Titsias
    - Havasi et al (arXiv, 2018)
- VI+Random feature-based approximations
  - 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 $\Psi = (\Omega^{(0)}, \ldots, \mathsf{W}^{(0)}, \ldots)$
- Lower bound for $\log\left[p(\mathsf{Y}|\mathsf{X}, \boldsymbol{\theta})\right]$

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

  where $q(\Psi)$ approximates $p(\Psi|\mathsf{Y}, \boldsymbol{\theta})$.
- DKL computable analytically if q and p are Gaussian!

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

Stochastic Variational Inference

- Assume that the likelihood factorizes

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

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

Stochastic Variational Inference

- Assume that the likelihood factorizes

$$p(\mathrm{Y}|\mathrm{X}, \Psi, \boldsymbol{\theta}) = \prod_{k} p(\mathrm{y}_k|\mathrm{x}_k, \Psi, \boldsymbol{\theta})$$

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

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

Stochastic Variational Inference

- Assume that the likelihood factorizes

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

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

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

  - ▶ Monte Carlo

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

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

Stochastic Variational Inference

- Reparameterization trick

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

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

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

  ... and the ones for Ω
- Optimization with automatic differentiation in TensorFlow

Kingma and Welling, *ICLR*, 2014

Convolutional GPs and DGPs

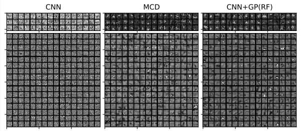
- Wilson et al (NeuriPS, 2016)
- van der Wilk et al (NeurIPS, 2017)
- Bradshaw et al (Arxiv, 2017)
- Tran et al (AISTATS, 2019)

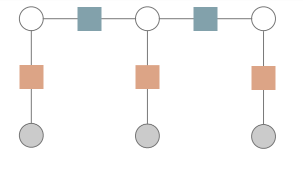
Structured Prediction

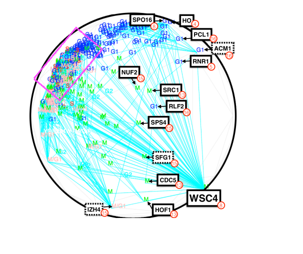
- Galliani et al (AISTATS, 2017)

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)

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)

Theory

Asymptotics & Consistency

- The GP posterior mean minimizes the following functional:

$$J(f) = \frac{1}{2}\|f\|^2_{\mathcal{H}} + \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - f(\mathbf{x}_i))$$

  where $\|f\|^2_{\mathcal{H}}$ is the RKHS norm corresponding to the covariance function κ.

- What happens when $N \to \infty$?

Asymptotics & Consistency

- The GP posterior mean minimizes the following functional:

$$J(f) = \frac{1}{2}\|f\|^2_{\mathcal{H}} + \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - f(\mathbf{x}_i))$$

where $\|f\|^2_{\mathcal{H}}$ is the RKHS norm corresponding to the covariance function κ.

- What happens when $N \to \infty$?
- f converges to $\mathbb{E}_{p(y,\mathbf{x})}[y|\mathbf{x}]$ ...
- ... under some regularity conditions (nondegenerate κ, regression function well-behaved)

GPs & Stochastic Differential Equations

- Consider the Markov process:

$$a_m \frac{d^m f(x)}{dx^m} + a_{m-1} \frac{d^{m-1} f(x)}{dx^{m-1}} + \dots a_1 \frac{df(x)}{dx} + a_0 f(x) = w(x)$$

  where $w(x)$ is a zero-mean white-noise process.
- The solution is a GP
- The covariance depends on the form of the SDE
- Solving SDEs is easy in low dimensions!
- We can solve GPs in $\mathcal{O}(N \log N)$

Saatçi, *Ph.D. Thesis*, 2011

Other Interesting Topics

- Average-case Learning Curves
- PAC-Bayesian Analysis
- Theory for Sparse GPs - Best Paper Award ICML 2019

Code

Code for Gaussian Processes

- python
  - GPy
- MatLab
  - gptoolbox
- R
  - kernlab

Code for Gaussian Processes - With Automatic Differentiation

- TensorFlow:
  - GPflow
  - AutoGP
- PyTorch
  - CandleGP
  - GPyTorch
  - BoTorch

- TensorFlow:
  - GPflow
  - Doubly-Stochastic DGPs
- PyTorch
  - DGPs with Random Features

Conclusions (1)

- LGPMs: General framework for GP priors and non-linear likelihoods
- Applications in multi-class classification, multi-output regression, modelling count data and more
- Generic inference via optimisation of the variational objective (ELBO)
- Scalability via inducing-variable approach
- AutoGP

Conclusions (2)

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

CSIRO's Data61: Looking for the Next Research Stars in ML

Interested in working at the cutting edge of research in ML and AI?

https://ebonilla.github.io/